ComSIS

Computer Science and Information Systems

Volume 21, Number 4, September 2024

Contents

Editorial
Guest Editorial: Explainable and trustworthy methods for next-generation artificial intelligence for Reasonable Data Size
Guest Editorial: Recent Advances in AI Methods for Image Processing: Theory, Algorithms, and Applications
Guest Editorial: Selected papers from the European Conference on Advances in Databases and Information Systems

Papers

1221 Threshold Segmentation Based on Information Fusion for Object Shadow Detection in Remote Sensing Images
Shoulin Yin, Liguo Wang, Lin Teng
1243 Tourist services management through clients scoring using a bio-inspired agent architecture
Raúl Moreno, Alberto Fernández-Isabel, Víctor Aceña, Isaac Martínde Diego, Javier M. Moguerza
1269 PTB-FLA Development Paradigm Adaptation for ChatGPT
Miroslav Popovic, Marko Popovic, Ivan Kastelan, Miodrag Djukic, Ilija Basicevic
1293 TRL-PROTAC: A pre-trained generator of PROTACs targeting specific proteins optimized by reinforcement learning
Yuhao Dai, Fei Zhu
1321 Psychological Effect Computation of Courtroom Arguments: A Deep Learning Approach of EEG Signal Data
Xuan Zhou, Yaming Liu, Baoqian Jiao, Hanzhen Ouyang, Weihui Dai
1335 A Method for Solving Reconfiguration Blueprints Based on Multi-Agent Reinforcement Learning
Jing Cheng, Wen Tan, Guangzhe Lv, Guodong Li, Wentao Zhang, Zihao Liu
1359 Biometric Lock with Facial Recognition Implemented with Deep Learning Techniques
José Misael Burruel-Zazueta, Héctor Rodríguez-Rangel, Gloria Ekaterine Peralta-Peñuñuri, Vicenç Puig Cayuela, Ignacio Algredo-Badillo, Luis Alberto Morales-Rosales
1389 BLSAE-SNIDS: A Bi-LSTM Sparse Autoencoder Framework for Satellite Network Intrusion Detection
Shi Shuxin, Han Bing, Wu Zhongdai, Han Dezhi, Wu Huafeng, Mei Xiaojun
1411 Sentiment Polarity Analysis of Love Letters: Evaluation of TextBlob, Vader, Flair, and Hugging Face Transformer
Gaganpreet Kaur, Amandeep Kaur, Meenu Khurana, Robertas Damaševičius

Special section: Explainable and trustworthy methods for next-generation artificial intelligence for Reasonable Data Size

1435 Advancing Crack Segmentation Detection: Introducing AAMC-Net Algorithm for Image Crack Analysis
Wang Xiaofang, Liu Chenfang, Hou Junliang, Zhou Liang
1457 Enhancing Architectural Image Processing: A Novel 2D to 3D Algorithm Using Improved Convolutional Neural Networks
Qianying Zou, Fengyu Liu, Yuan Liao
1483 Learning Discriminative Representations through an Attention Mechanism for Image-based Person Re-identification
Jing Liu, Guoqing Zhou
1499 Semantic Feature-Based Test Selection for Deep Neural Networks: A Frequency Domain Perspective
Zhouxian Jiang, Honghui Li, Xuetao Tian, Rui Wang
1523 AI Large Models Bring Great Opportunities to Reusable Design of CAD Software
Yunlei Sun, Bingyi Yan, Zhaotong Shao
1547 A Hybrid GA-Powell Algorithm for Geometric Constraint Solving
Sun Yunlei, Li Yucong
1567 ALFormer: Attribute Localization Transformer in Pedestrian Attribute Recognition
Yuxin Liu, Mingzhe Wang, Chao Li, Shuoyan Liu

Special section: Recent Advances in AI Methods for Image Processing: Theory, Algorithms, and Applications

1583 GCN-LSTM: Multi-label educational emotion prediction based on graph Convolutional network and long and short term memory network fusion label correlation in online social networks
Zhiguang Liu, Fengshuai Li, Guoyin Hao, Xiaoqing He, Yuanheng Zhang
1607 A multi-feature Fusion Model Based on Long and Short Term Memory Network and Improved Artificial Bee Colony Algorithm for English Text Classification
Tianying Wen
1629 A Novel Industrial Big Data Fusion Method Based on Q-learning and Cascade Classifier
Xi Zhang, Jiyue Wang, Ying Huang, Feiyue Zhu
1651 Remora Optimization Algorithm-based Adaptive Fusion via Ant Colony Optimization for Traveling Salesman Problem
Lin Piao
1673 GAN-DNADE: Image Encryption Algorithm Based on Generative Adversarial Network and DNA Dynamic Encoding
Xi Wang
1699 Multi-frame Network Feature Fusion Model and Self-attention Mechanism for Vehicle Lane Line Detection
Guang Zhu, Yajuan Liu, Jiyue Wang
1725 Stacked Denoised Auto-encoding Network-based Kernel Principal Component Analysis for Cyber Physical Systems Intrusion Detection in Business Management
Zhihao Song
1745 A Novel Image Enhancement Method Using Retinex-based Illumination Map Weighted Guided Filtering
Su Chen, Dahai Li
1765 Underwater Image Denoising Based on CurvedWave Filtering and Two-dimensional Variational Mode Decomposition
Lin Teng, Yulong Qiao, Shoulin Yin
1783 A Novel Deep Fully Convolutional Encoder-Decoder Network and Similarity Analysis for English Education Text Event Clustering Analysis
Zhenping Jing
1801 Attitude Estimation of Aircraft Based on Quaternion SRCKF-SLAM Algorithm
Dandan Wang, Zhaokun Zhu, Liang Yu, Hongjie Li, Kaituo Tan
1823 Spatio-Temporal-based Multi-level Aggregation Network for Physical Action Recognition
Yuhang Wang
1845 Multi-object Real-time Tracking for Intelligent Breeding of Animal
Fei Wang, Bin Xia, Liwu Pan
1865 MFE-Transformer: Adaptive English Text Named Entity Recognition Method Based on Multi-feature Extraction and Transformer
Liuxin Gao

Special section: Selected papers from the European Conference on Advances in Databases and Information Systems

1097 Natural Language Processing: An Overview of Models, Transformers and Applied Practices
Santiago Canchila, Carlos Meneses-Eraso, Javier Casanoves-Boix, Pascual Cortés-Pellicer, Fernando Castelló-Sirvent
1887 Identifying Occurrences of the Cnidarian Physalia physalis in Social Media Data
Heloisa F. Rocha, Lorena S. Nascimento, Leonardo Camargo, Mauricio A. Noernberg, Aurora T. Ramirez Pozo, Carmem S. Hara
1913 Automatic Conceptual Database Design based on Heterogeneous Source Artifacts
Goran Banjac, Drazen Brdjanin, Danijela Banjac
1963 Maritime Trajectory Mining: An Automatic Zones of Interests Discovery and Annotation Framework
Omar Ghannou, Etienne Thuillier, Omar Boucelma

ISSN: 2406-1018 (Online)

Computer Science and Information Systems

Vol 21, No 4, September 2024

ComSIS

Computer Science and Information Systems

Published by ComSIS Consortium

Volume 21, Number 4
September 2024

ComSIS is an international journal published by the ComSIS Consortium

ComSIS Consortium:
University of Belgrade:
Faculty of Organizational Science, Belgrade, Serbia
Faculty of Mathematics, Belgrade, Serbia
School of Electrical Engineering, Belgrade, Serbia
Serbian Academy of Science and Art:
Mathematical Institute, Belgrade, Serbia
Union University:
School of Computing, Belgrade, Serbia
University of Novi Sad:
Faculty of Sciences, Novi Sad, Serbia
Faculty of Technical Sciences, Novi Sad, Serbia
Technical Faculty "Mihajlo Pupin", Zrenjanin, Serbia
University of Niš:
Faculty of Electronic Engineering, Niš, Serbia
University of Montenegro:
Faculty of Economics, Podgorica, Montenegro

EDITORIAL BOARD:
Editor-in-Chief: Mirjana Ivanović, University of Novi Sad
Vice Editor-in-Chief: Boris Delibašić, University of Belgrade
Managing Editors:
Vladimir Kurbalija, University of Novi Sad
Miloš Radovanović, University of Novi Sad

Editorial Assistants:
Jovana Vidaković, University of Novi Sad
Ivan Pribela, University of Novi Sad
Davorka Radaković, University of Novi Sad
Slavica Kordić, University of Novi Sad
Srđan Škrbić, University of Novi Sad

Editorial Board:
A. Badica, *University of Craiova, Romania*
C. Badica, *University of Craiova, Romania*
M. Bajec, *University of Ljubljana, Slovenia*
L. Bellatreche, *ISAE-ENSMA, France*
I. Berković, *University of Novi Sad, Serbia*
D. Bojić, *University of Belgrade, Serbia*
Z. Bosnic, *University of Ljubljana, Slovenia*
D. Brđanin, *University of Banja Luka, Bosnia and Hercegovina*
R. Chbeir, *University Pau and Pays Adour, France*
M-Y. Chen, *National Cheng Kung University, Tainan, Taiwan*
C. Chesñevar, *Universidad Nacional del Sur, Bahía Blanca, Argentina*
W. Dai, *Fudan University Shanghai, China*
P. Delias, *International Hellenic University, Kavala University, Greece*
B. Delibašić, *University of Belgrade, Serbia*
G. Devedžić, *University of Kragujevac, Serbia*
J. Eder, *Alpen-Adria-Universität Klagenfurt, Austria*
Y. Fan, *Communication University of China*
V. Filipović, *University of Belgrade, Serbia*
T. Galinac Grbac, *Juraj Dobrila University of Pula, Croatia*
H. Gao, *Shanghai University, China*
M. Gušev, *Ss. Cyril and Methodius University Skopje, North Macedonia'*
D. Han, *Shanghai Maritime University, China*
M. Heričko, *University of Maribor, Slovenia*
M. Holbl, University of Maribor, Slovenia
L. Jain, *University of Canberra, Australia*
D. Janković, *University of Niš, Serbia*
J. Janousek, *Czech Technical University, Czech Republic*
G. Jezic, *University of Zagreb, Croatia*
G. Kardas, *Ege University International Computer Institute, Izmir, Turkey*
Lj. Kašćelan, *University of Montenegro, Montenegro*
P. Kefalas, *City College, Thessaloniki, Greece*
M-K. Khan, *King Saud University, Saudi Arabia*
S-W. Kim, *Hanyang University , Seoul, Korea*
M. Kirikova, *Riga Technical University, Latvia*
A. Klašnja Milićević, *University of Novi Sad, Serbia*
J. Kratica, *Institute of Mathematics SANU, Serbia*
K-C. Li, *Providence University, Taiwan*
M. Lujak, *University Rey Juan Carlos, Madrid, Spain*
JM. Machado, *School of Engineering, University of Minho, Portugal*
Z. Maamar, *Zayed University, UAE*
Y. Manolopoulos, *Aristotle University of Thessaloniki, Greece*
M. Mernik, *University of Maribor, Slovenia*
B. Milašinović, *University of Zagreb, Croatia*
A. Mishev, *Ss. Cyril and Methodius University Skopje, North Macedonia*
N. Mitić, *University of Belgrade, Serbia*
N-T. Nguyen, *Wroclaw University of Science and Technology, Poland*
P Novais, *University of Minho, Portugal*
B. Novikov, *St Petersburg University, Russia*
M. Paprzicky, *Polish Academy of Sciences, Poland*
P. Peris-Lopez, *University Carlos III of Madrid, Spain*
J. Protić, *University of Belgrade, Serbia*
M. Racković, *University of Novi Sad, Serbia*
M. Radovanović, *University of Novi Sad, Serbia*
P. Rajković, *University of Nis, Serbia*
O. Romero, *Universitat Politècnica de Catalunya, Barcelona, Spain*
C, Savaglio, *ICAR-CNR, Italy*
H. Shen, *Sun Yat-sen University, China*
J. Sierra, *Universidad Complutense de Madrid, Spain*
B. Stantic, *Griffith University, Australia*
H. Tian, *Griffith University, Australia*
N. Tomašev, *Google, London*
G. Trajčevski, *Northwestern University, Illinois, USA*
G. Velinov, *Ss. Cyril and Methodius University Skopje, North Macedonia*
L. Wang, *Nanyang Technological University, Singapore*
F. Xia, *Dalian University of Technology, China*
S. Xinogalos, *University of Macedonia, Thessaloniki, Greece*
S. Yin, *Software College, Shenyang Normal University, China*
K. Zdravkova, *Ss. Cyril and Methodius University Skopje, North Macedonia*
J. Zdravković, *Stockholm University, Sweden*

ComSIS Editorial Office:
University of Novi Sad, Faculty of Sciences,
Department of Mathematics and Informatics
Trg Dositeja Obradovića 4, 21000 Novi Sad, Serbia
Phone: +381 21 458 888; **Fax:** +381 21 6350 458
www.comsis.org; Email: comsis@uns.ac.rs

Volume 21, Number 4, 2024
Novi Sad

Computer Science and Information Systems

ISSN: 2406-1018 (Online)

The ComSIS journal is sponsored by:

Ministry of Education, Science and Technological Development of the Republic of Serbia
http://www.mpn.gov.rs/



Computer Science and Information Systems

AIMS AND SCOPE

Computer Science and Information Systems (ComSIS) is an international refereed journal, published in Serbia. The objective of ComSIS is to communicate important research and development results in the areas of computer science, software engineering, and information systems.

We publish original papers of lasting value covering both theoretical foundations of computer science and commercial, industrial, or educational aspects that provide new insights into design and implementation of software and information systems. In addition to wide-scope regular issues, ComSIS also includes special issues covering specific topics in all areas of computer science and information systems.

ComSIS publishes invited and regular papers in English. Papers that pass a strict reviewing procedure are accepted for publishing. ComSIS is published semiannually.

Indexing Information

ComSIS is covered or selected for coverage in the following:

- Science Citation Index (also known as SciSearch) and Journal Citation Reports / Science Edition by Thomson Reuters, with 2023 two-year impact factor 1.2,
- Computer Science Bibliography, University of Trier (DBLP),
- EMBASE (Elsevier),
- Scopus (Elsevier),
- Summon (Serials Solutions),
- EBSCO bibliographic databases,
- IET bibliographic database Inspec,
- FIZ Karlsruhe bibliographic database io-port,
- Index of Information Systems Journals (Deakin University, Australia),
- Directory of Open Access Journals (DOAJ),
- Google Scholar,
- Journal Bibliometric Report of the Center for Evaluation in Education and Science (CEON/CEES) in cooperation with the National Library of Serbia, for the Serbian Ministry of Education and Science,
- Serbian Citation Index (SCIndeks),
- doiSerbia.

Information for Contributors

The Editors will be pleased to receive contributions from all parts of the world. An electronic version (LaTeX), or three hard-copies of the manuscript written in English, intended for publication and prepared as described in "Manuscript Requirements" (which may be downloaded from http://www.comsis.org), along with a cover letter containing the corresponding author's details should be sent to official journal e-mail.

Criteria for Acceptance

Criteria for acceptance will be appropriateness to the field of Journal, as described in the Aims and Scope, taking into account the merit of the content and presentation. The number of pages of submitted articles is limited to 20 (using the appropriate LaTeX template).

Manuscripts will be refereed in the manner customary with scientific journals before being accepted for publication.

Copyright and Use Agreement

All authors are requested to sign the "Transfer of Copyright" agreement before the paper may be published. The copyright transfer covers the exclusive rights to reproduce and distribute the paper, including reprints, photographic reproductions, microform, electronic form, or any other reproductions of similar nature and translations. Authors are responsible for obtaining from the copyright holder permission to reproduce the paper or any part of it, for which copyright exists.

Computer Science and Information Systems

Volume 21, Number 4, September 2024

CONTENTS

Editorial

Guest Editorial: Explainable and trustworthy methods for next-generation artificial intelligence for Reasonable Data Size

Guest Editorial: Recent Advances in AI Methods for Image Processing: Theory, Algorithms, and Applications

Guest Editorial: Selected papers from the European Conference on Advances in Databases and Information Systems

Papers

1221 **Threshold Segmentation Based on Information Fusion for Object Shadow Detection in Remote Sensing Images**
Shoulin Yin, Liguo Wang, Lin Teng

1243 **Tourist services management through clients scoring using a bio-inspired agent architecture**
Raúl Moreno, Alberto Fernández-Isabel, Víctor Aceña, Isaac Martínde Diego, Javier M. Moguerza

1269 **PTB-FLA Development Paradigm Adaptation for ChatGPT**
Miroslav Popovic, Marko Popovic, Ivan Kastelan, Miodrag Djukic, Ilija Basicevic

1293 **TRL-PROTAC: A pre-trained generator of PROTACs targeting specific proteins optimized by reinforcement learning**
Yuhao Dai, Fei Zhu

1321 **Psychological Effect Computation of Courtroom Arguments: A Deep Learning Approach of EEG Signal Data**
Xuan Zhou, Yaming Liu, Baoqian Jiao, Hanzhen Ouyang, Weihui Dai

1335 **A Method for Solving Reconfiguration Blueprints Based on Multi-Agent Reinforcement Learning**
Jing Cheng, Wen Tan, Guangzhe Lv, Guodong Li, Wentao Zhang, Zihao Liu

1359 **Biometric Lock with Facial Recognition Implemented with Deep Learning Techniques**
José Misael Burruel-Zazueta, Héctor Rodríguez-Rangel, Gloria Ekaterine Peralta-Peñuñuri, Vicenç Puig Cayuela, Ignacio Algredo-Badillo, Luis Alberto Morales-Rosales

1389 **BLSAE-SNIDS: A Bi-LSTM Sparse Autoencoder Framework for Satellite Network Intrusion Detection**
Shi Shuxin, Han Bing, Wu Zhongdai, Han Dezhi, Wu Huafeng, Mei Xiaojun

1411 **Sentiment Polarity Analysis of Love Letters: Evaluation of TextBlob, Vader, Flair, and Hugging Face Transformer**
Gaganpreet Kaur, Amandeep Kaur, Meenu Khurana, Robertas Damaševičius

Special section: Explainable and trustworthy methods for next-generation artificial intelligence for Reasonable Data Size

1435 **Advancing Crack Segmentation Detection: Introducing AAMC-Net Algorithm for Image Crack Analysis**
Wang Xiaofang, Liu Chenfang, Hou Junliang, Zhou Liang

1457 **Enhancing Architectural Image Processing: A Novel 2D to 3D Algorithm Using Improved Convolutional Neural Networks**
Qianying Zou, Fengyu Liu, Yuan Liao

1483 **Learning Discriminative Representations through an Attention Mechanism for Image-based Person Re-identification**
Jing Liu, Guoqing Zhou

1499 **Semantic Feature-Based Test Selection for Deep Neural Networks: A Frequency Domain Perspective**
Zhouxian Jiang, Honghui Li, Xuetao Tian, Rui Wang

1523 **AI Large Models Bring Great Opportunities to Reusable Design of CAD Software**
Yunlei Sun, Bingyi Yan, Zhaotong Shao

1547 **A Hybrid GA-Powell Algorithm for Geometric Constraint Solving**
Sun Yunlei, Li Yucong

1567 **ALFormer: Attribute Localization Transformer in Pedestrian Attribute Recognition**
Yuxin Liu, Mingzhe Wang, Chao Li, Shuoyan Liu

Special section: Recent Advances in AI Methods for Image Processing: Theory, Algorithms, and Applications

1583 **GCN-LSTM: Multi-label educational emotion prediction based on graph Convolutional network and long and short term memory network fusion label correlation in online social networks**
Zhiguang Liu, Fengshuai Li, Guoyin Hao, Xiaoqing He, Yuanheng Zhang

1607 **A multi-feature Fusion Model Based on Long and Short Term Memory Network and Improved Artificial Bee Colony Algorithm for English Text Classification**
Tianying Wen

1629 **A Novel Industrial Big Data Fusion Method Based on Q-learning and Cascade Classifier**
Xi Zhang, Jiyue Wang, Ying Huang, Feiyue Zhu

1651 **Remora Optimization Algorithm-based Adaptive Fusion via Ant Colony Optimization for Traveling Salesman Problem**
Lin Piao

1673 **GAN-DNADE: Image Encryption Algorithm Based on Generative Adversarial Network and DNA Dynamic Encoding**
Xi Wang

1699 **Multi-frame Network Feature Fusion Model and Self-attention Mechanism for Vehicle Lane Line Detection**
Guang Zhu, Yajuan Liu, Jiyue Wang

1725 **Stacked Denoised Auto-encoding Network-based Kernel Principal Component Analysis for Cyber Physical Systems Intrusion Detection in Business Management**
Zhihao Song

1745 **A Novel Image Enhancement Method Using Retinex-based Illumination Map Weighted Guided Filtering**
Su Chen, Dahai Li

1765 **Underwater Image Denoising Based on CurvedWave Filtering and Two-dimensional Variational Mode Decomposition**
Lin Teng, Yulong Qiao, Shoulin Yin

1783 **A Novel Deep Fully Convolutional Encoder-Decoder Network and Similarity Analysis for English Education Text Event Clustering Analysis**
Zhenping Jing

1801 **Attitude Estimation of Aircraft Based on Quaternion SRCKF-SLAM Algorithm**
Dandan Wang, Zhaokun Zhu, Liang Yu, Hongjie Li, Kaituo Tan

1823 **Spatio-Temporal-based Multi-level Aggregation Network for Physical Action Recognition**
Yuhang Wang

1845 **Multi-object Real-time Tracking for Intelligent Breeding of Animal**
Fei Wang, Bin Xia, Liwu Pan

1865 **MFE-Transformer: Adaptive English Text Named Entity Recognition Method Based on Multi-feature Extraction and Transformer**
Liuxin Gao

Special section: Selected papers from the European Conference on Advances in Databases and Information Systems

1887 **Identifying Occurrences of the Cnidarian Physalia physalis in Social Media Data**
Heloisa F. Rocha, Lorena S. Nascimento, Leonardo Camargo, Mauricio A. Noernberg, Aurora T. Ramirez Pozo, Carmem S. Hara

1913 **Automatic Conceptual Database Design based on Heterogeneous Source Artifacts**
Goran Banjac, Drazen Brdjanin, Danijela Banjac

1963 Maritime Trajectory Mining: An Automatic Zones of Interests Discovery and Annotation Framework
Omar Ghannou, Etienne Thuillier, Omar Boucelma

Computer Science and Information Systems 21(4): i–ii https://doi.org/10.2298/CSIS240400iI

Editorial

Mirjana Ivanović, Miloš Radovanović, and Vladimir Kurbalija

University of Novi Sad, Faculty of Sciences
Novi Sad, Serbia
{mira,radacha,kurba}@dmi.uns.ac.rs

The fourth and final issue of Volume 21 of Computer Science and Information Systems is comprised of 9 regular articles and three special sections: "Explainable and trustworthy methods for next-generation artificial intelligence for Reasonable Data Size" (7 articles), "Recent Advances in AI Methods for Image Processing: Theory, Algorithms, and Applications" (14 papers) and "Selected papers from the European Conference on Advances in Databases and Information Systems" (3 articles).

The first regular article, "Threshold Segmentation Based on Information Fusion for Object Shadow Detection in Remote Sensing Images" by Shoulin Yin et al., propose a novel threshold segmentation method where object shadow pixels are first screened using intensity and chromaticity information in HSI color space. Then, principal component analysis (PCA) is applied to the remote sensing image to obtain the first principal component, after which a new shadow index is constructed by fusing the results obtained from HSI and the first principal component. Finally, a threshold segmentation model is established using an improved threshold segmentation algorithm.

In the second regular article, "Tourist Services Management through Clients Scoring Using a Bio-Inspired Agent Architecture," Raúl Moreno et al. present a complete framework, Pharaoh, which can make recommendations to customers covering the final user perspective and provide support to travel agents, by filtering the best travel, accommodation, and activity options according to the desires of the customers. This is achieved through effective use of a bio-inspired multi-agent system (MAS).

"PTB-FLA Development Paradigm Adaptation for ChatGPT," by Miroslav Popovic et al. adapts the original four-stage Python testbed for federated learning algorithms (PTB-FLA) paradigm into two new paradigms for ChatGPT, denoted the adapted four-phases paradigm and the adapted two-phases paradigm. The four-phase paradigm consists of producing the sequential code, the federated code, the federated code with callbacks, and the PTB-FLA code, respectively, while the two-phase paradigm produces the sequential and the PTB-FLA code.

Yuhao Dai and Fei Zhu, in "TRL-PROTAC: A Pre-Trained Generator of PROTACs Targeting Specific Proteins Optimized by Reinforcement Learning," propose an integrated protocol for design and evaluation of proteolysis-targeting chimeras (PROTACs) targeting specific proteins, called TRL-PROTAC. TRL-PROTAC is focused on the de novo design of complete PROTACs by effectively joining the designed ligands targeting the proteins of interest (POI) with linkers.

The article "Psychological Effect Computation of Courtroom Arguments: A Deep Learning Approach of EEG Signal Data," authored by Xuan Zhou et al., explores a neural reaction observation method for psychological effect analysis of attorney's speeches in courtroom scenarios. A corpus of courtroom arguments from legal movies and television series were used as source material, and participants' psychological responses to

the speeches were monitored with wearable electroencephalography (EEG) devices. Deep learning approaches based on convolutional neural networks (CNN) and long short-term memory (LSTM) networks were then employed to model attention intensity, cognitive load, and emotional changes.

In their article entitled "A Method for Solving Reconfiguration Blueprints Based on Multi-Agent Reinforcement Learning," Jing Cheng et al. propose an improved value-decomposition network (VDN) method based on average sequential cumulative reward. By refining the characteristics of the integrated modular avionics system, mathematical models are developed for both the system and the reconfiguration blueprint. The improved VDN algorithm demonstrates superior convergence characteristics and optimization effects compared to traditional reinforcement learning algorithms such as Q-learning, deep Q-learning network (DQN), and VDN.

"Biometric Lock with Facial Recognition Implemented with Deep Learning Techniques," by José Misael Burruel-Zazueta et al. develops a biometric lock system (BLS) whose opening is performed by recognizing a person's face, which solves the challenges of re-training, antispoofing, real-time response, and executes inside an embedded system. The proposed BLS overcomes these challenges using a pre-trained FaceNet network for feature extraction and coding, and incorporating ResNet18 + remote photoplethysmography (rPPG) to avoid spoofing.

Shi Shuxin et al., in their article "BLSAE-SNIDS: A Bi-LSTM Sparse Autoencoder Framework for Satellite Network Intrusion Detection," present a satellite network intrusion detection system named Bi-LSTM sparse self encoder (BLSAE-SNIDS) to address various challenges in intrusion detection for satellite networks. Through the development of an innovative unsupervised training Bi-LSTM stacked self-encoder, BLSAE-SNIDS facilitates feature extraction from satellite network traffic, diminishes dimensionality, considerably reduces training and testing durations, and enhances attack prediction accuracy of the classifier.

Finally, "Sentiment Polarity Analysis of Love Letters: Evaluation of TextBlob, Vader, Flair, and Hugging Face Transformer," by Gaganpreet Kaur et al. investigates the performance and accuracy of four popular Python libraries for sentiment analysis (TextBlob, Vader, Flair, and Hugging Face Transformer) in determining the polarity and intensity of sentiments in love letters. A corpus of 300 love letters was collected and randomly sampled to provide 500 sentences for analysis, with human experts evaluating the quality and accuracy of the sentiment annotations.

In the name of our editorial team, we extend our sincerest gratitude to everybody who contributed to the pub- lication of ComSIS, including (but not limited to) our authors, reviewers, guest editors, colleagues, team members, and boards.

Computer Science and Information Systems 21(4): iii–iv https://doi.org/10.2298/CSIS240400iiiZ

Guest Editorial: Explainable and trustworthy methods for next-generation artificial intelligence for Reasonable Data Size

Dalin Zhang[1] and Ivan Luković[2]

[1] Beijing Jiaotong University, China
dalin@bjtu.edu.cn
[2] University of Belgrade, Serbia
ivan.lukovic@fon.bg.ac.rs

This special issue showcases cutting-edge research in artificial intelligence and computer vision, with a focus on novel deep learning architectures and algorithms for various recognition and analysis tasks. The collected papers present innovative approaches that push the boundaries of performance and efficiency across applications ranging from infrastructure inspection to pedestrian analysis.

In the paper "Advancing Crack Segmentation Detection: Introducing AAMC-Net Algorithm for Image Crack Analysis", the authors introduce AAMC-Net, an innovative convolutional neural network algorithm that leverages a mixed domain attention mechanism to enhance feature extraction for road crack segmentation. The proposed approach demonstrates superior performance in crack detail processing, model training efficiency, and real-time detection compared to several traditional algorithms, achieving notable improvements across multiple evaluation metrics. By addressing challenges in road crack segmentation while maintaining flexibility for broader image segmentation tasks, AAMC-Net represents a significant advancement in computer vision techniques for infrastructure maintenance and safety applications.

The paper titled "Enhancing Architectural Image Processing: A Novel 2D to 3D Algorithm Using Improved Convolutional Neural Networks" introduces an innovative architectural image processing model that combines Restricted Boltzmann Machines (RBMs) with Convolutional Neural Networks (CNNs) to convert 2D architectural images into 3D models. The proposed approach demonstrates superior performance in terms of accuracy and efficiency, achieving a 97% detection rate and significantly outperforming comparative algorithms in runtime and efficiency. By integrating advanced preprocessing techniques with a novel deep learning architecture, this research makes a valuable contribution to the field of architectural digitization and smart city development.

In the paper "Attention Mechanism for Image-based Person Re-identification", Liu and Zhou introduce a novel hybrid framework called PGAN (Patch-Global Attention Network) for person re-identification, which combines a patch-network and global-network with a channel-wise attention mechanism to learn discriminative feature representations. The proposed approach incorporates CSwin Transformer to re-extract features from residual blocks and demonstrates superior performance on benchmark datasets, achieving state-of-the-art results with mAP scores of 91.3%, 83.4%, and 81.8% on Market-1501, DukeMTMC-ReID, and CUHK03-NP datasets respectively. By effectively integrating local and global features through attention mechanisms and transformer architectures, this work makes a significant contribution to improving the accuracy and efficiency of person re-identification systems.

In the paper "Semantic Feature-Based Test Selection for Deep Neural Networks: A Frequency Domain Perspective", Jiang et al. introduce SaFeTS, an innovative test selection method for deep neural networks that leverages frequency domain analysis to extract semantic features from test cases. By clustering these features and sampling diverse test cases, SaFeTS demonstrates superior performance in exposing varied model errors and improving adversarial and out-of-distribution robustness compared to baseline methods. The authors' approach offers a fresh perspective on enhancing DNN testing and robustness by focusing on semantic diversity, with experiments showing up to 20% improvement in adversarial accuracy over existing techniques.

In their paper "AI Large Models Bring Great Opportunities to Reusable Design of CAD Software", Sun et al. explore how AI large models can enhance reusable design in CAD software by improving rule-based and case-based reasoning techniques. The authors propose a novel framework that integrates large language models, knowledge graphs, and databases to enable more intelligent and interpretable design reuse. By leveraging the advanced language understanding and reasoning capabilities of large AI models, this approach aims to overcome limitations of traditional methods and open new possibilities for computer-aided engineering augmented by artificial intelligence.

The paper titled "A Hybrid GA-Powell Algorithm for Geometric Constraint Solving" introduces a novel hybrid GA-Powell algorithm to address limitations in traditional geometric constraint solving methods. By combining the global search capabilities of genetic algorithms with the local refinement strengths of Powell's method, the proposed approach demonstrates superior accuracy and efficiency in solving multi-solution problems compared to existing techniques. The researchers' findings offer valuable insights for enhancing CAD systems and geometric constraint solvers, potentially advancing the field of computer-aided engineering design.

In the paper "ALFormer: Attribute Localization Transformer in Pedestrian Attribute Recognition", ALFormer, a novel transformer-based framework for pedestrian attribute recognition that addresses limitations in accurately localizing discriminative regions has been introduced. By incorporating Mask Contrast Learning to suppress misleading regional correlations and Attribute Spatial Memory to capture inherent attribute locations, ALFormer achieves state-of-the-art performance on benchmark datasets. The authors' approach demonstrates the potential of leveraging spatial consistency to enhance fine-grained recognition tasks in computer vision.

In conclusion, this special issue highlights several promising directions for advancing AI and computer vision capabilities. The presented works demonstrate the potential of hybrid architectures, attention mechanisms, and large language models to address key challenges in tasks like image segmentation, 3D modeling, person re-identification, and attribute recognition. By combining global and local feature extraction, leveraging spatial consistency, and enhancing robustness through semantic diversity, these approaches offer new paradigms for developing more accurate, efficient and interpretable AI systems. As the field continues to evolve rapidly, the novel techniques introduced here provide valuable insights to guide future research and development of intelligent visual analysis systems. We hope you enjoy reading the papers as well as we did.

Computer Science and Information Systems 21(4): v–viii https://doi.org/10.2298/CSIS240400vY

Guest Editorial: Recent Advances in AI Methods for Image Processing: Theory, Algorithms, and Applications

Shoulin Yin[1] and Mirjana Ivanović[2]

[1] Shenyang Normal University, China
yslin@hit.edu.cn
[2] Faculty of Sciences, University of Novi Sad, Serbia
mira@dmi.uns.ac.rs

The special section Recent Advances in AI Methods for Image Processing: Theory, Algorithms, and Applications is to contribute original researches on recent theoretical advances, algorithmic developments, and practical applications of the AI-based frameworks. Topics include new AI models, adaptive AI algorithms, robustness analysis, stability analysis, and imaging processing applications.

The rise of machine learning approaches, and in particular deep learning, has led to a significant increase in the performance of AI-based systems. Many current image applications also include at least one AI-based component. Images, video, sound, text-traditional multimedia data are now completed by a multitude of multimodal sources, on which the fundamental principles of image research are being applied.

For this special issue 58 submissions have been submitted where the corresponding authors were majorly counted by the deadline for manuscript submission with an open call-for-paper period of 6 months. All these submissions are considered significant in the field, but however, only one-third of them passed the pre-screening procedure by guest editors. The qualified papers then went through double-blinded peer review based on a strict and rigorous review policy. After a totally three-round reviews, 14 papers were accepted for publication. A quick overview to the papers in this section can be revealed below, and we expect the content may draw attentions from public readers, and furthermore, prompt the society development.

The first paper "GCN-LSTM: Multi-label educational emotion prediction based on graph Convolutional network and long and short term memory network fusion label correlation in online social networks" by Zhiguang Liu, Fengshuai Li, Guoyin Hao, Xiaoqing He, and Yuanheng Zhang, proposes a novel multi-label educational emotion prediction based on graph convolutional network and long and short term memory network fusion label correlation in online social networks. This model uses Word2Vec method to train word vectors and combines graph convolutional neural network (GCN) with long and short term memory network (LSTM). This paper aims to improve the accuracy of multi-label classification and to effectively learn the possible correlations between each label.

The second paper, "A multi-feature Fusion Model Based on Long and Short Term Memory Network and Improved Artificial Bee Colony Algorithm for English Text Classification" by Tianying Wen, proposes a multi-feature fusion model based on long and short term memory network and improved artificial bee colony algorithm for English text classification. In this method, the character-level vector and word-level vector representations of English text are calculated using a pre-training model to obtain a more comprehensive text feature vector representation. Then the multi-head attention mechanism is used to

capture the dependencies in the text sequence to improve the semantic understanding of the text.

The third paper, "A Novel Industrial Big Data Fusion Method Based on Q-learning and Cascade Classifier" by Xi Zhang, Jiyue Wang, Ying Huang, and Feiyue Zhu presents an improved Q-learning algorithm, which makes the improved algorithm randomly select actions in the early stage, and dynamically change in the late stage in the random selection of actions and actions with the highest reward value. It effectively improves the defects of traditional Q-learning algorithm that it is easy to fall into the local optimal and has slow convergence speed.

"Remora Optimization Algorithm-based Adaptive Fusion via Ant Colony Optimization for Traveling Salesman Problem" by Lin Piao, proposes a novel remora optimization algorithm-based adaptive fusion via ant colony optimization for solving TSP to enhance the local search ability of ACO, speed up the efficiency of ACO and avoid the premature problem. In addition, the initial solution is optimized by the k-opt strategy. Finally, the random recombination strategy is used to recombine the pheromone and random excitation to make the algorithm jump out of the local optimal as far as possible and improve the accuracy of the algorithm.

The fifth paper, "GAN-DNADE: Image Encryption Algorithm Based on Generative Adversarial Network and DNA Dynamic Encoding" by Xi Wang introduces GAN into random key generation and uses GAN to learn and train the random key generated by hyperchaotic system. A parallel chaotic system is used to generate two sets of pseudo-random sequences, and DNA dynamic encoding is introduced to further transform them to generate a new sequence. The pixel-level diffusion and scrambling of images within and between channels are carried out by using random sequences. The experimental results show that the randomness of GAN can significantly expand the key space, and the proposed algorithm has significant advantages in the security and anti-attack ability of ciphertext images.

Next paper, "Multi-frame Network Feature Fusion Model and Self-attention Mechanism for Vehicle Lane Line Detection" by Guang Zhu, Yajuan Liu, Jiyue Wang, proposes a lane parallel detection network based on multi-frame network feature fusion model and self-attention mechanism according to the scene characteristics that vehicles can obtain continuous images during normal driving. Firstly, a parallel feature extraction structure is designed. On the one hand, a single frame network with high precision is used to extract the features of the current frame. On the other hand, a lightweight multi-frame network is designed to extract features of low-resolution multi-frame temporal images. And the recurrent neural network module is used to fuse the extracted temporal features and obtain multi-frame features. Self-attention mechanism can effectively capture the relevant information of internal features. Then the fusion module of single frame feature, multi-frame feature and self-attention feature is designed. The feature map of lane line is output by up-sampling network. The experimental results show that the network in this paper has significant improvement in both objective detection accuracy and subjective effect compared with other methods.

The seventh paper, "Stacked Denoised Auto-encoding Network-based Kernel Principal Component Analysis for Cyber Physical Systems Intrusion Detection in Business Management" by Zhihao Song, proposes a stacked denoised auto-encoding network based on kernel principal component analysis for industrial cyber physical systems intrusion

detection. A multi-stacked denoised auto-encoding network model is used to classify and identify the data after dimensionality reduction by voting. Experimental results show that the proposed method has better classification performance and detection efficiency by comparing the state-of-the-art intrusion detection methods.

The eighth paper, "A Novel Image Enhancement Method Using Retinex-based Illumination Map Weighted Guided Filtering" by Su Chen, Dahai Li) try to solve the traditional defogging physical models. These models only enhance the images based on dark channels prior, resulting in different depths of field in local areas, and it can lead to some problems such as image overexposure and halo artifacts. To solve this problem, the method of combining light and dark channels is adopted to calculate the atmospheric light value and transmittance. For the problem that edge information is easily lost, the illumination gradient domain weighted guided filtering is utilized to improve the thinning transmittance. Experimental results with the proposed method have obvious improvement in denoising, halo elimination, brightness adjustment and edge preservation in the low-illumination image under different conditions.

Paper, "Underwater Image Denoising Based on Curved Wave Filtering and Two-dimensional Variational Mode Decomposition" by Lin Teng, Yulong Qiao, Shoulin Yin, proposes a novel underwater image denoising method based on curved wave filter and two-dimensional variational mode decomposition. Firstly, the noisy image is decomposed by two-dimensional variational mode decomposition, and a series of modal components with different center frequencies are obtained. The effective modal components are selected by correlation coefficient and structural similarity. And the effective modal components are processed by the curve-wave filter.

The tenth paper, "A Novel Deep Fully Convolutional Encoder-Decoder Network and Similarity Analysis for English Education Text Event Clustering Analysis" by Zhenping Jing, proposes a novel deep fully convolutional encoder-decoder network and similarity analysis for English education text event clustering analysis in online social networks. At the encoder end, the features of text events are extracted step by step through the convolution operation of the convolution layer. Compared to the supervised cluster model with the same pre-training, the clustering index of the proposed model is improved.

Paper, "Attitude Estimation of aircraft Based on Quaternion SRCKF-SLAM Algorithm" by Dandan Wang, Zhaokun Zhu, Liang Yu, Hongjie Li, and Kaituo Tan, takes the attitude quaternion error and the gyro drift error as the state quantity, and measures the attitude quaternion of SINS/SLAM navigation. The square root cubature Kalman filter algorithm is used for pose estimation, which not only solves the standardization problem of traditional quaternion, but also reduces the state dimension and complexity of the square root UKF algorithm of traditional quaternion, and improves the numerical stability.

The twelfth paper, "Spatio-Temporal-based Multi-level Aggregation Network for Physical Action Recognition" by Yuhang Wang, introduces spatio-temporal-based multi-level aggregation network (ST-MANet) for action recognition. It utilizes the correlations between different spatial positions and the correlations between different temporal positions on the feature map to explore long-range spatial and temporal dependencies, respectively, generating the spatial and temporal attention map that assigns different weights to features at different spatial and temporal locations. Additionally, a multi-scale approach is introduced, proposing a multi-scale behavior recognition framework that models various visual rhythms while capturing multi-scale spatiotemporal information. A spatial diversity

constraint is then proposed, encouraging spatial attention maps at different scales to focus on distinct areas. This ensures a greater emphasis on spatial information unique to each scale, thereby incorporating more diverse spatial information into multi-scale features.

The thirteenth paper, "Multi-object Real-time Tracking for Intelligent Breeding of Animal" by Fei Wang, Bin Xia, Liwu Pan, proposes a new multi-object real-time tracking within deep framework for intelligent breeding of animal (MRT-IB), which consists of semantic feature extraction module, center point prediction module, and object and trajectory calibration module. MRT-IB reduces the difficulty of modeling animal trajectories by performing animal detection on consecutive frames, resulting in higher robustness in real farming scenarios compared to traditional multi-object tracking schemes that directly model animal motion trajectories.

The last paper, "MFE-Transformer: Adaptive English Text Named Entity Recognition Method Based on Multi-feature Extraction and Transformer" by Liuxin Gao, proposes an adaptive English named text entity recognition method based on multi-feature extraction and transformer. Experiments are carried out on five English datasets and CrossNER datasets, and compared with current cross-domain methods, the results show that the proposed method achieves better results in cross-domain tasks, indicating that learning and using structured knowledge can better promote cross-domain knowledge transfer.

The first guest editor would like to thank Prof. Mirjana Ivanović, the editor-in chief of ComSIS, for her support during the preparation of this special section in the journal.

Finally, we gratefully acknowledge all the hard work and enthusiasm of authors and reviewers, without whom the special section would not have been possible.

Computer Science and Information Systems 21(4): ix–ix https://doi.org/10.2298/CSIS240400ixB

Guest Editorial: Selected papers from the European Conference on Advances in Databases and Information Systems

Francesca Bugiotti, Johann Gamper, Genoveva Vargas-Solar, and Ester Zumpano

This special guest section publishes extended versions of three selected papers accepted in workshops held in conjunction with the ADBIS 2023: European Conference on Advances in Databases and Information Systems (http://adbis.eu/). The workshops covered diverse topics, such as intelligent data processing, large-scale knowledge graph analysis, modern data engineering, advanced data management, and personalization and recommender systems. The selected papers were chosen based on their technical quality, assessed by the international program committee evaluating workshop papers. They went through a strict evaluation of four reviewers, specialists in the topics of the papers. The following papers were selected for this special guest section of ComSIS:

1. "Identifying Occurrences of the Cnidarian Physalia physalis in Social Media Data", by Heloisa Fernanda Rocha, Leonardo Camargo, Mauricio Almeida Noernberg, Aurora Trinidad Pozo, Carmem Hara, Lorena Silva Nascimento: This paper reports the process of collecting and labeling Instagram posts, based on hashtag searches for improving knowledge of biodiversity. An experimental study shows the effect of unquestioning the adoption of labels given by a specialist compared to labels adapted for machine learning training.
2. " Automatic Conceptual Database Design based on Heterogeneous Source Artifacts", by Goran Banjac, Drazen Brdjanin, Danijela Banjac: This paper presents an early prototype of the tool named DBomnia – the first online web-based tool enabling automatic derivation of conceptual database models from heterogeneous source artifacts (business process models and textual specifications).
3. "Maritime Trajectory Mining: An Automatic Zones of Interests Discovery and Annotation Framework ", by Omar Ghannou: This paper presents an automatic framework for discovering and annotating specific zones of interest within maritime trajectories. The contribution is significant in the discovery of mobility patterns and the improvement of transportation systems.

The Guest Editors extend their sincere gratitude to everyone who contributed to the success of this special issue. We thank the authors for sharing their scientific insights with the community. The authors invested effort in producing the versions published in ComSIS, considering reviewers' suggestions and requests to consolidate the work presented at the conference. We are also profoundly thankful to Prof. Mirjana Ivanovic, the editor-in-chief of Computer Science and Information Systems (ComSIS), for her invaluable guidance and support throughout the process. A special acknowledgement is due to the reviewers' constructive feedback, which significantly enhanced the quality of the papers accepted for this special issue.

Computer Science and Information Systems 21(4):1221–1241 https://doi.org/10.2298/CSIS231230023Y

Threshold Segmentation Based on Information Fusion for Object Shadow Detection in Remote Sensing Images

Shoulin Yin[1], Liguo Wang[2,⋆], and Lin Teng[1]

[1] School of Information and Communication Engineering, Harbin Engineering University
Harbin 150001 China
yslin@hit.edu.cn
[2] College of Information and Communications Engineering, Dalian Minzu University
Dalian, 116600, China
wangliguo@hrbeu.edu.cn

Abstract. In the shadow detection task, the shadow model is usually consistent with the approximate contour of ontology semantics, it is difficult to extract the features of land covered objects or ground pixels, and easy to be confused into foreground objects in gray scale. Therefore, we present to formulate and apply one new threshold segmentation method based on information fusion for object shadow detection in remote sensing images. Firstly, object shadow pixels are screened using intensity and chromaticity information in HSI color space. Secondly, the remote sensing image is carried out by principal component analysis (PCA) to obtain the first principal component. A new shadow index is constructed using the results obtained from HSI and the first principal component. Thirdly, based on the results of the above two information fusion, a threshold segmentation model is established using the improved threshold segmentation algorithm between the maximum and the minimum threshold segmentation algorithm, so as to obtain the final object shadow detection results. Finally, affluent experiments are conducted on the datasets collected from Google Earth. The results show that the proposed object shadow detection algorithm in remote sensing images can achieve better segmentation and detection (more than 95%) effect compared with state-of-the-art methods.

Keywords: object shadow detection, threshold segmentation, information fusion, remote sensing images, HSI color space, PCA.

1. Introduction

With the rapid development of space remote sensing technology, remote sensing image is more and more widely used in urban planning, digital agriculture and resource exploration. Shadow, which reflects the spatial structure characteristics of ground objects, has always been a hot spot in the research of ground objects recognition. According to the length and irradiation Angle of the profile, the height and shape of the ground objects can be calculated, which is conducive to urban construction, remote sensing mapping and other applications [1,2]. However, shadow will also cover the ground objects, resulting in attenuation or loss of the effective information of the ground objects, which brings difficulties to the interpretation of remote sensing image target recognition, feature extraction and image matching. With the continuous development of sensor technology,

⋆ Corresponding author

sub-meter high resolution image has been more widely used in urban planning, road mapping and other fields [3-5]. In high-resolution images, the categories of objects are more precise, and the objects are easily affected by shaded areas, which makes the extraction and recognition of objects difficult. Therefore, further improving the precision and accuracy of shadow detection of high-resolution images has important application value and significance for urban planning and natural disaster assessment.

In the early days, people mainly used color transformation to detect shadow in color remote sensing images. Kumar et al. [6] used color and geometric position of image to extract shadow region. Srikantha et al. [7] carried out shadow extraction through image color region segmentation and image reproduction, combining with dynamic range compression. Zhang et al. [8] designed a shadow detection algorithm by analyzing the difference in pixel brightness values of RGB channels in shaded areas and non-shaded areas and combining the idea of seed filling with difference operators and covariance operators. Using color feature to extract shadow had certain application value, but it was not good for high-resolution image with higher complexity of features.

Because of the different characteristics of the shadow in different bands, the algebraic operation of the band can extract the shadow. Abd-El et al. [9] constructed a normalized blue-red band index to distinguish vegetation and shadow according to the difference in the decline amplitude of radiation brightness of blue and red bands in the shadow area, but this method was difficult to distinguish shadow from red and blue ground objects. Shi et al. [10] calculated the difference ratio of green and blue wave segments and extracted shadows by combining with histogram threshold segmentation of near infrared band. This method could not extract shadows effectively in complex atmospheric and ground conditions. Wang et al. [11] carried out shadow extraction through linear calculation and nonlinear calculation between bands and combined with digital elevation model. This method would cause loss of data accuracy, and the accuracy of shadow extraction was easily affected by image quality.

Ward et al. [12] proposed a radiation model based on direct light and reflected light, and used the difference of direct light and reflected light to extract shadow regions. However, this method required the height and Angle attributes of the sun, which had obvious limitations and high cost, it was not conducive to its implementation. Du et al. [13] proposed a shadow segmentation algorithm that could make use of histogram threshold, but this method had the problem of poor threshold segmentation effect. Fu et al. [14] proposed a shadow detection model using HIS changes, which used shadow index to distinguish shaded areas from non-shaded areas, but this method had a high misjudgment rate.

Therefore, we propose a novel object shadow detection threshold segmentation method based on information fusion in remote sensing images. Our main contributions are as follows.

1. Object shadow pixels are firstly screened using intensity and chromaticity information in HSI color space.
2. The remote sensing image is carried out by principal component analysis (PCA) to obtain the first principal component. A new shadow index is constructed using the results obtained from HSI and the first principal component.
3. Based on the results of the above two information fusion, a threshold segmentation model is established using the improved threshold segmentation algorithm between

the maximum and the minimum threshold segmentation algorithm, so as to obtain the final object shadow detection results.

2. Related Works

At present, some scholars have summarized some shadow detection methods. For example, Hou et al. [15] summarized and classified the existing shadow detection methods of remote sensing images, and put forward the problems of generalization and robustness that needed to be solved in the current shadow detection research. Liu et al. [16] mainly introduced several cloud detection methods for common satellite data. By comparing different shadow detection methods, they pointed out that existing methods had problems such as weak universality and missing judgment and misjudgment for thin and transparent clouds. Mahajan et al. [17] reviewed shadow detection methods from the aspects of classical algorithms and machine learning methods, compared and analyzed the detection accuracy of the two types of methods in different scenarios. Based on a large number of domestic and foreign references, this paper summarized the existing shadow detection methods into three categories: Threshold method, spatial change detection method and machine learning method. Threshold method is a widely used shadow detection method, such as the early shadow coverage extraction of single image [18], pixel level shadow detection for a series of Landsat satellites [19], and the new Sentinel-2 satellite shadow detection. The Fmask (Function of mask) algorithm proposed by Zhu et al. [20] had been integrated into the basic remote sensing camera processing flow of U.S. Geological Survey (USGS). The spatial change detection method used the spatial change information of multi-period images to achieve higher cloud detection accuracy than that of a single image [21]. Machine learning methods include traditional machine learning methods, such as Support Vector Machine (SVM), Artificial Neural Network (ANN), Random Forest (RF), etc. It also includes the most popular methods of deep learning. Among them, traditional machine learning methods are trained through a large number of training sets, and the trained classifier is used to separate shadow pixels and clear sky pixels [22,23]. However, the deep learning method is driven by a large number of marked samples of shadows and adopts an algorithm framework with semantic information extraction and learning such as convolutional neural network to achieve the purpose of shadow detection [24].

2.1. Thresholding Method

The threshold method is to separate the cloud and the ground object by using the different spectral characteristics of the cloud and the underlying surface, and compare the detected value of the pixel with the threshold value to determine whether it is the ground object or the cloud. The key of this method lies in the determination of threshold. In the early stage, most of them are fixed threshold method. Later, dynamic threshold, adaptive threshold and multi-band combination threshold are developed, and different thresholds can be selected according to different conditions. In recent years, the threshold method has been further developed on the basis of the existing development. For example, based on the Fmask (Function of Mask) algorithm, the Fmask 4.0 algorithm [25] is further developed to improve the cloud detection method. The improvement mainly includes three

aspects: First, the integration of auxiliary data, in which the Global Surface Water Occurrence (GSWO) data is used to improve the separation of land and water, and the global digital elevation model is used to standardize the tropical and cirrus zones. Second, a new cloud probability is developed, which uses the cloud probability based on Haze Optimized Transformation (HOT) to replace the temperature probability of Sentinel-2 image, and then corrects the global optimal threshold of cloud probability for different sensors. The Spectral Context-Contextual Snow Index (SCSI) is created to better differentiate snow/ice and clouds in polar regions. Landsat 4-8 and Sentinel-2 data are used for training and verification, and the results show that the overall accuracy of Fmask 4.0 algorithm is higher than that of Fmask algorithm. Different from Fmask 4.0 Algorithm, which used fixed threshold to perform cloud detection, Chang et al. [27] proposed an improved cloud detection algorithm-generating (CDAG) method based on automatic threshold. According to the characteristics of GF-6 WFV data, the CDAG method is improved from the two aspects of adding bright surface pixel database and band combination. The results show that the improved CDAG cloud detection algorithm can achieve better recognition effect in the cloud recognition of GF-6 WFV data. However, due to the lack of short-wave infrared band of GF-6 WFV sensor, the improved CDAG method cannot identify cloud pixels in ice and snow regions. In summary, the threshold method is simple, high computation speed and easy to implement. However, its disadvantage is that the setting of threshold requires more prior knowledge and human participation, and the subjectivity is strong. Moreover, different strategies and thresholds are required for images taken by different sensors, different periods or places, which limits the universality of the method.

In addition to the spectral information, the spatial information of the image can also be used as the feature to set the threshold method for cloud detection. With the continuous improvement of spatial resolution of remote sensing images, cloud texture presents unique characteristics. Fractal dimension and gray co-occurrence matrix can be used to quantitatively express texture features. Fractal dimension is used to describe the complexity of fractals. In remote sensing images, the complexity of ground objects is higher than that of clouds, so the fractal number of ground objects is greater than that of clouds, so as to distinguish the two. The angular second order moments in the gray co-occurrence matrix reflect the texture law. Clouds in the image have larger angular second order moments than ground objects, which can distinguish them. Xu et al. [27] proposed a cloud detection algorithm combining spectral threshold and texture features, and used this algorithm to detect the cloud of hyperspectral remote sensing images captured by the Hyperion sensor of EO-1 satellite, which not only greatly improved the cloud detection accuracy of remote sensing images, but also met the requirements of fast cloud detection of space-borne hyperspectral images. In addition, Calin et al. [28], aiming at cloud detection in infrared images, adopted a method combining fractal dimension and gray co-occurrence matrix to effectively realize cloud detection in infrared images. Due to the complexity of cloud spatial information, cloud detection methods based on cloud texture characteristics were easy to cause a small number of clouds in the image to be missed, resulting in certain errors.

2.2. Spatial Change Detection Method

The presence of cloud on remote sensing image will significantly affect the radiation change of image, which is the basis of cloud detection based on spatial change. The prin-

ciple of spatial change detection method is to detect clouds and shadows by using the temporal characteristic changes of multi-temporal remote sensing images. For example, Zhang et al. [29] proposed the Multi-Temporal Cloud Detection (MTCD) method. The cloud detection results of FORMOSAT-2 and Landsat 5, 7 remote sensing images were better than those of artificial labels. However, MTCD required time series images and had high requirements on images, so its application in some sensors was limited. Zhang et al. [30] proposed the Multi-Temporal mask method, an automatic detection algorithm for cloud/cloud shadow and snow in multi-temporal landsat images. Based on the Fmask method, the time information of cloud-free pixels in multi-temporal remote sensing images was added to improve the cloud detection accuracy. Automatic time-series Analysis (ATSA) used the unsupervised classification method to detect clouds in time series images, and analyzed the geometric relationship between the sun's azimuth Angle, altitude Angle and clouds to detect cloud shadows. In the cloud detection experiment of multi-sensor remote sensing images, the detection effect of ATSA method was superior to that of Fmask. Sorasak etc. [31] presented Multi-temporal Cloud Masking (MCM) method, the method was used in tropical and subtropical environment Landsat 8 cloud/cloud shadow detection. Compared with Landsat 8 Cloud Cover Assessment (L8CCA) proposed by Candra et al. [32], MCM method could achieve higher accuracy. In addition, Qiu et al. [33] developed a cloud detection method with Cirrus cloud mask (Cmask), which used cirrus band (spectrum range 1.36-1.39) to detect cirrus cloud in Landsat 8 time series images [34].

Compared with the threshold method, on the one hand, the spatial change detection method adds time information into the cloud detection process, which makes the method more suitable for cloud detection of time series remote sensing images. On the other hand, because the method relies on time information for detection, the application scenario of this method is limited.

2.3. Temporal Substitution Method

With the rapid development of data fusion technology, cloud removal methods based on multi-phase image replacement also have a certain development. The cloud removal method based on multi-phase image replacement uses cloudless pixels to replace cloudless pixels according to certain principles to achieve the purpose of generating clear sky images. For example, the Closest Spectral Fit (CSF) [35] cloud removal method, which used cloud images and cloud-free images taken at different times in the same region to replace pixels to achieve cloud removal. CSF achieved high accuracy in cloud removal for Landsat and Quick Bird images. Moreover, the accuracy of the method was not affected by the size, thickness and density of the cloud. Similar to the principle of CSF, Liu [36] replaced the cloudless pixel in Radarsat-2 fully polarized SAR image with the least spectral difference from the pixel with cloud region in Landsat 8 OLI false color image by weighted Euclidean distance, and realized cloud noise removal and repair in Landsat 8 remote sensing image. This method could integrate the advantages of microwave and optical images to achieve cloud region restoration of multi-spectral images and achieve a better cloud removal effect. In addition, Huang et al. [37] proposed a method to reconstruct cloud region information by using the correlation of multi-temporal images. First, images at different times were obtained with similar cloud-free patches to reconstruct cloud pollution regions, and then Poisson equation was used to globally optimize the filled cloud

images to improve the radiation inconsistency problem. Li et al. [38] further proposed a joint determination method to replace Poisson equation to reconstruct and optimize cloud pollution images. Compared with the original method, the optimized method performed better in terms of radiation accuracy and consistency. Ji et al. [39] proposed an automatic cloud removal algorithm, which firstly used temporal similarities of multi-temporal Landsat images to reconstruct the cloud polluted part, and then adopted Poisson hybrid algorithm to further eliminate the radiation difference between the cloud-free region and the reconstructed region. The algorithm automatically generated a cloud-free dataset for Landsat 8 from 2013 to 2017, covering the whole of China, and the method could also generate cloud-free time series images for other satellites.

The cloud removal method based on multi-phase image replacement can make full use of existing images to achieve the purpose of cloud removal, but the prerequisite for the application of this method is not only to solve the registration and correction problems between different images, but also to solve the radiation difference between the original image and the corrected image, and to ensure that multiple sensors obtain remote sensing data in the same area, which is very demanding.

3. Proposed Object Shadow Detection Model

Considering the accuracy and real-time detection requirements, this paper presents an object shadow detection threshold segmentation method based on information fusion. The detailed detection structure is shown in Figure 1.

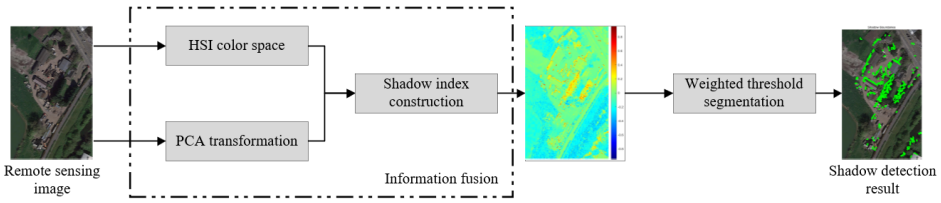


Fig. 1. Proposed detection flowchart

3.1. HSI Color Space Transformation

According to lighting model in reference [40], the brightness of an object mainly comes from ambient reflected light, diffuse reflected light and specular reflected light, which can be expressed as:

$$I = I_a k_a + I_p k_d \cos\theta + I_p k_s \cos^n \varphi. \tag{1}$$

Where I_a is the ambient brightness. k_a is the ambient reflection coefficient of the object surface. I_p is the intensity of direct light source. k_d and k_s are the diffuse and specular reflection coefficients of the object surface to the light source, respectively. θ is

the incidence Angle of the direct light source. φ is the Angle between the mirror reflection direction and the camera observation direction. n is the convergence index of specular reflected light. When the direct light in the scene is partially or completely blocked by the object and the shadow appears, the diffuse reflection and specular reflection of the direct light source in the area will be significantly reduced, so the brightness of the shadow area can be expressed as:

$$I' = I_a k_a + \lambda I_p (k_d \cos\theta + k_s \cos^n \varphi) < I. \tag{2}$$

Where, $0 < \lambda < 1$, whose size is determined by the spatial position of the pixel or area in the scene. Through the above analysis, it can be found that a prominent feature before and after the shadow appears in the background area is that the brightness of pixels in the current frame is lower than that of corresponding pixels in the background, so the shadow can be screened by brightness changes.

The perceived color characteristics of an object depend on the spectral reflection characteristics of the object surface, independent of changes in the scene and light. When the brightness of an object changes, the color information perceived by human eyes is basically unchanged, that is, chroma is independent of brightness. For example, the normalized color feature RGB in rgb space, UV component in YUV space, hue H and saturation S in HSI space, etc. Therefore, it is possible to distinguish the target from the shadow by using the approximately identical chromaticity of the pixel in the shadow area and the corresponding background area.

HSI color space clearly expresses Hue, Saturation and Intensity changes and greatly simplifies the workload of image analysis and processing. Therefore, this paper selects HSI color space for preliminary screening of shadows. According to the above analysis of color characteristics, compared with the corresponding background region, the brightness information of the shadow will change greatly, while the hue H and saturation S remain basically unchanged. According to the above characteristics, shadow pixels can be detected by satisfying equation (3), i.e.,

$$S_1 = \begin{cases} 127 & \alpha_1 \le P_F^i / P_B^i \le \alpha_2 \\ 255 & others \end{cases} \tag{3}$$

Here, $|P_F^h - P_B^h| \le \beta$, $|P_F^s - P_B^s| \le \gamma$. S_1 is the screened shadow region. h, s and i respectively represent hue, saturation and brightness after the normalization of HSI color space. P_F^h, P_F^s, P_F^i and P_B^h, P_B^s, P_B^i represent h, s and i values corresponding to pixel point P in the current detection image F and background image B respectively. α_1, α_2, β and γ are the shadow discrimination thresholds of each color component. Threshold setting is to select the weak threshold from the optimal threshold range obtained by manual experiment, so as to screen out the shadow area more adequately.

3.2. Principal Component Normalization Method

Principal component analysis method can reduce the dimension of data and reduce the redundancy of data [41,42]. For different remote sensing images, different bands can be selected for principal component analysis. Different eigenvalues can be obtained by principal component analysis of different wavebands. The larger eigenvalue denotes the better corresponding eigenvector. For remote sensing images, since the variance contribution

rate of the first principal component is usually above 90%, the first principal component image can represent most of the information of the original image. The original image and the first principal component image are shown in Figure 2.

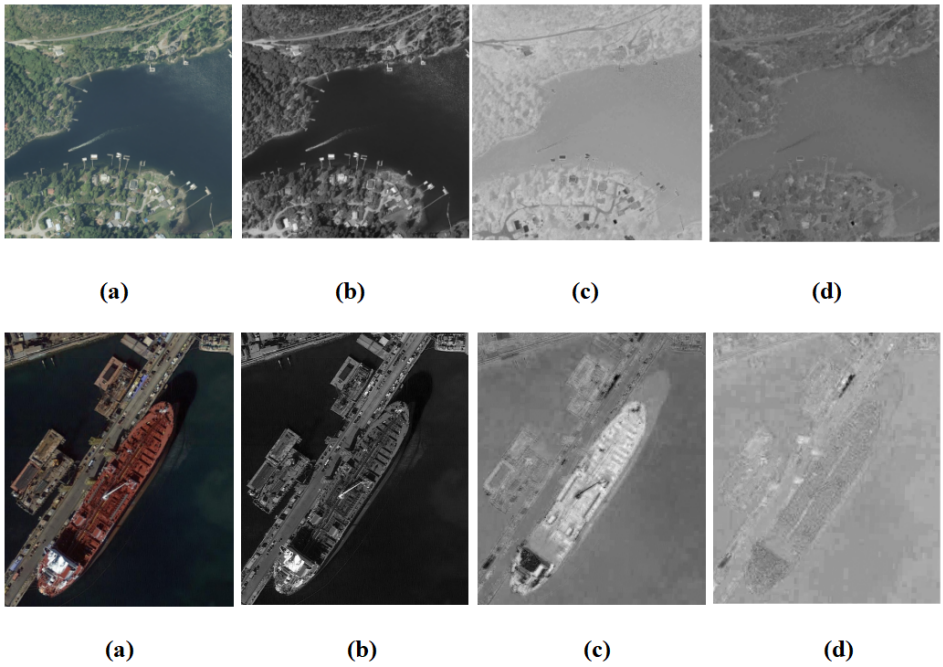


Fig. 2. Diagram of principal component analysis results. (a) original image; (b) First principal component image; (c) Second principal component image; (d) Third principal component image

As can be seen from Figure 2(b), after principal component transformation, the brightness of the shadow region is greater than that of other ground objects, which is conducive to further shadow extraction. According to ENVI analysis, the shadow region is always at both ends of the first principal component, that is, at the maximum value of positive value and the minimum value of negative value. Therefore, normalization of the first principal component within [0,1] can better highlight the difference between shadows and other features.

3.3. Shadow Index Construction

After the color component C is extracted from the remote sensing image through HSI conversion of color space, the blue light value of the shadow area is larger than that of the red and green light value, which enhances the contrast with other ground objects. After principal component transformation of remote sensing image, the shadow region is located

at two endpoints of the first principal component. Through normalization, the shadow region occupies a larger value in the first principal component than other features, which is similar to the C component. Inspired by the similarity, this paper proposes a shadow index construction method combining C component and first principal component. The shadow index is expressed as follows:

$$shadow - index = \frac{C - PC1}{C + PC1}. \tag{4}$$

3.4. Weighted Threshold Segmentation

In the task of shadow detection, in order to detect the shadow region more conveniently, the three channels of RGB image are first separated to obtain the single channel information of R, G and B, and then the linear weighting algorithm of Equation (5) is used for information fusion processing between channels. Set the three channels of remote sensing image as R, G and B respectively, then the weighted gray value I_{gray} is:

$$I_{gray} = 0.2R(x, y) + 0.6G(x, y) + 0.2B(x, y). \tag{5}$$

After gray linear processing, the pixel intensity of the shadow part is lower than that of the non-shadow area. With the gray level intensity of 0-255 pixels as the horizontal coordinate and the frequency of each gray level pixel intensity as the vertical coordinate, the functional statistical mapping relationship between gray level intensity frequency and gray level intensity in remote sensing images is established.

In the gray distribution diagram (Fig.3 (d)), the area with low gray level is the shadow area, while the area with high gray level is the non-shadow area. The gray value changes of the shaded area are more concentrated, and the distribution range of the non-shaded area is more extensive than that of the shaded area. The trough between two crests is used as the threshold to segment the shadow region, although it can be recognized, but the segmentation and recognition effect is poor. At present, the most common method is to establish the threshold segmentation model based on pixel intensity by using the maximum inter-class variance method, and determine a threshold by using the threshold segmentation model adaptively. However, the method of determining the boundary threshold between shadow and non-shadow regions based on the maximum inter-class variance only describes the variance distance between shadow and non-shadow categories, and does not take into account the differences within each category. In this paper, a constraint term of minimum in-class variance is added to the maximum category variance. First, it calculates the maximum variance distance needed to distinguish between the two categories. The larger distance denotes the better the classification effect of shadow and non-shadow. Then it calculates the minimum variance distance between the elements in each class. The smaller minimum variance distance denotes the more dense the elements in the class, indicating that the probability of each gray pixel belonging to the same category is greater. The detailed steps are as follows.

1. Set the gray value interval of the shadow region and the non-shadow region in the remote sensing image as $[0, m]$, then the number of gray values of all pixels is shown

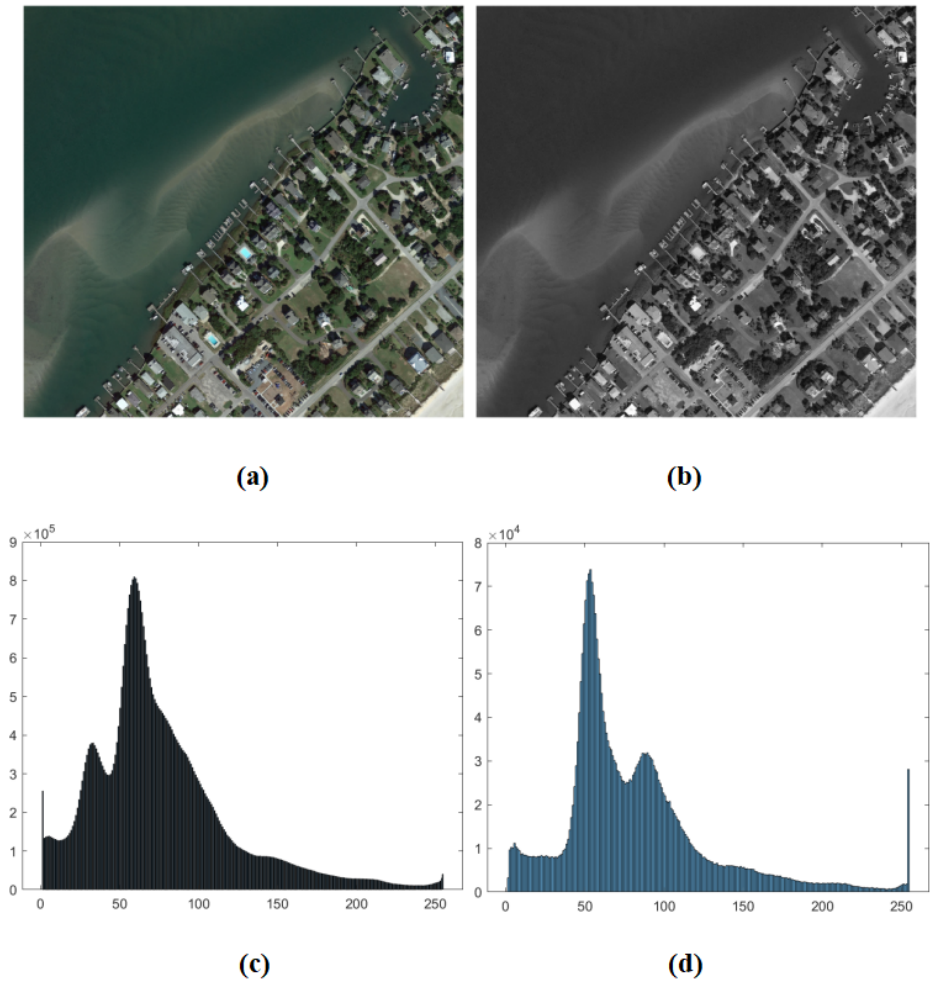


Fig. 3. Gray value distribution map. (a) original image; (b) Grayscale linear processing graph; (c) One-dimensional histogram of (a); (d) One-dimensional histogram of (b)

in Equation (6), the probability of each gray level is shown in Equation (7), and the overall gray mean μ_{all} is shown in Equation (8):

$$N = \sum_{i=0}^{m} h(i). \tag{6}$$

$$p_i = \frac{h(i)}{N}. \tag{7}$$

$$\mu_{all} = \frac{\sum_{i=0}^{m}(h(i) \times i)}{\sum_{i=0}^{m} h(i)}. \tag{8}$$

Where i is the gray level of the gray distribution map. $h(i)$ is the frequency corresponding to the gray level i. m is the maximum gray level of the gray distribution diagram (255 in this paper). μ_{all} indicates the global average gray level.

2. Let T be any gray value of the interval $[0, m]$. T is used as the segmentation threshold, and the image of shadow area and non-shadow area is divided into gray interval $[0, T]$ and $[T+1, m]$, which are denoted as $C1$ and $C2$ respectively. Formula (9) and formula (10) are used to calculate the average gray level μ_{C1} and μ_{C2} of the two parts respectively.

$$\mu_{C1} = \frac{\sum_{i=0}^{T}(h(i) \times i)}{\sum_{i=0}^{T} h(i)}. \tag{9}$$

Where i is the gray level of the dark pixel in the shadow area in the gray distribution map. $h(i)$ is the corresponding frequency of gray level i of dark pixel in shadow region. T is the threshold of boundary between shadow area and non-shadow area. μ_{C1} is the average gray level of $C1$ region.

$$\mu_{C2} = \frac{\sum_{i=T+1}^{m}(h(i) \times i)}{\sum_{i=T+1}^{m} h(i)}. \tag{10}$$

Where i is the gray level of bright pixels in the non-shadow area of the gray distribution map. $h(i)$ is the corresponding frequency of gray level i of bright pixel in non-shadow area. T is the threshold of boundary between shadow area and non-shadow area. μ_{C2} is the average gray level of the $C2$ region.

3. Equations (11-14) are used to calculate the variances of $C1$ in the dark shadow region and $C2$ in the bright non-shadow region respectively.

$$w_{C1} = \sum_{i=0}^{T} p(i). \tag{11}$$

$$w_{C2} = \sum_{i=T+1}^{m} p(i). \tag{12}$$

$$\sigma_{C1} = \frac{\sum_{i=0}^{T}(i - \mu_{C1})^2 \times h(i)}{\sum_{i=0}^{T} h(i)}. \tag{13}$$

$$\sigma_{C2} = \frac{\sum_{i=T+1}^{m}(i - \mu_{C2})^2 \times h(i)}{\sum_{i=T+1}^{m} h(i)}. \tag{14}$$

4. Formula (15) is used to calculate the maximum intra-class variance, maximum inter-class variance and the optimization evaluation function. The maximum intra-class variance σ_w at the threshold T is:

$$\sigma_w = w_{C1}\sigma_{C1} + w_{C2}\sigma_{C2}. \tag{15}$$

The maximum inter-class variance σ_b at the threshold T is:

$$\sigma_b = w_{C1}(\mu_{C1} - \mu_{all})^2 + w_{C2}(\mu_{C2} - \mu_{all})^2. \tag{16}$$

To ensure the optimal segmentation threshold, two conditions are required: one is that the variance between the shaded part and the non-shaded part is the largest; the other is that the variance within the shaded area and the non-shaded area is the smallest. In order to describe this phenomenon, the segmentation optimization function is defined:

$$\zeta(T) = max(\sigma_b(T)/\sigma_w(T)). \tag{17}$$

Where ζ is the optimization factor, and the iterative method is used to traverse all gray levels, then the threshold T corresponding to the largest optimization factor is the optimal segmentation threshold.

5. Set the above threshold T [43], and use Equation (18) to segment the image:

$$Segmentation = \begin{cases} 255 & Optimalthreshold > T \\ 0 & otherwise \end{cases} \tag{18}$$

4. Experiments and Analysis

After analyzing the experimental process and data, the accuracy rate (ACC) and false detection rate (FDR) are selected as the evaluation indexes of remote sensing image shadow detection effect [44,45]. The data set for this article is derived from Google Earth and DOTA [46]. Due to the limited content space, we choose four sets of images as experimental results analysis. Figure 4 is the sample data set.

Firstly, we obtain the one-dimensional and two-dimensional histogram of the four samples, in order to understand the pixel distribution of the original image more clearly. The results are shown in figure 5 and figure 6. The results show that the distribution of pixels in the original image is very uneven.

Figures 7,8,9 show the color map, threshold segmentation diagram and shadow detection effect diagram with the proposed algorithm in four scenes.

This paper is compared with three advanced shadow detection methods including DUS [47], MUSA [48], MUCS [49]. Table 1 shows that the accuracy rate and false detection rate of the proposed method on the data set are superior to the other three methods. It can be seen from the data in Table 1 that the shadow detection results in this paper are better than other methods.

For image 1, the accuracy of the proposed method is 95.4%, which is 6.0%, 4.1% and 7.1% higher than that of other methods, respectively. The FDR values of DUS, MUSA and MUCS are 12.51, 9.45 and 7.28 respectively, which are 6.29, 3.33 and 1.16 higher than that of the presented method in this paper (6.12). For image 2, ACC values of DUS,

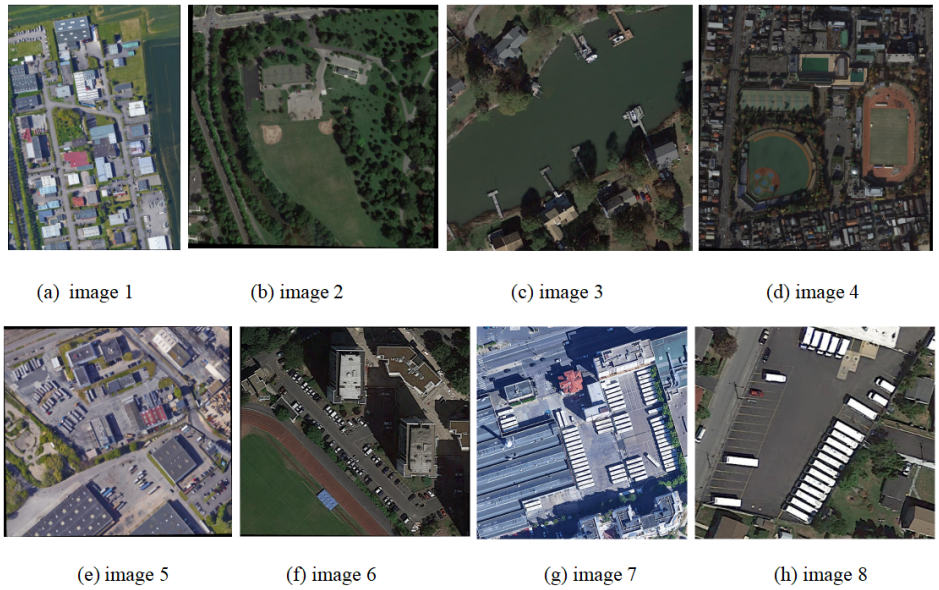


Fig. 4. Sample images

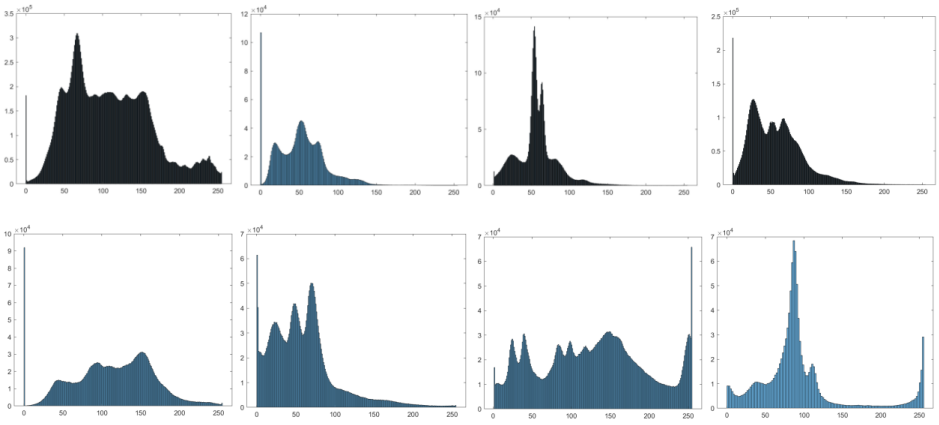


Fig. 5. One-dimensional histogram

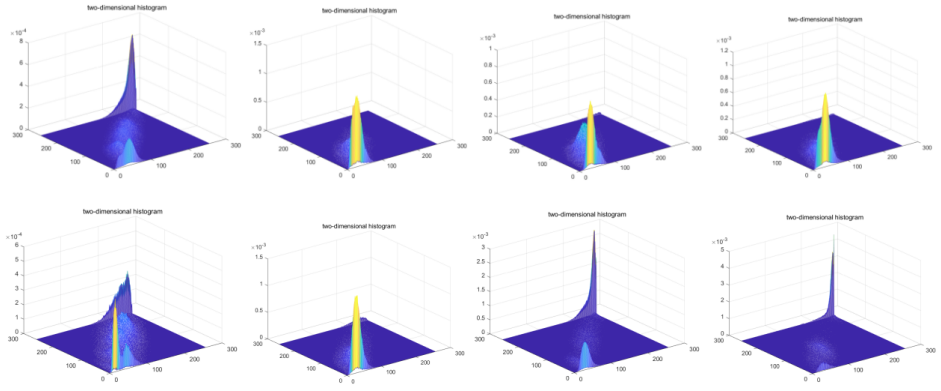


Fig. 6. Two-dimensional histogram

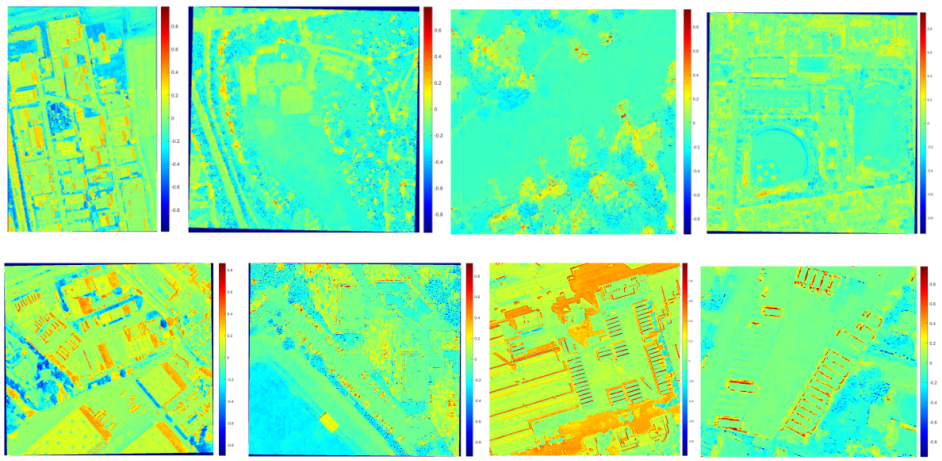


Fig. 7. Colour map

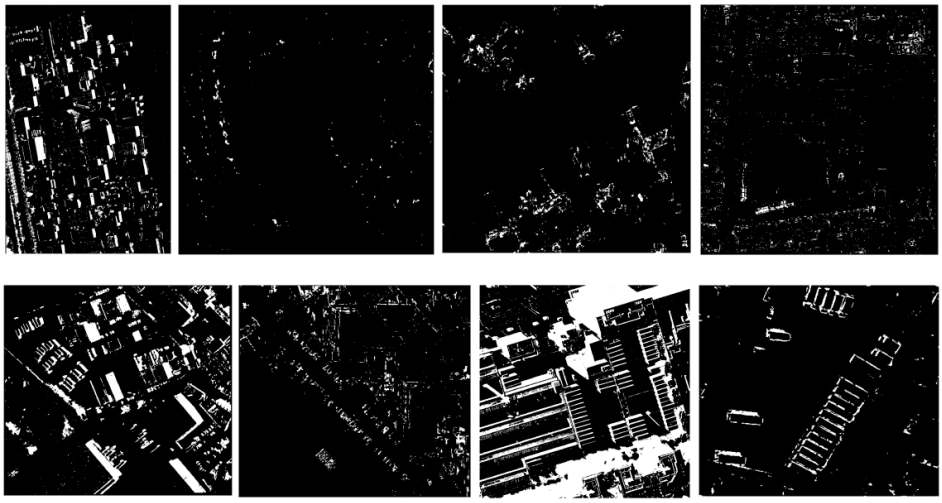


Fig. 8. Threshold segmentation

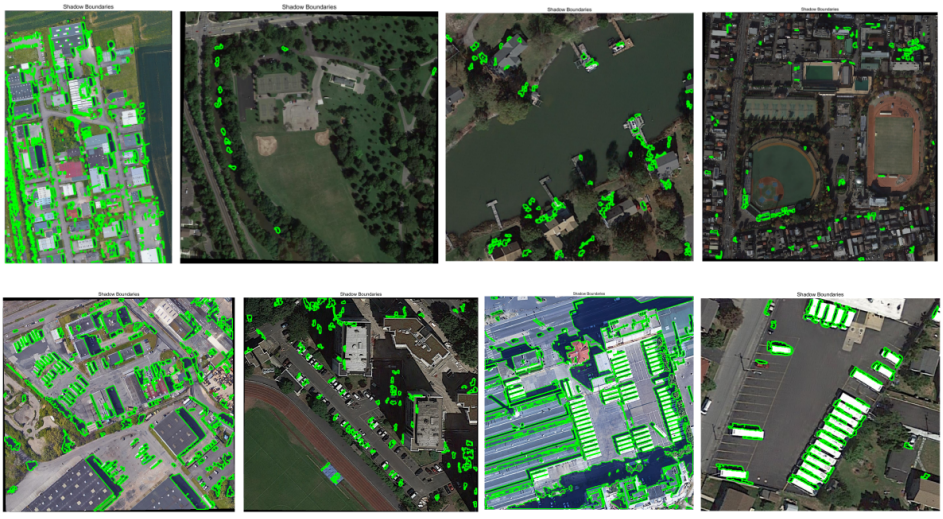


Fig. 9. Shadow detection result

Table 1. Comparison with other detection methods

Test image	Index	DUS	MUSA	MUCS	Proposed
Image1	ACC	0.894	0.913	0.937	0.954
Image1	FDR	12.51	9.45	7.28	6.12
Image2	ACC	0.875	0.886	0.907	0.935
Image2	FDR	17.32	14.81	10.64	9.78
Image3	ACC	0.907	0.928	0.935	0.961
Image3	FDR	14.69	11.58	10.44	8.94
Image4	ACC	0.863	0.892	0.917	0.934
Image4	FDR	18.52	16.74	13.55	10.95
Image5	ACC	0.906	0.915	0.921	0.946
Image5	FDR	11.85	9.93	8.31	7.25
Image6	ACC	0.885	0.891	0.907	0.922
Image6	FDR	13.25	11.74	9.33	7.21
Image7	ACC	0.927	0.939	0.951	0.960
Image7	FDR	11.24	8.11	7.58	6.93
Image8	ACC	0.902	0.927	0.933	0.958
Image8	FDR	12.35	11.26	9.75	7.07

MUSA, MUCS and proposed method are 87.5%, 88.6%, 90.7% and 93.5%, respectively. Obviously, the proposed method has some advantages over the other three methods. The FDR value of the method in this paper is 9.78, which is significantly lower than the other three methods. Similarly, in the other 7 images, the new method in this paper also has a similar trend. Also, we can see similar characteristics from Figure 10.

The running efficiency of the algorithm is also an important index to evaluate the performance of the algorithm. In order to measure the running efficiency of the algorithm proposed in this paper, the running time of different methods is given in Table 2. The visualized result is shown in Figure 11.

Table 2. Detection time with different methods/s

Test image	size	DUS	MUSA	MUCS	Proposed
Image1	2610×3931	5.71	4.66	4.25	3.36
Image2	936×913	3.04	2.89	2.37	1.24
Image3	1104×1103	4.71	3.47	2.86	2.35
Image4	1648×1732	3.97	3.65	3.17	2.83
Image5	1112×1026	4.56	3.74	3.25	2.84
Image6	1113×1015	3.75	3.23	2.77	2.51
Image7	1289×1302	3.82	3.31	2.86	2.55
Image8	673×594	3.87	3.19	2.64	2.36

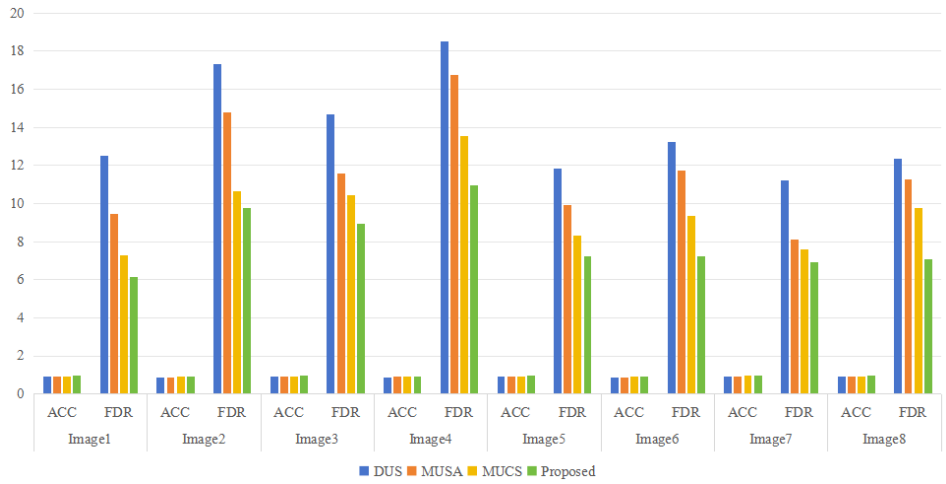


Fig. 10. Evaluation index comparison results

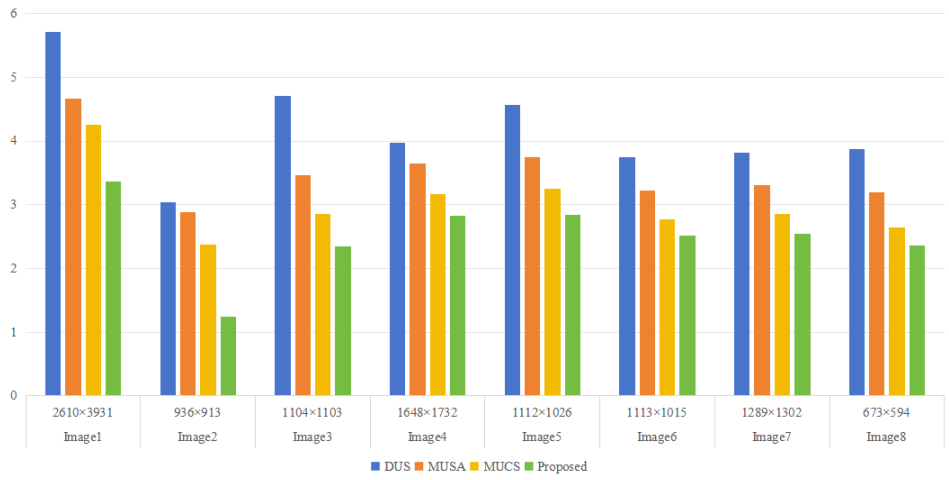


Fig. 11. Results of Table 2

5. Conclusions

Aiming at the low accuracy and robustness of shadow detection under single feature, one new threshold segmentation method based on information fusion for object shadow detection is presented in remote sensing images. Firstly, object shadow pixels are screened using intensity and chromaticity information in HSI color space. Secondly, the remote sensing image is carried out by principal component analysis (PCA) to obtain the first principal component. A new shadow index is constructed using the results obtained from HSI and the first principal component. Thirdly, based on the results of the above two information fusion, a threshold segmentation model is established using the improved threshold segmentation algorithm between the maximum and the minimum threshold segmentation algorithm, so as to obtain the final object shadow detection results. Finally, affluent experiments are conducted on the datasets collected from Google Earth. Experiments show that the proposed algorithm can accurately and quickly detect the shadow parts in the former scenic area, and has good robustness to the situation where the scene illumination changes or the local texture of the object is not obvious, which lays a good foundation for the accurate tracking and recognition of the target in the later stage.

Overall, the current cloud detection methods still have some limitations, and some areas need to continue to improve. In terms of cloud detection, there is almost no general algorithm that can detect cloud and cloud shadow on sensors, and different methods have different application conditions and scopes. Because some large-scale tasks require massive remote sensing image data, especially in some tasks that require intensive time series data and require multi-sensor data to work together, the existing cloud/shadow detection methods are limited in scope of application and poor in generalization. In addition, the accuracy of cloud detection will be affected by the background information of remote sensing images, such as ice and snow, which often cause misjudgment and missing judgment during detection, and the robustness is poor. In order to achieve a better cloud detection effect, auxiliary images with sufficient time continuity and non-overlapping cloud regions are needed, and the application prerequisite is high. At the same time, the problem of image registration and calibration across sensors and radiation difference between images should be solved.

Acknowledgments. This work was supported by the National Natural Science Foundation of China under Grants 62071084 and 62001434. The author is grateful for the anonymous review by the review experts.

References

1. Rodriguez-Moreno F, Kren J, Zemek F, et al. Advantage of multispectral imaging with sub-centimeter resolution in precision agriculture: generalization of training for supervised classification[J]. Precision Agriculture, 2017, 18: 615-634.
2. Sishodia R P, Ray R L, Singh S K. Applications of remote sensing in precision agriculture: A review[J]. Remote Sensing, 2020, 12(19): 3136.
3. Almeida L P, Almar R, Bergsma E W J, et al. Deriving high spatial-resolution coastal topography from sub-meter satellite stereo imagery[J]. Remote Sensing, 2019, 11(5): 590.
4. Moortgat J, Li Z, Durand M, et al. Deep learning models for river classification at sub-meter resolutions from multispectral and panchromatic commercial satellite imagery[J]. Remote Sensing of Environment, 2022, 282: 113279.

5. Wohlfeil J, Hirschmller H, Piltz B, et al. Fully automated generation of accurate digital surface models with sub-meter resolution from satellite imagery[J]. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2012, 39: 75-80.
6. Kumar S, Kaur A. Algorithm for shadow detection in real-colour images[J]. International Journal on Computer Science and Engineering, 2010, 2(07): 2444-2446.
7. Srikantha A, Sidib D. Ghost detection and removal for high dynamic range images: Recent advances[J]. Signal Processing: Image Communication, 2012, 27(6): 650-662.
8. Zhang H, Sun K, Li W. Object-oriented shadow detection and removal from urban high-resolution remote sensing images[J]. IEEE transactions on geoscience and remote sensing, 2014, 52(11): 6972-6982.
9. Abd-El Monsef H, Smith S E. A new approach for estimating mangrove canopy cover using Landsat 8 imagery[J]. Computers and Electronics in Agriculture, 2017, 135: 183-194.
10. Shi L, Fang J, Zhao Y. Automatic shadow detection in high-resolution multispectral remote sensing images[J]. Computers and Electrical Engineering, 2023, 105: 108557.
11. Wang X, Voytenko D, Holland D M. Accuracy evaluation of digital elevation models derived from Terrestrial Radar Interferometer over Helheim Glacier, Greenland[J]. Remote Sensing of Environment, 2022, 268: 112759.
12. Ward G J, Wang T, Geisler-Moroder D, et al. Modeling specular transmission of complex fenestration systems with data-driven BSDFs[J]. Building and Environment, 2021, 196: 107774.
13. Du H, Chen X, Xi J. An improved background segmentation algorithm for fringe projection profilometry based on Otsu method[J]. Optics Communications, 2019, 453: 124206.
14. Fu H, Zhou T, Sun C. Object-based shadow index via illumination intensity from high resolution satellite images over urban areas[J]. Sensors, 2020, 20(4): 1077.
15. Hou S W, Sun W F, Zheng X S. Overview of cloud detection methods in remote sensing images[J]. Space Electronic Technology, 2014, 11(3): 68-76.
16. Liu Z, Yang J, Wang W, et al. Cloud detection methods for remote sensing images: a survey[J]. Chinese Space Science and Technology, 2023, 43(1): 1.
17. Mahajan S, Fataniya B. Cloud detection methodologies: Variants and developmentA review[J]. Complex & Intelligent Systems, 2020, 6: 251-261.
18. Siddiq S, Kaur K, Dhir R. Automatic Detection of Cloudy and Non-Cloudy SAR Images Using Convolutional Neural Networks[C]//2023 Third International Conference on Secure Cyber Computing and Communication (ICSCCC). IEEE, 2023: 350-355.
19. Scaramuzza P L, Bouchard M A, Dwyer J L. Development of the Landsat data continuity mission cloud-cover assessment algorithms[J]. IEEE Transactions on Geoscience and Remote Sensing, 2011, 50(4): 1140-1154.
20. Zhu Z, Woodcock C E. Object-based cloud and cloud shadow detection in Landsat imagery[J]. Remote sensing of environment, 2012, 118: 83-94.
21. Yin S, Wang L, Wang Q, et al. M2F2-RCNN: Multi-functional faster RCNN based on multi-scale feature fusion for region search in remote sensing images[J]. Computer Science and Information Systems, 2023 (00): 54-54.
22. Zhu Z, Woodcock C E. Automated cloud, cloud shadow, and snow detection in multitemporal Landsat data: An algorithm designed specifically for monitoring land cover change[J]. Remote Sensing of Environment, 2014, 152: 217-234.
23. Teng L, Qiao Y, Shafiq M, et al. FLPK-BiSeNet: Federated Learning Based on Priori Knowledge and Bilateral Segmentation Network for Image Edge Extraction[J]. IEEE Transactions on Network and Service Management, 2023.
24. Xie F, Shi M, Shi Z, et al. Multilevel cloud detection in remote sensing images based on deep learning[J]. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2017, 10(8): 3631-3640.
25. Qiu S, Zhu Z, He B. Fmask 4.0: Improved cloud and cloud shadow detection in Landsats 4C8 and Sentinel-2 imagery[J]. Remote Sensing of Environment, 2019, 231: 111205.

26. Chang H, Fan X, Huo L, et al. Improving Cloud Detection in WFV Images Onboard Chinese GF-1/6 Satellite[J]. Remote Sensing, 2023, 15(21): 5229.
27. Xu D, Li X, Zhao L, et al. Hyperspectral Remote Sensing Image Cloud Detection Based on Spectral Analysis and Dynamic Fractal Dimension[J]. Laser & Optoelectronics Progress, 2019, 56(10): 101003.
28. Calin M A, Calin A C, Nicolae D N. Application of airborne and spaceborne hyperspectral imaging techniques for atmospheric research: Past, present, and future[J]. Applied Spectroscopy Reviews, 2021, 56(4): 289-323.
29. Zhang H, Huang Q, Zhai H, et al. Multi-temporal cloud detection based on robust PCA for optical remote sensing imagery[J]. Computers and Electronics in Agriculture, 2021, 188: 106342.
30. Zhang Y, Yang C, Tao R, et al. Multi-temporal Cloud Detection Method for Qinghai-Tibet Plateau based with FY-4A Data[J]. Remote Sensing Technology and Application, 2020, 35(2): 389-398.
31. Khoomboon S, Kasetkasem T, Rakwatin P. A land cover mapping algorithm for thin to medium cloud-covered remote sensing images using a level set method[J]. International Journal of Remote Sensing, 2022, 43(10): 3803-3842.
32. Candra D S, Phinn S, Scarth P. Automated cloud and cloud-shadow masking for Landsat 8 using multitemporal images in a variety of environments[J]. Remote Sensing, 2019, 11(17): 2060.
33. Zhang X, Liu L, Chen X, et al. A novel multitemporal cloud and cloud shadow detection method using the integrated cloud Z-scores model[J]. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2019, 12(1): 123-134.
34. Yin S. Object Detection Based on Deep Learning: A Brief Review[J]. IJLAI Transactions on Science and Engineering, 2023, 1(02): 1-6.
35. Zhu X, Gao F, Liu D, et al. A modified neighborhood similar pixel interpolator approach for removing thick clouds in Landsat images[J]. IEEE Geoscience and Remote Sensing Letters, 2011, 9(3): 521-525.
36. Liu M, Liu X, Li J, et al. Evaluating total inorganic nitrogen in coastal waters through fusion of multi-temporal RADARSAT-2 and optical imagery using random forest algorithm[J]. International journal of applied earth observation and geoinformation, 2014, 33: 192-202.
37. Huang S, Ding J, Liu B, et al. The capability of integrating optical and microwave data for detecting soil moisture in an oasis region[J]. Remote Sensing, 2020, 12(9): 1358.
38. Li T, Cheng X. Estimating daily full-coverage surface ozone concentration using satellite observations and a spatiotemporally embedded deep learning approach[J]. International Journal of Applied Earth Observation and Geoinformation, 2021, 101: 102356.
39. Ji S, Dai P, Lu M, et al. Simultaneous cloud detection and removal from bitemporal remote sensing images using cascade convolutional neural networks[J]. IEEE Transactions on Geoscience and Remote Sensing, 2020, 59(1): 732-748.
40. Abdullah A S S, Abed M A, Al Barazanchi I. Improving face recognition by elman neural network using curvelet transform and HSI color space[J]. Periodicals of Engineering and Natural Sciences, 2019, 7(2): 430-437.
41. Yin S, Li H. Hot region selection based on selective search and modified fuzzy C-means in remote sensing images[J]. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2020, 13: 5862-5871.
42. LI Z W, ZHENG W, FANG J, et al. Optimizing suitability area of underwater gravity matching navigation based on a new principal component weighted average normalization method[J]. Chinese Journal of Geophysics, 2019, 62(9): 3269-3278.
43. Jia Y L, Zhang W, Meng L K. A study of selection method of NDWI segmentation threshold for GF1 image[J]. Remote Sensing for Land and Resources, 2019, 31(01): 95-100.
44. Hu Q, Wu W, Xia T, et al. Exploring the use of Google Earth imagery and object-based methods in land use/cover mapping[J]. Remote Sensing, 2013, 5(11): 6026-6042.

45. Amani M, Ghorbanian A, Ahmadi S A, et al. Google earth engine cloud computing platform for remote sensing big data applications: A comprehensive review[J]. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2020, 13: 5326-5350.
46. Ding J, Xue N, Xia G S, et al. Object detection in aerial images: A large-scale benchmark and challenges[J]. IEEE transactions on pattern analysis and machine intelligence, 2021, 44(11): 7778-7796.
47. Li D, Wang S, Xiang S, et al. Dual-stream shadow detection network: biologically inspired shadow detection for remote sensing images[J]. Neural Computing and Applications, 2022, 34(12): 10039-10049.
48. Liu D, Zhang J, Wu Y, et al. A shadow detection algorithm based on multiscale spatial attention mechanism for aerial remote sensing images[J]. IEEE Geoscience and Remote Sensing Letters, 2021, 19: 1-5.
49. Alvarado-Robles G, Osornio-Rios R A, Solis-Munoz F J, et al. An approach for shadow detection in aerial images based on multi-channel statistics[J]. IEEE Access, 2021, 9: 34240-34250.

Shoulin Yin received the B.S. and M.S. degrees in image processing from Software College, Shenyang Normal University, Shenyang, China, in 2013 and 2015, respectively. He is currently a PHD with School of Information and Communication Engineering, Harbin Engineering University. He has published more than 20 technical articles in scientific journals and conference proceedings. His research interests include image fusion, object detection, and image recognition. Email:yslin@hit.edu.com.

Liguo Wang received his M.A. degree in 2002 and Ph.D. degree in signal and information processing in 2005 from Harbin Institute of Technology, Harbin, China. He held postdoctoral research position from 2006 to 2008 in the College of Information and Communications Engineering, Harbin Engineering University. He is currently a Professor with Information and Communications Engineering, Dalian Minzu University, Dalian, China. His research interests are remote sensing image processing and machine learning. He has published three books, 27 patents, and more than 200 papers in journals and conference proceedings. Email: wangliguo@hrbeu.edu.cn.

Lin Teng received the M.A. degree in computer application technology from Shenyang Normal University, Shenyang, China, in 2020. She is currently working toward the Ph.D. degree in communication engineering with the College of Information and Communication Engineering, Harbin Engineering University, Harbin, China. Her research interests include image processing and semantic segmentation.

Received: December 30, 2023; Accepted: April 20, 2024.

Computer Science and Information Systems 21(4):1243–1268 https://doi.org/10.2298/CSIS231120034M

Tourist services management through clients scoring using a bio-inspired agent architecture ⋆

Raúl Moreno, Alberto Fernández-Isabel, Víctor Aceña,
Isaac Martín de Diego, and Javier M. Moguerza

Data Science Lab. at Rey Juan Carlos University
C/ Tulipán, s/n, Móstoles, 28933, Madrid, Spain
r.morenoi@alumnos.urjc.es
alberto.fernandez.isabel@urjc.es
victor.acena@urjc.es
isaac.martin@urjc.es
javier.moguerza@urjc.es

Abstract. Tourism has become an economic engine for several countries during the last decades. Each time more and more individuals consider visiting other places during their vacation period. These places cover very different options, from the typical beach and mountain tourism to the less common urban and cultural trips. These travels hoard multiple transport means and facilities in the destination place that have to be correctly managed. Therefore, the rise of automatic systems to address the related operations and processes is a crucial issue nowadays. These systems are usually focused on the final users (the tourists) and make recommendations about their available possibilities. However, it is not easy to find a multi-purpose recommendation system covering all the needs from the perspective of the travel agency. In this paper, a complete framework called *Pharaoh* able to make recommendations to customers covering the final user perspective, and to provide support to the travel agents, is presented. This assistance filters the best travel, accommodation, and activity options according to the desires of the customers. This novel functionality allows selecting the customer with the best propensity to book a tourist service. This workload is distributed using a bio-inspired Multi-Agent System (MAS). Moreover, *Pharaoh* considers the feedback from clients after the completion of the tourist opportunity to improve future recommendations. Several experiments in real environments have been addressed to show the viability of the proposal. It can be concluded that the system enhances the quality of the service provided by the travel agency and its profits.

Keywords: Tourism, Recommendation system, Propensity assessment, Intelligent agents, Smart assistant framework

1. Introduction

Today, tourism is one of the world's most profitable industries [10]. Each year several tourists travel to visit famous cities and mythic architectures or to meet new cultures

⋆ It is an advanced release of a previously published conference paper: Moreno, R., Viajes, M., Fernández-Isabel, A., de Diego, I. M., Moguerza, J. M., Lancho, C., & Cuesta, M. (2022, September). Automatic detection of potential customers by opinion mining and intelligent agents. In 2022 17th Conference on Computer Science and Intelligence Systems (FedCSIS) (pp. 93-101). IEEE

and ways of living. Moreover, other tourists only look for revelry or calm and relaxation during vacation time, where wonderful landscapes and beaches light up the stage [42].

All these travel events produce the use of several means of transport like airplanes, trains, buses, and cars [12]. These events generate millions of bookings in hotels, apartments, and other similar buildings specifically dedicated to hosting tourists.

All these issues must be managed by travel agencies, retrieving the information from destination places, and the preferences of the different tourists [34]. Therefore, the appearance of systems focused on providing support to the final user (the tourist and the travel agent) that simplifies the booking process is necessary.

Furthermore, travel agents also need support to provide customers with the most interesting options to capture their interest. Therefore, systems able to manage this task automatically, making recommendations of the most suitable places and trips, or selecting the best transport schedules according to the client´s needs are also basic [46].

Nevertheless, these systems have their limitations. The main bottleneck consists of the difficulty of elaborating budgets by the travel agency employee. It is a very time-consuming task since the configuration of a trip increases its complexity when multiple destinations are selected. Another relevant problem is the typical inability to provide feedback about the tourist services offered to clients. This point is essential for travel agencies because they need to know if the work achieved was adequate and if customers were satisfied after finishing the contracted tourist event. In this sense, the recommendations that filter the initial preferences of the clients are the key point. These usually mark the proper evolution of the tourist services, as they must capture the interest of the client in the first instance.

The COVID-19 outbreak is another point to consider in the tourism area. The pandemic situation all over the world has caused governments to implement several restrictions to control the spreading of the coronavirus. This has had a drastic impact on the mobility of the population, reducing the number of travels, and, as a consequence, tourism has suffered significant economic losses [6]. However, with the advances in vaccination programs and the relaxation of the measures against the contagion of the virus, the rise of tourism is going to become a reality very soon.

For these reasons, it is mandatory the development of a complete framework that can cushion this new growing demand. This system must be able to address the weak points detected, managing and encompassing all the possible issues produced since a client shows interest in a tourist event, providing support to the travel agent in the selection of the possibilities, and finally obtaining the feedback of the client when the tourist service concludes.

Notice that the strength of a framework of this type lies in the management of tourist packages where multiple destinations and services are contracted. Typical examples of these trips are, among others, honeymoons and cruises.

In this regard, this paper presents the *Pharaoh* framework. It is an automatic analytical Enterprise Resource Planning (ERP) and Costumer Relationship Management (CRM) system for Travel Agencies that includes several graphical assistants to provide complete support to travel agents and tourists. Its novelty resides in the next points:

- It covers all the processes related to tourist services sales, focusing mainly on the interest of the client and the specific bargains that fit better for the travel agency.

Thus, travel agents and clients are guided synchronously by the opinion produced by the system at each step of the sale.
- It generates budgets in a fast way and with quotations in real time. This fact facilitates the decision-making tasks of the travel agencies and the client can be informed at the moment about the best individual available options or combinations in specific tourist packages.
- It can learn from the experience and can prioritize those clients who are prone to commit the book of tourist services. Notice that this ability is one of the most differential strengths of the proposal regarding other existing approaches. An Machine Learning (ML) model to estimate the probability of the sale has been included in the system to achieve that point.
- Intelligent agents following a Multi-Agent System (MAS) have been included to establish communication and knowledge interchange. This MAS has been built according to the rules of a bio-inspired anthill, efficiently promoting the distribution of the workload [35]. This is relevant because companies usually need real-time interactions to satisfy the requests made by customers and maximize the benefits through saving computer resources (e.g., possible expenses in specific cloud computing architectures).

A set of experiments has been achieved to show the viability of the proposal. The module in charge of selecting the best clients has been tested independently to evaluate its potential. Then, the complete system and its functionalities have been put into the spotlight. Promising results have been obtained from these tests.

The remainder of the paper is organized as follows. Section 2 introduces the foundations and relevant literature. Section 3 details the architecture and the different features of *Pharaoh*. Section 4 presents a set of experiments to illustrate the performance of the system. Finally, Section 5 concludes and provides further guidelines.

2. Background

This section introduces the foundations of the *Pharaoh* framework, covering the different perspectives included in it together with the existing state of the art in the tourism domain. First, an overview related to the knowledge management systems is achieved. There, it is detailed how this kind of system works, and similar approaches that provide interesting features are addressed. Second, recommendation systems are introduced explaining how they are used in real environments. The different perspectives of these systems are also detailed in this point. Finally, previous approaches oriented to managing tourism-related processes by travel agencies and customers are presented.

2.1. Recommendation systems

Recommendation systems are mainly focused on filtering the information they store. This filter is flexible and adaptable to the requirements and interests of the users [1].

Delving into the filtering process, these systems produce a profile for each one of the users. This profile contains the different parameters that are used to measure (and consequently compare) the interests of the user.

The parameters that produce the user´s profile can be provided to the system during the registration process of the user, or they can be included and modified during the interaction of the user with the system. This latter is the more relevant, as it allows the system adaptation to possible fluctuating opinions and interests of the users. Moreover, it also eases the improvement of the system, making it more accurate in the recommendations each time, as it can know if the previous recommendations were satisfied for the user. A well-known instance of this approach is the recommendation systems based on reinforcement learning [24].

Recommendation systems usually use three main perspectives to achieve the recommendation process: the selected topics, the relevance of the content, and the associated reputation. Notice that these perspectives are not excluded and they could be found mixed in some recommendation systems.

Systems that consider the topic recommendation perspective focus on the most common elements the users usually select. Thus, the software learns from the user according to the preferences [30]. For instance, if a user shows interest in beach destinations on a travel website, the recommendations should be related to other beach spots they might find appealing. This method aims to offer visitors new yet potentially attractive destinations, based on the preferences of users with similar profiles.

Systems for the relevance of content perspective are focused on the importance or popularity of the element to make recommendations. This importance is measured according to the number of users interacting with the product. Thus, these systems usually generate popularity rankings to promote the easy consultation of the products [25]. In tourism recommendation systems, if a point of interest attracts many visitors and stands out for its popularity at a given time, the system highlights it, presenting it among the top suggestions to the user.

A system based on reputation provides recommendations according to the obtained feedback [32]. In hotel competitiveness, where good service is crucial, positive feedback boosts the hotel's importance, improving its reputation. In contrast, negative feedback lowers its value, affecting its place in the market. They are specifically designed to reduce the uncertainty for the users and facilitate trust between entities [7]. A well-known instance is shared online where several users sell different products. Those systems usually support products and users with a better reputation.

In the case of the *Pharaoh* framework, it is a recommendation system that includes the three perspectives. Thus, it can organize different travels according to the topic (e.g., business, relaxation, or party among others), according to the relevance (e.g., well-known travel companies can be filtered), and also according to the reputation thanks to the feedback the users can provide when the travel finishes. Moreover, the system can also make recommendations to the travel agent regarding the possible customers. Therefore, this fact could be classified into the relevance-based recommendation perspective.

2.2. Intelligent agents to distribute the workload

Intelligent agents are software abstractions able to establish communication channels (direct or indirect), sharing a common environment. There, they interact with other agents and with the environment extracting and providing different pieces of information. These particularities allow the agents to simulate complex behaviors from the real world [18].

Concept	Meaning	Icon
Agent	An active element with explicit goals that is able to initiate some actions involving other elements of the simulation.	
Role	A specific collection of tasks performed by agents when pursuing a goal or offering some service to the other members of the society. It has as a result a specific behavior.	
Environment Application	An element of the environment. Agents can act on the environment using its actions and perceive information through its events.	
Goal	An objective of a role/agent. Roles/agents try to satisfy their goals executing tasks. A goal is achieved or fails if some elements (i.e. frame facts and events) are present or absent in the agent groups or the environment.	
Task	A capability of a role/agent. To execute a task, certain elements (i.e. frame facts and events) must be available. The execution produces/consumes some elements as result.	
Frame Fact	An element produced by a task, and therefore by the roles/agents.	
Mental State	Part of the internal state of a role/agent. It groups goals, frame facts and events, and specify conditions on them.	
Belief	Part of the mental state. It contains the knowledge (specific, role-based or general) that an agent possesses. This knowledge can be used to interact with its surrounded environment.	
Conversation	Communication between two or more agents to exchange information. One of these agents plays the role of initiator of the conversation, while the others are the receptors.	
Plan	Representation of the means by which a goal can be satisfied. It is usually structured to provide a deterministic meaning to the operations.	
Mental State Manager	Part of the internal state of a role/agent. It provides for operations to create, destroy and modify mental entities.	
Mental State Processor	Part of the internal state of a role/agent. It determines how a mental state evolves, described in terms of rules or planning.	

Fig. 1. Main concepts of INGENIAS for developing MASs

Regarding the features of the agents, they are proactive, autonomous, and independent. Agents base their knowledge on predefined rules to tackle the raised problems. Their ability to interact can be exploited to solve complex issues or to distribute the workload with certain coordination.

The life cycle of intelligent agents consists of satisfying a collection of goals following the knowledge available (from a predefined set of rules or dynamically acquired from the environment or other agents), and through several associated concepts (see Fig. 1). These goals can be independent, or they can be organized hierarchically. This latter promotes the idea of having sets of sub-goals that accomplish other goals at higher levels when they are satisfied. Goals are associated with a task (or a set of them) that is executed by the agents. Both goals and tasks are elements of the mental state of agents. This mental state

plays the role of a mind, being the place where the set of rules and knowledge are stored. Therefore, it supports the execution of the tasks and the satisfaction of the goals.

On the other hand, agents can take advantage of their ability to interact with the environment to solve complex problems having partial or reduced knowledge about a problem. The organization in MASs opens the collaboration, the competition, and also the negotiation. Agent-Based Modeling (ABM) [43] and Agent-Oriented Software Engineering (AOSE) [17] are standards in the domain that provide the elements and entities to address this issue. Well-known approaches that use MASs to solve complex problems or simulate real environments are road traffic simulations [13], distributed decision support systems [20], bio-inspired ML-based systems [9], and computer games [44].

MASs are usually designed using specific artifacts and entities to address the development phases of complex systems. It allows the production relationships and interactions between agents that are graphically represented and later transformed to source code automatically. INGENIAS, GAIA, Prometheus, and Tropos are well-known agent modelling methodologies in the domain [40].

Agent platforms are usually the standard solution to implement a previously modelled MAS. These platforms are commonly organized into source code libraries of a specific programming language. There, they include features to ease the distribution of the agents and manage their communication channels. Highlighted approaches in this area are JADE (Java) [8] and MESA (Python)[26].

For the development of *Pharaoh*, the selected agent methodology has been INGENIAS, adapting the agent model to the MESA framework for implementation purposes. The agent model is a bio-inspired distribution model based on the organization of ant colonies to generate a MAS distributed in several cumuli of agents (anthills) working together or independently according to the workload of the system.

2.3. Automatic managing of tourism

Automatic tourism management is one of the basic tasks in the area due to the large number of different processes to consider related to available offers, journey configurations, marketing opportunities, and customer indications and preferences [27]. In addition, travel agencies are evolving to manage tourist packages and customer preferences through the Internet. This means that personal relationships and face-to-face meetings have been changed for interactions with web pages that provide support and guidance.

The tourism management process is based on two main perspectives: the client and the seller (i.e., the final destination or the travel agency). The first one is the most widely addressed by the different studies and the developed systems. This issue is related to the implementation of new technologies and e-tourism (i.e., marketing and offers of tourism services on the Internet), which means that the tourist provider has direct contact with the customer [21].

Regarding the approaches that automatize the different tourism-related processes, most of the systems are recommendation systems for both perspectives.

In the case of the perspective of the travel agencies, they usually face clients who do not know where to travel and what their preferences are. Therefore, systems able to extract and process information from clients could provide support to the decision-making or to select the optimal vendor from the company for that customer [23]. The information can be textual content [28] or generated through the creation of profiles [2].

In the case of the perspective of customers, there are several possibilities to address. For instance, systems able to elaborate specific trip configurations for clients with special needs are very useful [36]. However, other approaches are more general. Examples of these general approaches are those that are focused on the selection of the best route according to the current weather or traffic conditions [41], or the recommendation of the best tourist trips for individuals or groups [22]. These latter are like virtual trip planner designers [3] that are usually based on the tastes of the clients and their former trips already completed [16]. For destination selection, similar systems have also been considered. They process the opinions of the tourists to find the most popular places for them [48] and also produce rankings of interest [4]. Finally, the issue of finding a suitable residence to live in during the journey according to the specific features of the client has also been considered. These features are mainly the needs of the clients [14], and their economic capacity [38].

The *Pharaoh* framework is mainly focused on the travel agency perspective providing specific elements to interact with the clients. In the first case, the vendor obtains real-time recommendations according to the indications made by the clients and their interests. Moreover, the system presents a novel component able to inform the vendor about the clients who are more prone to buy a tourist activity. This eases the work and increases the profit of the agency. In the second case, clients can introduce their particularities and interests. The system uses these to provide filtered information to the vendor.

3. The Pharaoh framework

The main objective of the *Pharaoh* framework is to support customers and travel agents (i.e., the final users in the tourism domain). Therefore, the system is a ERP and CRM that covers both perspectives related to tourist service management.

From the tourist perspective, *Pharaoh* provides different tourist services according to specific preferences and features (i.e., customer searches and filters the available options). From the travel agent's perspective, the system makes available the most relevant tourist services according to the preferences provided by the customer, the previously configured ratings, and the feedback provided by former customers about previous experiences with the tourist service of interest. Moreover, *Pharaoh* includes a novel functionality for easing the selection of the optimal customers (i.e., those prone to complete the booking process of a tourist service). This functionality allows travel agents to establish priorities and different actions according to each customer.

The system uses several information sources to obtain the desired knowledge like available hotels, flights, cars, etc. It also includes a visualization tool where users (customers and travel agents) can interact. This tool is composed of different graphical assistants to support the different tasks implemented by *Pharaoh*.

The following sections detail the architecture of the *Pharaoh* framework considering the different modules and emphasizing the processes that cover its functionalities.

3.1. General architecture and modules

The architecture of the *Pharaoh* framework consists of five different modules: the *Information gatherer*, the *Tourist service manager*, the *Customer manager*, the *Travel agent*

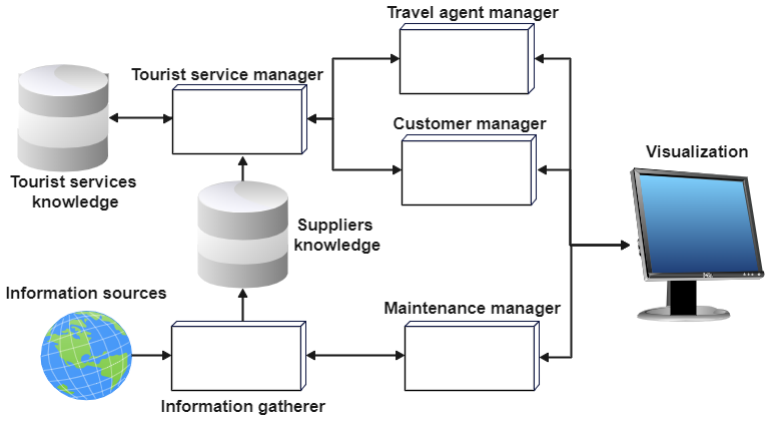


Fig. 2. Overview of the architecture of the *Pharaoh* framework

manager and the *Maintenance manager*. These modules are completed with two knowledge bases to persist the information: the *Suppliers knowledge* and the *Tourist services knowledge* repositories (see Fig. 2). These modules are capable of carrying out the three main processes: sale of a tourist service, customer feedback, and information update.

The *Maintenance manager* module serves as a central point for the maintenance of information and common parameters used by the rest of the modules of *Pharaoh*. Typical examples of maintenance of the information are related to the modifications of the tourist resource (e.g., name changes of hotels, new appearance of companies in the market, or permanent close events). It is also necessary to have the mapping of the same tourist resource from different providers to facilitate the system to generate services with different rates. It uses the *Information gatherer* module to achieve these actions (see Fig. 3).

The *Information gatherer* module collects the information from the virtual tourism market. This market is based on the exchange of information coming from web sources between suppliers and consumers through XML and JSON technologies. The module retrieves two types of information: static and dynamic. The first is not frequently modified, like the name, description, features, and relevant pictures. The second one fluctuates several times daily. Typical instances of this type of information are the occupancy level of a hotel or the rates. This information is gathered through web scraping techniques to automate the process and stored in the *Suppliers knowledge* repository. It is important to indicate that some sensible information is not stored due to contracts with suppliers, being only passively consumed by this module.

The *Suppliers knowledge* repository has as the main objective to manage the acquired knowledge corresponding to suppliers. The visual interface of *Pharaoh* provides specific graphical assistants to guide the user in managing this task. The system uses this stored information to produce automatic recommendations.

The *Tourist service manager* module is in charge of managing tourist opportunities and creating personalized offers to make recommendations (see Fig. 4). These recommendations are based on the calculations focused on providing the best tourism-related resources for an operation ordered by interest. Thus, it is defined the type of each tourism resource grouped by category. In the transport category, flights, collective public transport

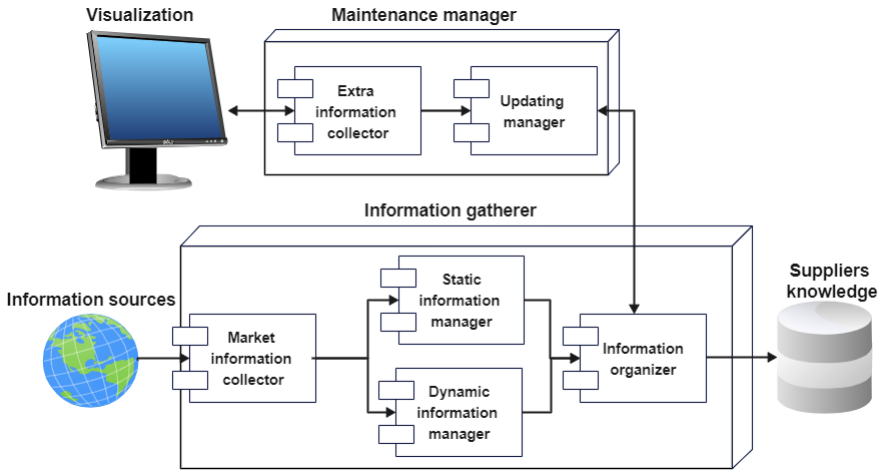


Fig. 3. Except of the *Maintenance manager* and *Information gatherer* modules

(e.g., trains or buses), and renting of private vehicles are included. In the accommodation category, hotels, apartments, and similar places are considered, Finally, in the extras category, tourist activities (e.g., tourist visits or reserved circuits in the destination) and insurance are classified here.

The expression that provides the best tourist resources to configure a complete tourist service (TS) is described as follows:

$$TS = \max_{k \in \{1,\ldots,|T|\cdot|A|\cdot|E|\}} TS_k = \sum_{C \in \{T,A,E\}} \theta_C \cdot Score^C \tag{1}$$

where T is for transport, A is for accommodation, and E is for extras. In addition, the θ values are weights from 0 to 1 to indicate relevance. Note that the sum of these weights must be 1. Moreover, $Score^C$ is the average value of the values $Score_j^C$ corresponding to the selected tourist resources of the tourist service. Thus, $Score_j^C$ is defined as follows:

$$Score_j^C = w_{u,j}^C \cdot f_{u,j}^C + w_{e,j}^C \cdot f_{e,j}^C \tag{2}$$

where $j \in \{1, \ldots, |C|\}$, where $|C|$ is the number of elements in category C. The values denoted with w are the weights of the partial contributions to the global score subsequently defined.

The element $f_{u,j}^C$ is about the preferred variables, and it is expressed as follows:

$$f_{u,j}^C = \frac{1}{n} \sum_{i}^{n} preferred_{j,i}^C \tag{3}$$

where $preferred_{j,i}^C$ is a value $\in [0, 1]$ indicating the feedback value provided by customers based on their previous experience. Thus, $f_{u,j}^C$ is the average value of the opinions of the customers.

In the case of the element $f_{e,j}^T$, it depends on the considered category (T, A, E):

$$f^T_{e,j} = w^T_{quality} \cdot quality^T_j + w^T_{stopovers} \cdot stopovers^T_j + w^T_{time} \cdot time^T_j + w^T_{price} \cdot price^T_j \quad (4)$$

$$f^A_{e,j} = w^A_{quality} \cdot quality^A_j + w^A_{location} \cdot location^A_j + w^A_{services} \cdot services^A_j + w^A_{price} \cdot price^A_j \quad (5)$$

$$f^E_{e,j} = w^E_{quality} \cdot quality^E_j + w^E_{flexibility} \cdot flexibility^E_j + w^E_{services} \cdot services^E_j + w^E_{price} \cdot price^E_j \quad (6)$$

At each category, it is considered a set of measures with values $\in [0, 1]$. With the Transport Category: *quality* means the quality of the transport, *stopovers* means the number of possible stopovers, *time* means the estimated time spent to reach the destination, and *price* means the expensiveness. With the Accommodation Category: *quality* means the quality of the accommodation, *location* means the location concerning the distance to the tourist areas and the safety of the area, *services* means the level of the provided services, and *price* means the expensiveness. Finally, with the Extras Category: *quality* means the quality of the extra, *flexibility* means the flexibility in hours and hiring, *services* means the level of the provided services, and *price* means the expensiveness. Note that these measures are pondered by weights w whose sum is 1 for each category.

With the measures from each one of the categories, the system intends to answer the following key questions for a trip:

- *HOW*: by defining the quality or level of the resource.
- *WHEN*: by classifying the time needed for reaching the destination.
- *WHERE*: by classifying the place and its surroundings.
- *WHAT*: by classifying the acceptability of the extra resource in terms of what is offered.
- *WHY*: that defines the set of extras and safety offered by the resource.

The module is responsible for filtering customers according to their propensity to complete the purchase. A propensity purchase estimation algorithm is in charge of establishing this classification. This algorithm uses classic scoring processing techniques and two different ML methods in a two-layer architecture to categorize the clients into five groups according to their booking probability. More details about this issue are presented later.

The module comprises three components: the *tourist services organizer*, the *tourist service filter* and the *customer feedback gatherer*. The first contains a MAS to distribute the workload and achieve the decision-making task parcelled out. The second is in charge of filtering activities according to the requests made by the customers, and the third manages the feedback provided by these latter.

Regarding the MAS, it achieves tasks related to the ML methods using a bio-inspired structure based on an anthill with a queen (one per anthill), soldiers and several workers agents. The queen agent distributes the different tasks between the soldiers and builds the final result. The soldiers organize the requests and make use of the workers to tackle the problem. This agent architecture is relevant for applying possible load-balancing politics.

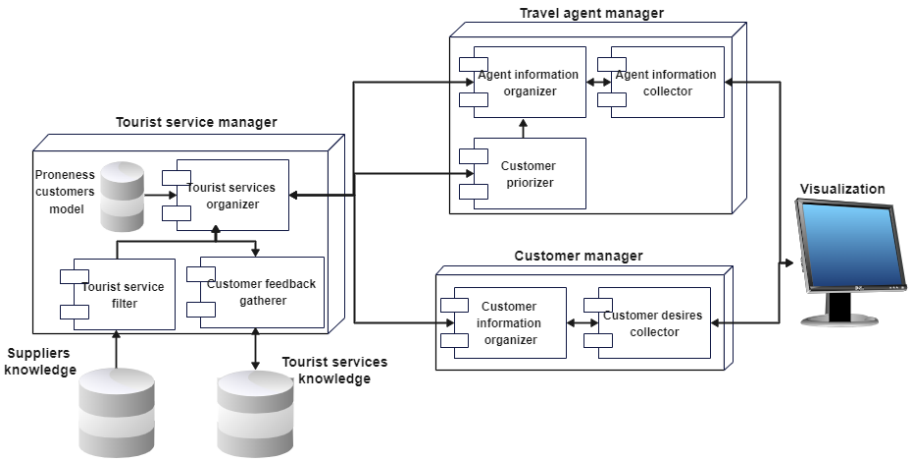


Fig. 4. Except of the *Tourist service manager*, *Travel agent manager*, and *Customer manager* modules

These politics consist of creating new soldiers and workers, or even new complete anthills to increment the capability of the system. More details about the MAS are provided in the next sections.

The *Tourist services knowledge* repository stores the information related to contracted tourist services for subsequent modifications or consultations made by the travel agents. This information refers to the management decisions and configurations made by the *Tourist service manager* module. Thus, it serves as a guide in the processes of composition or self-generation of tourist services. The feedback returned from the customers after the tourist service concluded is also considered.

The *Customer manager* module has as the main functionality to provide support to the customer to perform the necessary operations to create a tourist activity in real-time. This module communicates with the *Tourist services knowledge* and *Suppliers knowledge* repositories through the *Tourist service manager* module to make compositions of different elements to include in the complete tourist service. The created service is stored for later verification and approval by the travel agent.

The *Travel agent manager* module has as a purpose to serve as the internal central management point for the operations related to the tourist service creation. It is also in charge of the review of these activities created by customers through the *Customer manager* module. It is directly connected to the graphical interface, being able to take information about the requests made by the users. These requests are used to consult the *Tourist services knowledge* and the *Suppliers Knowledge* repositories through the *Tourist service manager* module and to obtain interesting tourist services for the user. This module can produce a tourist activity from scratch by selecting different possibilities. Note that the main difference for this module with respect to the previous one resides in the configuration of privileges of the end-user (i.e., a travel agent has more privileges than a customer).

Regarding the visual interface, it provides different configurations according to the two perspectives: the customer and the travel agent. For the first case, the visualization

supports the creation of travel services (i.e., a complete set of tourist activities). Graphical assistants try to alleviate the complexity of the travel service design by reducing customer decision-making through a finite set of guided options. When the customer approves the travel service, it is sent to the travel agency for its management: approval or modification. A series of exchanges take place between the customer and the travel agency until the customer confirms the tourist service firmly and in turn is approved. Once the services have been confirmed and the full payment has been verified, a process of sending the necessary documentation is executed for the client to enjoy the service. In addition, a user area is made available to the customer to facilitate the consultation of the travel information and its associated legal documentation. For the second case, similar functionalities are provided. However, travel agents have the initiative here, generating multiple travel services for the customers. Note that more details about the sale processes are addressed in the next sections.

3.2. Algorithm for the detection of potential customers

The *Tourist service manager* module contains a propensity purchase estimation algorithm. It classifies the potential customers according to their propensity to commit the book of a tourist service. This algorithm provides valuable information to the travel agency and enables the travel agent to differentiate between good and bad clients (customers who book a tourist service or not). The algorithm consists of three key elements: variable transformation, clustering, and classification. The first element prepares the inputs for the probability estimation, conducted by clustering and classification methods. The second element provides an initial purchase probability, and the last element refines the probability in some of the clusters. The algorithm's output categorizes the probability into five groups according to the conversion rate (i.e., the proportion of clients who finally book a tourist service).

The variable transformation element is carried out by Weight of Evidence (WoE) encoding [45]. This type of transformation and its benefits have been studied within the scope of credit scoring deeply [5, 47]. It helps deal with categorical and numerical data simultaneously; besides, it can lead to missing data with any additional technique. Traditionally, WoE encoding is used to separate bad clients from good clients, which fits perfectly with an algorithm of propensity purchase estimation. The equation of the WoE transformation is:

$$WoE = ln\frac{\%non_events}{\%events} \tag{7}$$

where *events* value refers to good clients, and *non-events* value represents bad clients.

One of the most powerful advantages of WoE for the current scope is that variables with too many discrete values lead to numeric variables, and they will be ordered according to a monotonic relationship to the dependent variable. WoE provides a numeric variable that regroups huge discrete values into densely populated categories that express information for the new category. The other great advantage that benefits the proposed solution is that WoE is a standardized value. The former allows using Decision Tree (DT) algorithms. Note that a tree-based algorithm can reduce its predictive potential with categorical variables with too many categories [15]. The latter allows using distance-based algorithms like *k-means* clustering algorithm, which needs variables with comparable values because it employs Euclidean distance [33].

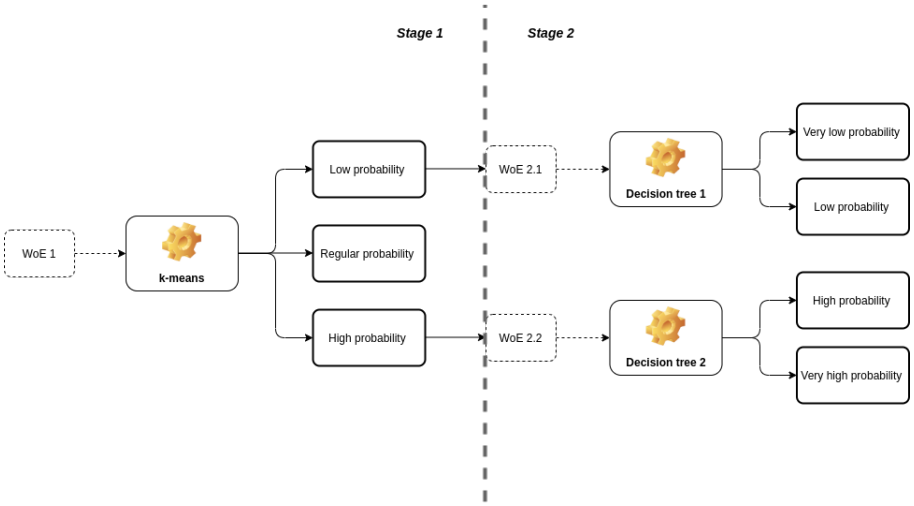


Fig. 5. Bi-stage model for the probability estimation of propensity to commit the book of a tourist service

The algorithm is a bi-stage model as depicted in Figure 5, that first applies *k-means* clustering to achieve three clusters with different conversion rates (low, regular, high). Secondly, for more accurate results, a DT classifier acts over the low and high clusters by subdividing them into low and very low, and high and very high respectively. Hence, the proposed algorithm eventually provides five output categories: very low, low, regular, high, and very high. This type of output is calculated on the base of the prior probability of propensity to purchase [37], and it offers an interpretable result for a travel agent.

The key to the model is how to group the variables in the two stages. In the first stage, WoE transformation is applied to the set of variables that contains information about how the customer is and where they come from. Examples of these kinds of variables are: means of contact (e.g., web, email, physical presence in the agency or phone), getting information for short term, or large term, the travel is related to a particular event (e.g., wedding or anniversary). With these variables, clustering is developed to first obtain three groups (low, regular, and high), which later will be split into the final desired five groups. Then, in the second stage, WoE transformation is applied to variables inherent in travel information like the number of places to be visited, trip duration, flexibility, final price, etc. Note that WoE encoding must be applied separately to low and high-probability clusters, even concerning the same variable, because the goal is to capture the patterns related to each independent cluster. Finally, two DT applied into low and high probability clusters subdivide those clusters into two groups each, resulting in five categories with different propensity purchases.

3.3. Bio-inspired MAS organization

Intelligent agents are independent entities that are used by *Pharaoh* to tackle the ML methods. They establish cooperative activities organized through the hierarchical architecture of a MAS.

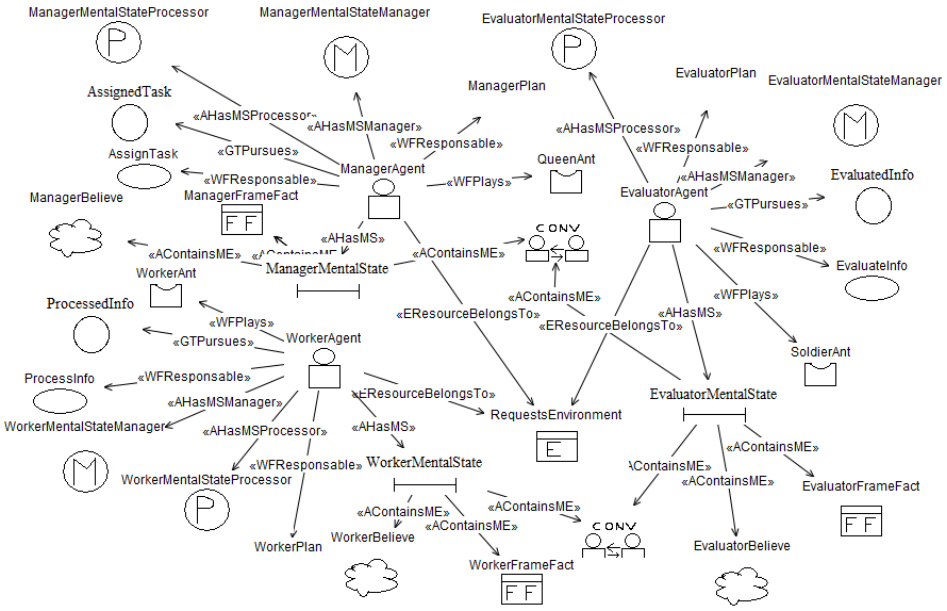


Fig. 6. Excerpt of the main entities involved in the anthill MAS

The proposed MAS consists of an anthill that comprehends agents playing three types of ant-roles: queen, soldiers, and workers (see Fig. 6). Therefore, it is organized following a bio-inspired ant social structure. The queen (i.e., the manager agent) is responsible for the anthill, receiving the requests to analyze. Notice that several requests can be sent to this agent. These requests come from the customers' preferences for a specific tourist service. Then, the queen agent assigns the activity of evaluating the elements of the request to one soldier agent (i.e., the evaluator agent). This agent is in charge of assigning the relevant information to the worker agents. Thus, this information is decomposed into more simple requests organized by the different features indicated by the customer. Worker agents process the information and apply the ML methods provided by the system. Notice that a copy of the ML model is available for each worker and they have access to the relevant information provided by the suppliers to configure the possible tourist packages that are offered to the customer. Once the worker agents have concluded their task, the soldiers join the result if necessary (protecting and supervising the result obtained by a worker or a set of them) and they return the result to the queen agent.

This process is based on the Belief-Desire-Intention (BDI) model [11] where each agent presents a goal or a set of them that must be satisfied to complete its life cycle. In this sense, the queen has the *assignedtask* goal, the soldiers have the *evaluatedinfo* goal and the workers have the *processedinfo* goal. All the goals usually include associated tasks that are actions to achieve. Agents have at least one task that solves their corresponding goals. These tasks are applied in the shared environment. In this case, the environment is formed by the requests of the possible customers. Each agent incorporates a mental state, its processor and manager, and a set of beliefs (motivations). The queen has rules and a plan to manage the requests, while soldiers present similar rules to distribute the

information about the requests between the workers. However, workers include the ML model in their mental states to achieve the evaluation of the information, a plan, and some simple rules to organize the process in the beliefs. Finally, interactions between the individuals follow the hierarchical structure which mainly consists of direct conversations. Notice that in this case, workers do not need to establish conversations with other workers since they tackle their commitment individually according to the orders of the soldier.

Regarding the design of the anthill model, it has been addressed through the INGENIAS agent methodology. Then, the resulting composition has been transformed to be compliant with the MESA framework. The conversations and interactions of the agents have been implemented following the Foundation for Intelligent Physical Agents (FIPA) standards [31].

3.4. Internal system processes

The *Pharaoh* framework presents three main processes according to its architecture. The first process encompasses the different actions achieved by the travel agent and the customer until the sale of a tourist service is completed. The second process refers to the feedback task. Finally, the third process comprehends the steps to update the information about rates, services, and other relevant information.

The sale of a tourist service process consists of a set of steps most of them common to both types of end-users (i.e., travel agent and customer). It occurs due to both roles interacting several times during the process (see Fig. 7). Initially, the end-users are logged into the system. If the users are customers, they can request a tourist service and ask the travel agents about it. However, if the users are travel agents, they can take the initiative (the customers are physical with them) or they answer the possible requests made by customers previously. Then, the system analyzes the customers to decide if they are prone to sale or not. In case they are prone, the travel agents prioritize their demands. If the customers respond to the answer provided by the travel agents and one of the tourist services is interesting to them, the system finishes the process of sending the documents with the transaction and the tourist service-related information. However, the customers could not respond to the tourist services stipulated by the travel agents. In this case, the process concluded. Moreover, the customers could not find interesting the offers made by the travel agents. In this case, the travel agents should produce new tourist services more adjusted to the desires of the customers.

The feedback process follows a set of steps where the customers introduce feedback about a tourist service previously bought and concluded (see Fig. 8). It is a simple and optional process carried out only by the customers where their opinion is stored in the system. The information provided is used to update the formulas for the recommendations that are considered by *Pharaoh* to select the best options.

The updating information process is the typical maintenance work in the system. It mainly modifies the stored knowledge in the two repositories: the *Suppliers knowledge* and the *Tourist services knowledge*. This task must be daily performed to maintain the *Pharaoh* framework up to date. This process is achieved by a complete Extract, transform and load (ETL) architecture that automatizes the steps to manage the actions and simplifies the task for the administrator. Note that the administrator is not considered an end-user by the system (see Fig. 7).

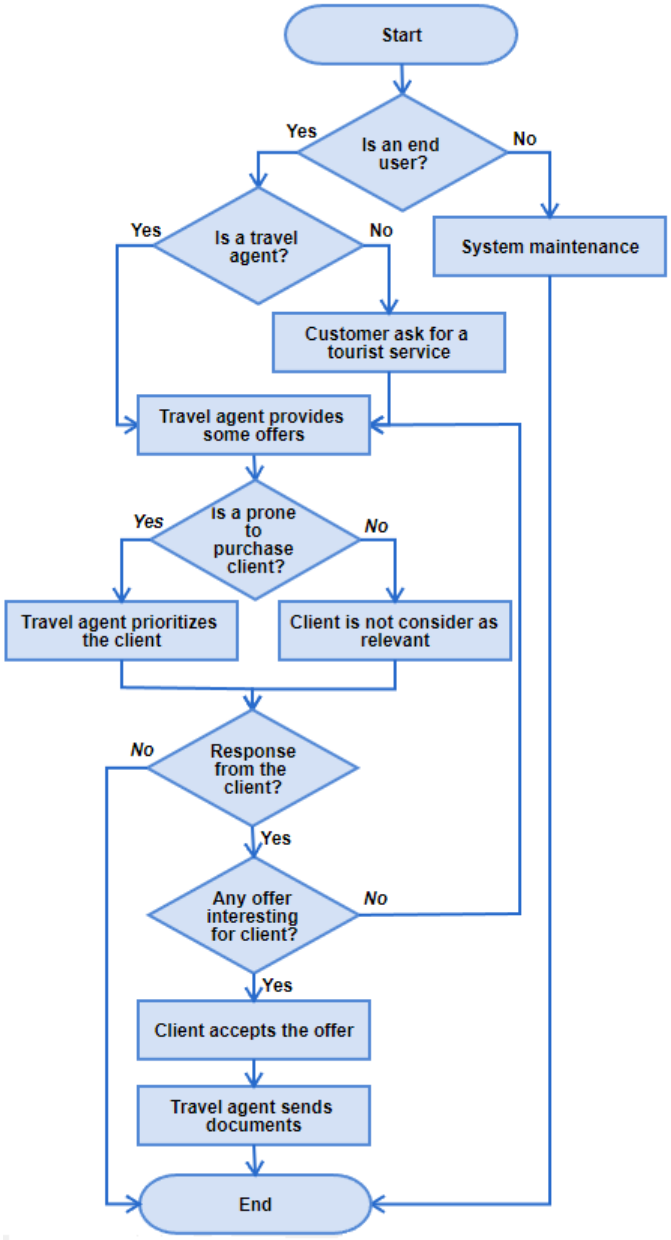


Fig. 7. Offer and sale processes of a tourist service provisioning

4. Experiments

This section details the experiments carried out to validate the performance of the *Pharaoh* framework. Three experiments have been considered and developed in a real environment.

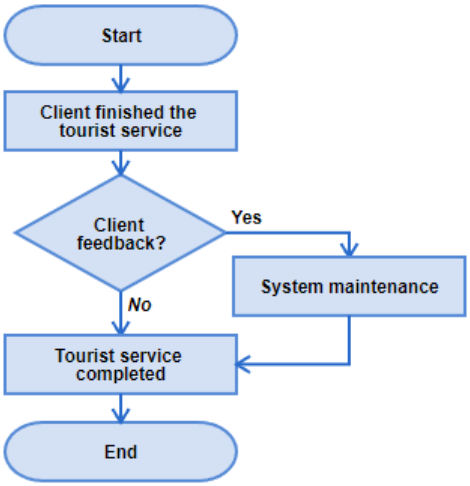


Fig. 8. Information updating process based on the feedback of the customer

JOF ASSOCIATES SLU also called Madox Viajes S.L. [1] was the selected travel agency used to develop the experiments. Notice that this is the first company where the system has been implanted in production once the development stage has been completed.

Regarding the MAS included in the system, it presents a basic configuration adapted to the needs of the company. These needs have been previously tested by adjusting the parameters before the presented experiments. Thus, the MAS comprises two anthills, with five soldiers and ten workers.

The first experiment covers the parameters configuration and the setup of the system. Moreover, a basic test is presented to evaluate the satisfaction level produced by the system between travel agents and customers according to the recommendations made. The second experiment details the performance of the detection of potential customers algorithm. A comparison with other methods of the state-of-the-art is included. The third experiment evaluates the performance of the system once it is implanted in the travel agency. Comparisons between the profits of the travel agency using and not using the *Pharaoh* framework are addressed. This fact illustrates the relevance of the system during the management of a travel agency.

4.1. Parameters configuration and initial validation of the system

The first experiment consists of a set of steps to show how the proposed system works. Initially, the parameters of the *Pharaoh* framework were configured to produce recommendations about tourist services to customers and travel agents. These values were provided by 3 experts in the domain from the selected travel agency (i.e., the people responsible for the company). Then, 10 customers (preserving their anonymity) and 3 travel agents of the company (i.e., some of the workers of the company) evaluated the obtained recommendations for their specific tourism desires. Finally, these actors provided feedback

[1] https://www.madoxviajes.com/

Table 1. Satisfaction values provided by the travel agents

Role	Satis VAS
Agent A	0.85
Agent B	0.93
Agent C	0.73
Mean* ± *S.D.	***0.84* ± *0.10***

Table 2. Satisfaction values provided by the customers

Role	Satis VAS
Customer 1	0.71
Customer 2	0.62
Customer 3	0.89
Customer 4	0.76
Customer 5	0.95
Customer 6	0.58
Customer 7	0.78
Customer 8	0.69
Customer 9	0.92
Customer 10	0.76
Mean* ± *S.D.	***0.77* ± *0.12***

about the performance of the system and their satisfaction with the obtained results. This satisfaction has been measured according to a Satisfaction Visual Analogue Scale (Satis VAS) [29]. This VAS was included in an anonymous document. The result of the VAS was converted to a range of values from 0 to 1 to obtain the resulting opinion.

Regarding the parameters configuration (see Section 3.1), the travel agency experts decided the next values for them. For θ_T. θ_A, and θ_E, the values were fixed to 0.3, 0.45 and 0.25 respectively. This decision was motivated by the assumption that customers give more importance to the hosting and the location than the rest of the features of the tourist service. Then, $w^C_{u,j}$ and $w^C_{e,j}$ were fixed to 0.4 and 0.6, giving a moderate relevance to the feedback provided by previous customers. Lastly, the weights of the elements of each category (transport, accommodation, and extras) are established. For the transport category, $w^T_{quality}$, $w^T_{stopovers}$, w^T_{time}, and w^T_{price} are configured to 0.3, 0.1, 0.2, and 0.4 respectively. This decision of the experts was motivated by the assumption that users usually consider very relevant the price of the transport over the other features. For the accommodation category, $w^A_{quality}$, $w^A_{location}$, $w^A_{services}$, and w^A_{price} are configured to 0.35, 0.25, 0.15, and 0.25 respectively. These values correspond to the assumption that users usually consider as relevant the price and the quality of the hostage unless the location and the provided services are also very important. For the extras category, $w^E_{quality}$, $w^E_{flexibility}$, $w^E_{services}$, and w^E_{price} are configured to 0.45, 0.15, 0.15, and 0.25 respectively. This decision follows the assumption that the users give more importance to the price and the quality of the extras.

Once the parameters of the system are completely fixed, the three travel agents make petitions to *Pharaoh* to obtain recommendations according to their preferences. These

preferences were selected as heterogeneous as possible to validate the system through a wide spectrum of possibilities.

Next, the customers provided indications of the system being assisted by the travel agents to solve possible problems related to the operation of the system. Travel agents also provided users with the best configurations obtained by the system. If a customer agrees with one of the offered tourist services, the system completes the corresponding booking process.

Finally, the anonymous document with the VAS was given to the participants. The results are shown in Table 1 and Table 2. The average satisfaction of the travel agents was 0.84, while the average of customers was 0.77. Similar Standard Deviations were obtained in both cases. Thus, the global average satisfaction of the system was 0.80 ± 0.11.

In conclusion, it can be said that the result is very acceptable from both perspectives (customers and travel agents). Furthermore, customers are more pressing and demanding than the travel agents in the evaluation of the satisfaction produced by the recommendations made by the *Pharaoh* framework.

4.2. Propensity purchase estimation algorithm validation.

This experiment addresses the performance of the proposed propensity purchase estimation model (*bi-stage* model). The proposal is compared with the most typical state-of-the-art ML classifiers [39, 19]: Random Forest (RF), Gradient Boosting (GB), k-Nearest Neighborhood (kNN) and Logistic Regression (LR).

JOF ASSOCIATES SLU has provided the dataset for the development of this experiment. It contains about 10^4 instances presenting an initial conversion rate of 14%, that is, the percentage of customers who finally purchased the tourist service (i.e., the data are unbalanced). Each instance includes a variety of attributes: contact details, contact channel, service, travel destination, type of trip, etc. In this particular case, more precise information of the data is not available because of confidentiality. All the selected algorithms are trained with 80% of the whole data and tested using the remaining 20%. This division is carried out with stratified sampling. The hyper-parameters of the ML alternatives are selected by 10-fold-cross-validation. The output of each algorithm is presented into five categories attending to the output probability distribution on the train set. The performance is measured in terms of conversion rate and relative frequency on each classification group, expressed in percentage of the overall total. The predictive power is evaluated by checking the conversion rate in the different groups, and the frequency provides the significance of the group. Finally, to assess the possible overfitting train and test agreement are evaluated.

The results are shown in Table 3. The RF has a poor agreement between train and test, in both frequencies and conversion rates. Indeed, one of the groups has no representation in the test dataset. The kNN has similar behavior in train and test agreement, with differences up to 21%. This situation makes no sense to evaluate any other performance measure. The GB has a generally good agreement between train and test instances, except in conversion rates for *high* and *very high* categories with differences higher than 20%. These differences make, like in RF and kNN, no sense to evaluate the rest of the evaluation items. The LR has the best agreement between train and test, in both frequency and conversion rates. The data distribution is highly unbalanced throughout the categories,

Table 3. Performance of the ML classification algorithms in terms of relative frequency (freq.), expressed in percentage about the overall total, and conversion rate (c.rate), for each category

		RF		GB		kNN		LR		Bi-stage model	
		freq.	c.rate	freq.	c.rate	freq.	c.rate	freq.	c.rate	freq.	c.rate
Train	very low	75.0	0.0	74.2	5.8	0.0	100.0	72.5	11.5	16.6	1.7
	low	9.9	0.0	10.3	25.2	54.2	0.0	9.0	17.8	17.1	5.1
	regular	9.9	89.1	10.3	34.5	29.8	20.6	13.2	19.6	44.7	10.9
	high	3.9	100.0	4.1	63.4	11.9	42.2	4.2	26.7	15.8	32.8
	very high	1.2	100.0	1.0	87.3	4.1	100.0	1.1	28.2	5.9	48.0
Test	very low	57.5	4.9	73.6	7.0	0.0	-	72.3	6.7	17.0	2.5
	low	17.1	16.5	10.0	25.0	75.2	7.5	2.0	25.0	17.0	6.7
	regular	25.2	30.9	9.8	28.8	17.4	26.7	5.8	22.9	46.1	10.2
	high	0.0	-	4.8	41.4	5.3	40.6	7.3	34.1	14.6	34.2
	very high	0.2	100.0	1.8	63.6	2.0	58.3	12.6	35.5	5.4	51.5

where *low probability* group comprises more than 70% of the data. Besides, the five categories do not have enough distance between them in the conversion rate term, 2% in some cases, making it difficult to differentiate between close groups. Eventually, the proposed *bi-stage* model has an excellent agreement between train and test instances, with a maximum deviation of 3%. The data distribution is the most balanced of the models tested, where the *regular* category is the densest with 45% of the data and the *very high* the lower dense with 6% of the data for train, and 46% and 5% for test respectively. The probability categories obtained are different with five representative probability values, and it is validated in the test dataset with minimum deviations.

In conclusion, the experiment carried out shows that this particular problem needs an ad-hoc approach, such as the *bi-stage* model presented. This solution is adequate to solve the propensity to purchase problem provided by *JOF ASSOCIATES SLU*. The *bi-stage* model is the one that best classifies in the different probability categories with balanced frequencies. In addition, it is the one that provides the best concordance between training and testing, thus guaranteeing the non-existence of overfitting. Hence, according to the performance requirements, the proposed method serves better than the state-of-the-art classifier for the current aim.

4.3. Efficiency evaluation of the complete system

The last experiment checks the efficiency of *Pharaoh* in a real situation. The setup of the system with the parameters configuration has been maintained, being the same ones as the first experiment.

The experiment starts by organizing randomly and proportionally 184 potential customers into two groups: *Traditional* and *Innovators*, for making comparisons between them. All of them are looking for the same type of tourist services: *National Coasts*, *National Islands*, *Dubai-Maldives*, and *Safaris*. In addition, web canal and other canals such as phone calls, family recommendations, in-person visits to the office, etc., are considered. Note that these services comprehend a set of specific tourist services.

Table 4. Summary of the obtained results for *Traditional* and *Innovators* groups for each tourist service. n is the number of potential customers in each service and group. Time is the average time (in hours) each travel agent spends processing the tourist service until it is accepted or dismissed. The conversion rate (c.rate) and the number of recommendations (Rec.) are presented

		Traditional				**Innovators**			
	Services	n	Time	c.rate	Rec.	n	Time	c.rate	Rec.
Web	N. Coasts	20	3.2	15.0	2	20	2.9	25.0	3
	N. Islands	20	5.2	10.0	2	20	4.8	15.0	3
	Dubai-Maldives	16	10.1	6.3	1	16	9.6	12.5	2
	Safaries	12	11.2	16.7	1	12	10.7	16.7	2
Other	N. Coasts	8	4.2	37.5	1	8	2.7	50.0	2
	N. Islands	8	5.7	25.0	0	8	3.0	37.5	1
	Dubai-Maldives	4	12.0	25.0	1	4	8.0	50.0	1
	Safaries	4	12.5	25.0	0	4	9.0	25.0	1

Traditional group consists of 92 customers who were attended maintaining the traditional way that the company has been used (i.e., without the support of the *Pharaoh* framework). On the other hand, *Innovators* group consists of 92 customers who attended using the *Pharaoh* framework to carry out the requests made by the users and the configurations provided by the travel agents. Both groups worked with the same travel agents to minimize possible variations in the results.

The experiment considers three main variables: the average time (in hours) each agent spends processing the tourist service offered to customers, the conversion rate, and the number of recommendations. These results are presented in Table 4 for both groups.

For *Traditional* group, the results show that the average time to manage the services is less in national destinations. That is, the customers are more demanding when the travel destination is in the same country. The conversion rate fluctuates in the web canal, having a low value for the service *Dubai-Maldives* (i.e., only a few customers complete the purchase), while *Safaris* and *National Coasts* have a high percentage. A total of 8 recommendations have been obtained for this group.

For *Innovators* group, the service *Dubai-Maldives* has the lowest conversion rate and *National Coasts* has the highest one in the web canal. Relevance differences in the number of inverted hours, the number of recommendations, and the conversion rate between *Traditional* and *Innovators* have been detected. The system incremented acceptably the conversion rate, the number of hours was reduced significantly, and the number of recommendations was also higher. A total of 15 recommendations have been obtained for the *Innovators* group. The changes in the conversion rate and the number of hours spent are motivated by the propensity purchase estimation algorithm. It allowed the selection of the best customers for the travel agents, discarding the customers with less interest in the sale. Moreover, the visual information provided by the system, and the precise recommendations could have also done that customers more easily accepted the tourist service.

In conclusion, *Pharaoh* has demonstrated that it works perfectly with the daily issues of a travel company. Moreover, it could increase the conversion rate that is translated into

profits for the company, and reduce the number of hours until the customer accepts or dismiss the different possibilities offered for a specific tourist service. This fact is translated into less stressful situations for employees and more agility in the booking process.

5. Conclusions

The *Pharaoh* framework, a system devoted to completely covering the management of the sales of tourist services from the side of the travel agent and the customer, has been presented. It includes as a novelty a bio-inspired MAS based on the structure of anthills and a proneness customer model based on ML techniques. This latter can classify customers according to their proneness of completing a booking (i.e., it detects the most interesting customers). The model consists of three elements: variable transformation, clustering, and classification. The output categorizes the probability into five groups based on the conversion rate.

The system has been tested empirically through several experiments. They have illustrated that *Pharaoh* enhances the performance of a travel agency, increments the quality of the service, and generates interesting profits. Thus, it can be said that the framework provides very desirable functionalities and new insights in the tourism domain thanks to the proposed bio-inspired MAS and the ML techniques.

Future research will focus on the communication between the customer and the travel agent, trying to improve it by simplifying the steps during the conversation. The introduction of improvements at the graphical level both during the sale and feedback processes will also be considered. Moreover, in the domain of intelligent agents, it could be very interesting to include a complete MAS. It could cover all the functionalities of the system to distribute and emulate the competencies of travel agents. This will lead to developing a completely independent intelligent framework that could interact with customers becoming a virtual travel agent.

Acknowledgments. This work has been partially supported by the Spanish MICINN under the XMIDAS project (PID2021-122640OB-I00) and the donation of the Titan V GPU by NVIDIA. Special thanks to JOF ASSOCIATES INT S.L.U and Madoxviajes S.L.

References

1. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE transactions on knowledge and data engineering 17(6), 734–749 (2005)
2. Adriana, S.M.: The customer orientation principle - a prerequisite for increased efficiency in tourism. Ovidius University Annals, Economic Sciences Series 18(1), 405–408 (2018)
3. Al Fararni, K., Nafis, F., Aghoutane, B., Yahyaouy, A., Riffi, J., Sabri, A.: Hybrid recommender system for tourism based on big data and ai: A conceptual framework. Big Data Mining and Analytics 4(1), 47–55 (2021)
4. Alrasheed, H., Alzeer, A., Alhowimel, A., Althyabi, A., et al.: A multi-level tourism destination recommender system. Procedia Computer Science 170, 333–340 (2020)
5. Anderson, R.: The credit scoring toolkit: theory and practice for retail credit risk management and decision automation. Oxford University Press (2007)

6. Bakar, N.A., Rosbi, S.: Effect of coronavirus disease (covid-19) to tourism industry. International Journal of Advanced Engineering Research and Science 7(4), 189–193 (2020)
7. Bedi, P., Agarwal, S.K., Jindal, V., Richa: Marst: Multi-agent recommender system for e-tourism using reputation based collaborative filtering. In: Databases in Networked Information Systems: 9th International Workshop, DNIS 2014, Aizu-Wakamatsu, Japan, March 24-26, 2014. Proceedings 9. pp. 189–201. Springer (2014)
8. Bergenti, F., Caire, G., Monica, S., Poggi, A.: The first twenty years of agent-based software development with jade. Autonomous Agents and Multi-Agent Systems 34(2), 1–19 (2020)
9. Cabezas, J., Fernandez-Isabel, A., Fernández, R.R., González-Fernández, C., Alonso, A., de Diego, I.M.: Bio-inspired agent-based architecture for fraud detection. In: Proceedings of the 2020 3rd International Conference on Information Management and Management Science. pp. 67–71 (2020)
10. Camilleri, M.A.: The tourism industry: An overview. Travel marketing, tourism economics and the airline product pp. 3–27 (2018)
11. De Silva, L., Meneguzzi, F.R., Logan, B.: Bdi agent architectures: a survey. In: Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI), 2020, Japão. pp. 4914–4921 (2020)
12. Dinu, A.M., et al.: The importance of transportation to tourism development. Academic Journal of Economic Studies 4(4), 183–187 (2018)
13. Fernández-Isabel, A., Fuentes-Fernández, R., de Diego, I.M.: Modeling multi-agent systems to simulate sensor-based smart roads. Simulation Modelling Practice and Theory 99, 101994 (2020)
14. Forouzandeh, S., Berahmand, K., Nasiri, E., Rostami, M.: A hotel recommender system for tourists using the artificial bee colony algorithm and fuzzy topsis model: A case study of tripadvisor. International Journal of Information Technology & Decision Making 20(01), 399–429 (2021)
15. Friedman, J.H.: The elements of statistical learning: Data mining, inference, and prediction. springer open (2017)
16. Garcia, I., Sebastia, L., Onaindia, E., Guzman, C.: A group recommender system for tourist activities. In: International conference on electronic commerce and web technologies. pp. 26–37. Springer (2009)
17. Garro, A., Mühlhäuser, M., Tundis, A., Baldoni, M., Baroglio, C., Bergenti, F., Torroni, P., et al.: Intelligent agents: Multi-agent systems. Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics 315 (2018)
18. Garro, A., Mühlhäuser, M., Tundis, A., Mariani, S., Omicini, A., Vizzari, G.: Intelligent agents and environment. Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics p. 309 (2018)
19. González, S., García, S., Del Ser, J., Rokach, L., Herrera, F.: A practical tutorial on bagging and boosting based ensembles for machine learning: Algorithms, software tools, performance study, practical perspectives and opportunities. Information Fusion 64, 205–237 (2020)
20. González-Fernández, C., Cabezas, J., Fernández-Isabel, A., Martín de Diego, I.: Combining multi-agent systems and subjective logic to develop decision support systems. In: International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems. pp. 143–157. Springer (2020)
21. Hamid, R.A., Albahri, A.S., Alwan, J.K., Al-qaysi, Z., Albahri, O.S., Zaidan, A., Alnoor, A., Alamoodi, A., Zaidan, B.: How smart is e-tourism? a systematic review of smart tourism recommendation system applying data management. Computer Science Review 39, 100337 (2021)
22. Herzog, D., Laß, C., Wörndl, W.: Tourrec: a tourist trip recommender system for individuals and groups. In: Proceedings of the 12th ACM Conference on Recommender Systems. pp. 496–497 (2018)

23. Hsu, P.F., Lan, K.Y., Tsai, C.W.: Selecting the optimal vendor of customer relationship management system for medical tourism industry using delphi and ahp. International Journal of Enterprise Information Systems (IJEIS) 9(1), 62–75 (2013)
24. Intayoad, W., Kamyod, C., Temdee, P.: Reinforcement learning for online learning recommendation system. In: 2018 Global Wireless Summit (GWS). pp. 167–170. IEEE (2018)
25. Kang, E.y., Kim, H., Cho, J.: Personalization method for tourist point of interest (poi) recommendation. In: Knowledge-Based Intelligent Information and Engineering Systems: 10th International Conference, KES 2006, Bournemouth, UK, October 9-11, 2006. Proceedings, Part I 10. pp. 392–400. Springer (2006)
26. Kazil, J., Masad, D., Crooks, A.: Utilizing python for agent-based modeling: the mesa framework. In: International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction and Behavior Representation in Modeling and Simulation. pp. 308–317. Springer (2020)
27. Kler, B.K.: Tourism and restoration. In: Philosophical issues in tourism, pp. 117–134. Channel View Publications (2009)
28. Loh, S., Lorenzi, F., Saldaña, R., Licthnow, D.: A tourism recommender system based on collaboration and text analysis. Information Technology & Tourism 6(3), 157–165 (2003)
29. Lynskey, S.J., Ling, F., Greenberg, A.M., Penny-Dimri, J.C., Sutherland, A.G.: The influence of patient resilience and health status on satisfaction after total hip and knee arthroplasty. The Surgeon 19(1), 8–14 (2021)
30. Meehan, K., Lunney, T., Curran, K., McCaughey, A.: Context-aware intelligent recommendation system for tourism. In: 2013 IEEE international conference on pervasive computing and communications workshops (PERCOM workshops). pp. 328–331. IEEE (2013)
31. Melo, L.S., Sampaio, R.F., Leão, R.P.S., Barroso, G.C., Bezerra, J.R.: Python-based multiagent platform for application on power grids. International transactions on electrical energy systems 29(6), e12012 (2019)
32. Moguerza, J.M., Martín-Duque, C., Fernández-Muñoz, J.J.: The importance of service quality as an instrument for client customization: a methodological and practical approach within the hotel sector. Quality & Quantity pp. 1–12 (2021)
33. Mohamad, I.B., Usman, D.: Standardization and its effects on k-means clustering algorithm. Research Journal of Applied Sciences, Engineering and Technology 6(17), 3299–3303 (2013)
34. Moliner, M.A., Sánchez, J., Rodríguez, R.M., Callarisa, L.: Relationship quality with a travel agency: The influence of the postpurchase perceived value of a tourism package. Tourism and Hospitality Research 7(3-4), 194–211 (2007)
35. Moreno, R., Viajes, M., Fernández-Isabel, A., de Diego, I.M., Moguerza, J.M., Lancho, C., Cuesta, M.: Automatic detection of potential customers by opinion mining and intelligent agents. In: 2022 17th Conference on Computer Science and Intelligence Systems (FedCSIS). pp. 93–101. IEEE (2022)
36. Novikova, N.G., Kortunov, V.V., Krasnova, O.N., Zorina, N.M., Kireenkova, Z.A., et al.: A client with special needs in tourist services. Journal of Environmental Management and Tourism 6(2), 338 (2015)
37. Pons, J., Granados, A., Espinas, J., Borras, J., Martin, I., Moreno, V.: Assessing open heart surgery mortality in catalonia (spain) through a predictive risk model. European journal of cardio-thoracic surgery: official journal of the European Association for Cardio-thoracic Surgery 11(3), 415–423 (1997)
38. Ray, B., Garain, A., Sarkar, R.: An ensemble-based hotel recommender system using sentiment analysis and aspect categorization of hotel reviews. Applied Soft Computing 98, 106935 (2021)
39. Sen, P.C., Hajra, M., Ghosh, M.: Supervised classification algorithms in machine learning: A survey and review. In: Emerging technology in modelling and graphics, pp. 99–111. Springer (2020)
40. Siswahyudi, P., Kurniawan, T.A., Sugiarto, V.: Agent-oriented methodologies comparison: A literature review. Advanced Science Letters 24(11), 8710–8716 (2018)

41. Subramaniyaswamy, V., Logesh, R., Vijayakumar, V., Keerthana, K., Rakini, K., Swarnamalya, B.: Dynamo: dynamic multimodal route and travel recommendation system. In: 2018 International Conference on Recent Trends in Advance Computing (ICRTAC). pp. 47–52. IEEE (2018)
42. Tureac, C.E., Turtureanu, A.: Types and forms of tourism. Acta Universitatis Danubius. Œconomica 4(1) (2010)
43. Uddin, M., Wang, Q., Wei, H.H., Chi, H.L., Ni, M.: Building information modeling (bim), system dynamics (sd), and agent-based modeling (abm): Towards an integrated approach. Ain Shams Engineering Journal 12(4), 4261–4274 (2021)
44. Wang, J., Hong, Y., Wang, J., Xu, J., Tang, Y., Han, Q.L., Kurths, J.: Cooperative and competitive multi-agent systems: From optimization to games. IEEE/CAA Journal of Automatica Sinica 9(5), 763–783 (2022)
45. Weed, D.L.: Weight of evidence: a review of concept and methods. Risk Analysis: An International Journal 25(6), 1545–1557 (2005)
46. Zapata, G., Murga, J., Raymundo, C., Dominguez, F., Moguerza, J.M., Alvarez, J.M.: Business information architecture for successful project implementation based on sentiment analysis in the tourist sector. Journal of Intelligent Information Systems 53(3), 563–585 (2019)
47. Zdravevski, E., Lameski, P., Kulakov, A.: Weight of evidence as a tool for attribute transformation in the preprocessing stage of supervised learning algorithms. In: The 2011 international joint conference on neural networks. pp. 181–188. IEEE (2011)
48. Zheng, X., Luo, Y., Sun, L., Zhang, J., Chen, F.: A tourism destination recommender system using users' sentiment and temporal dynamics. Journal of Intelligent Information Systems 51(3), 557–578 (2018)

Raúl Moreno Izquierdo was born in Madrid, Spain in 1979. He holds a Master's in Decision Engineering in 2004 and Computer Science in 2003, from Universidad Rey Juan Carlos and a partially completed PhD in Computer Science. Since 2020 he is Associated Professor at Universidad Carlos III de Madrid with main topics as programming and geographical information systems. From 2013 to 2021, he was the IT director at Madox Viajes, overseeing technology decisions, online booking tool design, data analysis, CRM implementation, digital marketing, and security audits. Previously, he worked as a business consultant, presales and software developer in multiple IT companies. His research interests include methods, processes, and tools for Data Science mainly in artificial vision, and security, with a special interest in risk analysis methods.

Alberto Fernández-Isabel was born in Toledo, Spain in 1984. He received a PhD in Computer Science from Complutense University of Madrid (UCM) in 2015. He obtained a scholarship at the Spanish National Research Council (CSIC) as a technical assistant. He has been working for several years on European and national projects as a predoctoral and postdoctoral researcher. Since 2019 he is Assistant Professor at the Higher Technical School of Computer Engineering (ETSII) at Rey Juan Carlos University (URJC). He has authored more than 30 scientific articles and books. He completes his background with a Master´s degree in Artificial Intelligence and a Master´s degree in Information Systems. His research interests include intelligent agents, machine learning, data visualization, and natural language processing in various application domains, including distributed programming, sentiment analysis, agent-based collaboration and negotiation, smart cities, and simulations.

Víctor Aceña has PhD in Information and Communication Technologies (URJC). Graduated in Mathematics from the National University of Distance Education (UNED). Master's degree in Computational Statistical Information Processing from the Complutense University of Madrid and the Polytechnic University of Madrid (UCM-UPM). Junior Assistant Professor and researcher at the Data Science Laboratory (DSLAB) of the URJC. His main area of work is the design of advanced sampling methodologies to adjust statistical models in the field of machine learning, particularly in incremental and ensemble learning scenarios.

Isaac Martín de Diego was born in Campaspero, Valladolid, Spain in 1973. He received a PhD degree in Mathematical Engineering from Carlos III de Madrid University in 2005 (Extraordinary Doctorate Award). Since 2023 he is a full professor at the Higher Technical School of Computer Engineering at Rey Juan Carlos University (Associate Professor from 2018). He is the co-founder of the Data Science Laboratory and Head of the Master´s in Data Science at Rey Juan Carlos University. He has been head of the ERICSSON Chair on Data Science applied to 5G. He is the author of more than 100 articles. His research interests include methods, processes, and tools for Data Science in various application domains: artificial vision, opinion mining, security, and biostatistics, with a special interest in Machine Learning algorithms and a combination of information methods.

Javier M. Moguerza is Full Professor at Rey Juan Carlos University. His professional expertise is focused on Computational and Applied Mathematics, Optimization and Data Science. Regarding these topics, he has relevant publications and regularly coordinates and participates in publicly and privately funded research projects. He served as a Full Member of the Global Young Academy during the period 2011-2016, and has been a Full Academician of the Young Academy of Spain since its foundation, in 2019, and until 2024.

Received: November 20, 2023; Accepted: May 20, 2024.

Computer Science and Information Systems 21(4):1269–1292 https://doi.org/10.2298/CSIS231224036P

PTB-FLA Development Paradigm Adaptation for ChatGPT

Miroslav Popovic[1], Marko Popovic[2], Ivan Kastelan[1], Miodrag Djukic[1], and Ilija Basicevic[1]

[1] University of Novi Sad, Faculty of Technical Sciences, Trg D. Obradovica 6, 21000 Novi Sad, Serbia
{miroslav.popovic, ivan.kastelan, miodrag.djukic, ilija.basicevic}@rt-rk.uns.ac.rs
[2] RT-RK Institute for Computer Based Systems, Narodnog fronta 23a, 21000 Novi Sad, Serbia
marko.popovic@ rt-rk.com

Abstract. The Python Testbed for Federated Learning Algorithms (PTB-FLA) is a simple Python FL framework that is targeting edge systems and is by its design easy to use by human ML&AI developers. The original PTB-FLA development paradigm intended for humans consists of the four phases (producing the sequential code, the federated code, the federated code with callbacks, and the PTB-FLA code, respectively), and hence dubbed the four-phases (development) paradigm, was validated in the case study on the logistic regression. In this paper, we adapted the original paradigm into the two new paradigms for ChatGPT, named the adapted four-phases paradigm and the adapted two-phases paradigm, respectively. In tune with its name, the latter consists of two phases (producing the sequential and the PTB-FLA code, respectively). We successfully validated both new paradigms using the same case study on logistic regression that was used for the original paradigm. The results are positive and encouraging as the resulting program codes are of better quality than the codes solely made by humans using the original paradigm.

Keywords: Decentralized Intelligence, Federated Learning, Python, ChatGPT, Software Development.

1. Introduction

Contemporary Internet in a simplified view can be seen as the so-called cloud-fog-edge continuum (or hierarchy) with various devices (laptops, smart phones, smart sensors, actuators, IoTs, robots, etc.) connected at its edge. This complex system (or system of systems) is highly heterogenous (as there are various kinds of cloud, fog, and edge systems/networks) and dynamic (as devices, or complete systems, may dynamically connect/disconnect i.e., join, and leave, or reconfigure while being connected). Developing applications for this environment without tools is both tedious and costly.

Both academia and industry are aware of this issue, and one of many ongoing projects trying to address it is the ongoing EU Horizon 2020 project TaRDIS [1], which aims to create a toolbox for easy programming of innovative applications, such as: (1) multi-level smart electrical vehicles charging, (2) privacy-preserving learning through

decentralised training in smart homes, (3) distributed navigation concepts for LEO satellite constellations, and (4) highly resilient factory shop floor digitalization to be implemented in a real factory comprising several production lines, a warehouse, an intralogistics fleet of robots, etc. These systems must use some ML/AI algorithms and be trustworthy and responsible.

The Python Testbed for Federated Learning Algorithms (PTB-FLA) [2] is being developed within TaRDIS as a framework for developing federated learning algorithms (FLAs) i.e., as a runtime environment for FLAs under development. PTB-FLA was written in pure Python to keep application footprint small (to fit to IoTs) and to keep its installation simple. The PTB-FLA was designed with a simple API based on Single Program Multiple Data (SPMD) pattern, to be easy to use by ML&AI developers who do not need to be professional programmers, and as such, also amenable to ChatGPT.

Initially, PTB-FLA was validated and illustrated by three simple examples that were developed by humans using the ad hoc approach to directly write the decentralized PTB-FLA code [2]. For more complex applications, a more natural and feasible approach is to start from a given referent sequential code (either legacy or open source or newly crafted) and transform it into the semantically equivalent PTB-FLA code. Recently, PTB-FLA four-phases development paradigm appeared as a systematic guide for humans to transform a sequential code into the PTB-FLA code, and it was validated in a case study where the sequential logistic regression Python code [3] was transformed into the PTB-FLA codes (for both centralized and decentralized FLAs) [4].

More recently, PTB-FLA has been shown to be amenable to AI by successful redevelopment of the three simple examples from [2] using the ad hoc approach, where ChatGPT was given the context comprising the input sequential code and the target function main and was asked to create the target callback functions [5].

In the next three subsections we present the problem formulation and its solution, the paper achievements, and the paper organization.

1.1. Problem Formulation and Its Solution

The problem that we tried to solve in this paper, was how to adapt the PTB-FLA development paradigm used by humans for ChatGPT in order to (i) minimize the human labor by delegating a part of the job to ChatGPT, and at the same time to (ii) minimize the ChatGPT input by creating minimal contexts (i.e., prompts), because commercial AI services (like GPT-4) are charged according to the input they processed.

We solved that problem experimentally by using the simple iterative development process, which is described in Section 2 in detail. Here is a short summary: we used the text (guidelines) of the original development paradigm to create the initial ChatGPT context, and then in each iteration we used feedback from ChatGPT (in the form of questionnaire answers ChatGPT provided) to manually modify the context i.e., to adapt the paradigm, aiming towards the minimal context without redundant parts. In each iteration, the current paradigm (given in the current context) was also validated, because it had to produce the output PTB-FLA code that is semantically equivalent to the input sequential code. The application code that was used for this purpose is the logistic regression case study from [4].

1.2. Paper Achievements

As a continuation of [4] and [5], we firstly adapted the four-phases development paradigm originally devised for humans to guide ChatGPT to successfully develop the same algorithms as in [4] (to save space, here we present only the centralized algorithm development). Secondly, we adapted the four-phases paradigm into a two-phases development paradigm. As the first phase is always done by humans, the two-phases paradigm is rather close to the ideal solution where developers in the second phase immediately get the complete target PTB-FLA code. (All the paradigms are presented in Section 3.).

We would like to emphasize that both adapted paradigms (the four-phases and the two-phase) are original contributions that are presented for the first time in this paper, and both should be treated unbiasedly as both have their own strengths and weaknesses. The four-phases paradigm requires more human labor but provides better traceability (somewhat like gray box testing), whereas the two-phase paradigm requires minimal (almost no) labor but is less traceable (being based on the black box approach) and therefore riskier. The last statement should be taken as a simplified one and more as the authors' expectation, because of the limitations of the experimental validation made in this paper (as discussed in the paper).

Both development paradigms were experimentally validated, see Sections 4 and 5. In the experimental validation, which was made in early summer 2023, ChatGPT was using the well-known GPT-3.5 model. We report on the adapted development paradigms performance in terms of human labour and size of ChatGPT context needed to develop the logistic regression PTB-FLA code in Section 6.

To the best of authors knowledge this is the first paper that formulated the problem and its solution of this kind and therefore we could not compare our paper with similar papers of other authors, so originality came at the price of lack of closely related work. However, we did our best to cover a more broadly related work in Section 7.

1.3. Paper Organization

The paper is organized as follows. Section 2 presents the development process, Section 3 presents the adapted development paradigms, Section 4 presents the four-phases paradigm validation, Section 5 presents the two-phases paradigm validation, Section 6 presents the adapted paradigms performance, Section 7 presents related work, and Section 8 concludes the paper.

2. Development Process

In this paper, to adapt and validate development paradigms (see Section 1.1), we used the simple iterative development process, which comprises the following steps: (1) specify the context (i.e., the prompt), (2) ask ChatGPT to complete the assignment, (3) test the generated code, (4) if the code successfully passes testing, stop, else if the code is close to what is expected and contains small number of simple bugs, manually modify

the code (rational is further discussed below) and return to step 3, else return to step 1. We tried to create minimal contexts with the goal to minimize the overall effort/cost.

This paragraph is a short digression explaining the rational mentioned in the step 4 above. Initially, our intention was to make ideal contexts that would lead to ideal results. So, whenever the output code failed to pass the testing, we were returning to the step 1 and trying to refine the context. By doing this we discovered that ChatGPT exhibits one unexpected behaviour: given some context it may produce the code that is rather close to what is expected, but after giving it an extended context it may produce the code that is more far from what is expected than the previous one (see the paragraph below). Therefore, we switched to the following policy: if the output code is tweakable (i.e., has small number of bugs), then it is acceptable, and we consider the corresponding prompt sufficient.

We do not have an explanation for the unexpected behaviour mentioned above. This was happening both in the same session and in different sessions. A major difficulty in using ChatGPT is that by its design it tries not to repeat itself, so classical approach to testing with repetitive tries does not work for it.

For the sake of this research, to get an estimation of context quality (minimality), after an example was verified, we ask ChatGPT for feedback by asking it to complete the questionnaire comprising the following three questions: (i) On the scale from 1 to 10, where 10 is the hardest, how would you rate this assignment? (ii) What was the most important part of the context that you used to complete the assignment? and (iii) What was the redundant part of the context?

At the end of this section, we would like to clarify that we did not make any attempt to prevent ChatGPT hallucinations/confabulations, but if this happens and results in the generated code with bugs, this will be discovered during code testing (the phase 3 of the process outlined above), and this code will be rejected i.e., thrown away. However, preventing ChatGPT hallucinations/confabulations is an interesting research topic and a possible item of our future work.

3. Development Paradigms

In this section we present the PTB-FLA development paradigms. The next three subsections present the original four-phases paradigm that was initially devised for humans [4], the four-phases paradigm' phases, and the adapted two-phases paradigm for AI, respectively (the adapted four-phases paradigm adaptation is explained at the end of Section 3.3).

3.1. Original Four-Phases Paradigm for Humans

The original four-phases development paradigm [4] is primarily intended to serve as a FLA developer guide through the process of developing a target FLA using PTB-FLA, which we call the FLA development process. The input to this process is the Python sequential program code of target AI/ML algorithm, whereas the output from this process is the PTB-FLA code with the same semantics, which means that for given input

data it produces the same output data with some tolerance *e*. The tolerance *e* is typically some small error value (ideally zero).

Of course, the output PTB-FLA code must be compliant with the PTB-FLA programming model which is a restricted programming model that imposes the following two restrictions: (1) using the Single Program Multiple Data (SPMD) pattern, and (2) specifying code for server and client roles in form of callback functions. Obviously, there are many ways to define such a development process. The original intention was to prescribe it as a paradigm which is much more disciplined than ad hoc development, but also not too rigid to keep it attractive and creative.

The main idea was to follow the principle of correct-by-construction, which when applied in this context meant to define the development process that for a given referent code yields the output PTB-FLA code with the equivalent semantics. Following the approach used by program compilers, the original four-phases development paradigm was devised as the series of program code transformation phases where each phase consumes its input code and produces the semantically equivalent code that is closer to the target PTB-FLA code.

3.2. Four-Phases Paradigm' Phases

This section contains the text adapted from [4]. The four development phases (i.e., phase 1, phase 2, phase 3, and phase 4) are by the convention named by their output code i.e., (1) the referent sequential code, (2) the federated sequential code, (3) the federated sequential code with callbacks, and (4) the PTB-FLA code, respectively.

The input to phase 1 is the raw sequential code and the output is the referent sequential code. The input raw sequential code may come from various sources and may have various forms. Nowadays, many AI/ML algorithm solutions in Python are available online in Colab notebooks, where snippets of textual mathematical explanations, code snippets, and graph plots dynamically created by code play (i.e., execution) are interleaved. Typically, these solutions are primarily intended for learning/understanding the solutions through interactive experimentation, where readers are even encouraged to tweak the code and play with it.

In phase 1, a PTB-FLA developer essentially must select only the necessary code snippets (leaving out the alternative or redundant snippets), to tweak them if needed, and to integrate them into a standalone Python module(s) that they could preferably run on their PC (localhost), typically in a terminal of some IDE. The important requirement for the referent sequential code is that for a given input dataset it must deterministically produce some output data e.g., learned (trained) model coefficients and/or some quality indicators like accuracy, because this output data is used as the referent data by the next development phases, which they must produce to be semantically equivalent. To that end, a PTB-FLA developer should use asserts that automatically compare whether the output data is (approximately) equal to the referent data, and if not, report the corresponding assertion error.

In phase 2, a PTB-FLA developer makes the federated sequential code by the following three steps: (1) partition the input dataset into partitions (that could be distributed across clients), (2) split the monolithic computing of the complete input dataset into a series of computing on individual partitions (that could be collocated with

corresponding partitions) such that they produce the set of partition models, and (3) add the computing for aggregating the set of partition models into the final model and for computing quality indicators (that could be located on a server), as well as for comparing output and referent data. For example, a single function call (calling the function *f*) to process the complete dataset could be split into a series of function calls (calling the same function *f* with different arguments) to process individual dataset partitions. Note that this is still one sequential program that runs on a single machine (developer's PC).

In phase 3, a PTB-FLA developer makes the federated sequential code with callbacks by the following four steps: (1) copy (and tweak) the computing on an individual partition (say a partition number *i*) into the client callback function, (2) replace the series of computing on individual partitions with the series of client callback function calls (with the arguments corresponding to the partition *j* in the call number *j*), (3) copy (and tweak) the computing for aggregating the set of partition models to the final model into the server callback function, and (4) replace the former computing with the server callback function call (the code for computing quality indicators should remain in its place). In the running example, the series of function calls (calling the same function *f* with different arguments) to process individual dataset partitions should be replaced with the corresponding series of client callback function calls, which lead to indirect calls to the function *f* (each call to the client callback function maps to the corresponding call of the function *f*).

In phase 4, a PTB developer makes the PTB-FLA code by the following two steps: (1) add the code for creating the instance *ptb* of the class *PtbFla* and for preparing local and private data for all the instances, and (2) replace the code for calling callback functions (both the series of client callback function calls and the server callback function call) with the call to the function fl_centralized (in case of a centralized FLA) or fl_decentralized (in case of a decentralized FLA) on the object *ptb*.

3.3. Adapted Two-Phases Paradigm for ChatGPT

We adapted the original four-phases paradigm by keeping phase 1 and merging and optimizing the steps of the original phases 2 to 4 into the optimized phase 2 (with the intention to create a minimal context i.e., prompt). In phase 2, ChatGPT translates the referent sequential code into the semantically equivalent PTB-FLA code by following the instructions given in the next paragraph.

Rewrite the referent sequential code into the function main by following these four steps: (1) partition the input dataset into individual client partitions (that could be distributed across clients), (2) add the code to initialize the PtbFla object and prepare local and private data for the current instance, (3) replace the monolithic computing on the complete input dataset with the call to the function fl_(de)centralized on the object ptb, and (4) keep the code that evaluates the final model (i.e., the updated local data) on the test data. Put the computing on an individual partition that produces a partition model into the client callback function. Put the computing for aggregating the set of partition models into the final model into the server callback function.

The instructions above are meant to be general i.e., applicable to any code. However, note that the term "input dataset" mentioned in steps 1 and 3 above is somewhat

ambiguous, and a more precise term is “training dataset”. We discovered this during paradigms’ validation by ChatGPT for the case study on logistic regression (to be presented shortly), wherein we switched to the later term and added the prologue step: (0) split input dataset into the training data and the test data. This note is valid for the four-phases paradigm’ phases (in section 3.2), too.

4. Four-Phases Paradigm Validation

This section presents the validation of the four-phases paradigm adapted for ChatGPT. The next subsections present the validation of phase 2, phase 3, and phase 4, respectively (phase 1 is conducted manually, so it is skipped). For each phase two tables are given – the first table contains the context, whereas the second table comprises the code generated by ChatGPT that was manually tweaked to make it runnable and comparable to the handwritten code (this code passed runtime verification and its output results are the same as for the handwritten code), the ChatGPT explanation of the generated code, and the ChatGPT answers to the questionnaire.

In the tables, comments in italic and enclosed by angle brackets are added as clarifications or replacements and are note part of the original text. Similarly, parts of the contexts in italic were declared as redundant by ChatGPT.

4.1. Phase 2

The context for phase 2, see Table 1, comprises the revised general phase 2 guidelines, the description of the sequential code (the part in italic turned out to be redundant), the assignment for ChatGPT, and the referent sequential code from phase 1.

Table 1. The context for phase 2 of the four-phases paradigm

"""
Revised general phase 2 guidelines:
< Here was a copy of the text from Section 3.2 modified according to the note in Section 3.3. >

Description of the referent sequential code:
The code uses the Social Network Ads (SNA) dataset, which comprises 400 samples (or rows) corresponding to user profiles and each record comprises the following features (or columns): (1) User ID, (2) Gender, (3) Age, (4) Estimated (yearly income), and (5) Purchased. Note that in this example only features Age and Purchased are used.
The main steps in the code are: (1) split SNA dataset into training and test datasets (320 and 80 samples, respectively), (2) train the linear logistic regression model comprising two coefficients, namely b0 and b1, (3) using the trained model, make predictions whether users in the test dataset would make purchase or not, and calculate the prediction accuracy as the ratio of test users for whom the predictions were correct.

Assignment:
Using the phase 2 guidelines, rewrite the referent sequential code, which is given in the function seq_base_case, into the federated sequential code in the function seq_horizontal_federated. Important notes related to the assigment follow.
Note 1: data should be first train_test_split into X_train, X_test, y_train, y_test, and then X_train and y_train should be np.array_split into noClients (=2) partitions!
"""

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from math import exp
# Import supplementary functions
from sup_funs import logistic_regression, evaluate

def seq_base_case():
    # Load the data
    data = pd.read_csv("Social_Network_Ads.csv")

    # Divide the data to training set and test set. For reproducible set random_state=42.
     X_train, X_test, y_train, y_test = train_test_split(data['Age'], \
     data['Purchased'], test_size=0.20, random_state=42)

    # Training the model
    b0, b1 = logistic_regression(X_train, y_train)
    print(f"Training on all the training records (original sequential code): b0={b0}, b1={b1}")

    # Test the model
    y_pred, accuracy = evaluate(X_test, y_test, b0, b1)
    print(f"Accuracy = {accuracy}")

def seq_horizontal_federated():
    """ Please complete this function """
```

The tweaked code and the feedback for phase 2 are given in Table 2. The code generated by ChatGPT did not contain any logical bugs and just a couple of tweaks (marked by the keyword TWEAK) were needed to make it runnable and comparable to the handwritten code. It is worth mentioning that the code generated by ChatGPT is of better quality than the manually written code in [4] (it is generalized for any number of clients and better uses numpy features).

Both the explanation of the generated code and the questionnaire answers are sound and human-like. ChatGPT rated the hardness of the assignment as 6 and found the given context almost minimal (only the part describing the SNA dataset is redundant).

Table 2. The tweaked code and the feedback for phase 2 of the four-phases paradigm

```
< To save space only the function seq_horizontal_federated is shown. >

def seq_horizontal_federated():
    # TWEAK 1: set noClients, e.g. to 2 (to be able to compare with the handwritten code).
    noClients = 2

    # Load the data
    data = pd.read_csv("Social_Network_Ads.csv")

    # Divide the data into training set and test set
     X_train, X_test, y_train, y_test = train_test_split(data['Age'], \
     data['Purchased'], test_size=0.20, random_state=42)

    # Split the training data into partitions
    partitions_X_train = np.array_split(X_train, noClients)
    partitions_y_train = np.array_split(y_train, noClients)

    # Perform federated learning on each partition
```

```
models = []
for i in range(noClients):
   # TWEAK 2: setting the initial b0 and b1, and logistic_regression call with 4 args
   #     were added to align with handwritten code and make it input to the next phase.
   # Train the logistic regression model on each partition
   b0, b1 = 0.0, 0.0  # Initial values of coefficients
   b0, b1 = logistic_regression(partitions_X_train[i], partitions_y_train[i], b0, b1)
   models.append((b0, b1))

# Aggregate the models
aggregated_b0 = np.mean([model[0] for model in models])
aggregated_b1 = np.mean([model[1] for model in models])

# Test the aggregated model on the test data
y_pred, accuracy = evaluate(X_test, y_test, aggregated_b0, aggregated_b1)
print(f"Accuracy = {accuracy}")
```

In this revised code, the main changes include:

1. Splitting the training data X_train and y_train into noClients partitions using np.array_split.
2. Performing federated learning on each partition by training a logistic regression model using the logistic_regression function.
3. Storing the trained models in a list called models.
4. Aggregating the models by calculating the mean of the coefficients (b0 and b1) from all the models.
5. Testing the aggregated model on the test data X_test and y_test using the evaluate function.

Questionnaire answers:

1. On a scale from 1 to 10, I would rate this assignment as a 6. It involves understanding the concept of federated learning and applying it to a logistic regression model. The task requires splitting the data into partitions, performing federated learning on each partition, and aggregating the models. While it may not be the most challenging assignment, it does require a solid understanding of the concepts and some programming skills.
2. The most important part of the context for completing the assignment was understanding the concept of federated learning and its application in a sequential code setting. This included understanding the steps involved in federated learning, such as data partitioning, training models on individual partitions, and aggregating the models. Additionally, the instructions provided the necessary details about the dataset and the specific steps required for the assignment.
3. The redundant part of the context was the initial description of the Social Network Ads (SNA) dataset and its features. While it provides some background information, it is not directly relevant to the task of rewriting the code into a federated sequential code. The important part of the context is already captured in the assignment description, which specifies the steps required for the code transformation.

4.2. Phase 3

The context for phase 3, see Table 3, comprises the revised general phase 3 guidelines, the description of the formats (i.e., signatures) of the callback functions, the important note on the formats of messages, the assignment for ChatGPT, the tweaked code from phase 2, and the placeholders for code completions.

Table 3. The context for phase 3 of the four-phases paradigm

```
"""
General phase 3 guidelines:
< Here was a copy of the text from Section 3.2 modified according to the note in Section 3.3. >

The client and server callback functions have the following format.
sfun(ldata, pdata, msg), returns ldataPrime.
cfun(pdata, msgs), returns ldataSecond.
```

```
Where:
sfun is the server callback function,
ldata the local data (typically a model),
pdata the private data (typically training data),
msg is the message carrying the server local data,
ldataPrime is the client updated local data,
cfun the client callback function,
msgs is the list of all the clients updated local data, and
ldataSecond is the server updated local data.

IMPORTANT NOTE:
In this example, local data for a node is a list with two coefficents [b0, b1],
whereas private data is a list [X_train, y_train].
Therefore, msg is a list (with two element),
whereas msgs is a list of lists (where each of these lists has two elements).

Assignment:
Using the phase 3 guidelines, rewrite the federated sequential code, which is given in the function
seq_horizontal_federated, into the federated sequential code with callbacks in the function
seq_horizontal_federated_with_callbacks and the supplementary client and server callback functions.
Important notes related to the assigment follow.
Note 1: Please do not introduce any new functions, just complete the three incomplete functions below!
Note 2: Assume that the initial local data (ldata) for the server is [0., 0.].
"""
< Here we provided the tweaked code from phase 2 see Table 2. >

def seq_horizontal_federated_with_callbacks():
    """ Please complete this function """

"""
Client callback function arguments format and usage.
Format: localData = [locb0, locb1], privateData = [X_train, y_train], msg = [srvb0, srvb1].
Usage: localData is not used, privateData contains client training data, and msg contains the server model
i.e., the initial coefficients.
"""
def fl_cent_client_processing(localData, privateData, msg):
    """ Please complete this function """

"""
Server callback function arguments amd usage.
Format: msgs = [[b00, b01], [b10, b11], ...]. In bij: i - partion (or client) index, j - coefficent index.
Usage: privateData is not used and msgs contains clients models.
"""
def fl_cent_server_processing(privateData, msgs):
    """ Please complete this function """
```

The tweaked code and the feedback for phase 3 are given in Table 4. The code generated by ChatGPT in the first attempt contained just a single logical bug and after we explained it to ChatGPT (see our dialog with ChatGPT in the explanation filed in Table 4), it generated correct code in the second attempt. It is worth mentioning that the code generated by ChatGPT is of better quality than the manually written code in [4] (the code looks nice and in the callback functions really condensed).

Both the explanation of the generated code and the questionnaire answers are sound (except, perhaps, the third answer) and human-like. ChatGPT rated the hardness of the assignment as 6. The third answer is somewhat strange because SNA dataset is not mentioned in the context for phase 3 (we intentionally skipped it, because we wanted to create a minimal context), so the third answer came unexpectedly to us. After giving it a

second thought, it seems that either ChatGPT remembered the context for the previous phase (because it was conducted in the same session) or its comment is on possible generalization – consider these final words in the third answer: "as the transformation process would apply to other models and datasets as well", see them in Table 4. However, both alternatives do not compromise the context minimality.

Table 4. The tweaked code and the feedback for phase 3 of the four-phases paradigm

```
< To save space only the function seq_horizontal_federated_with_callbacks and the callback functions are shown. >

def seq_horizontal_federated_with_callbacks():
    # Load the data
    data = pd.read_csv("Social_Network_Ads.csv")

    # Divide the data into training set and test set
     X_train, X_test, y_train, y_test = train_test_split(data['Age'], \
     data['Purchased'], test_size=0.20, random_state=42)

    # Split the training data into partitions
    noClients = 2  # Number of clients
    partitions_X_train = np.array_split(X_train, noClients)
    partitions_y_train = np.array_split(y_train, noClients)

    # Initialize the server model coefficients
    server_model = [0.0, 0.0]

    # Perform federated learning on each partition using callbacks
    client_models = []  # List to store client models
    for i in range(noClients):
        # Train the logistic regression model on each partition with the client callback
        client_model = fl_cent_client_processing([], [partitions_X_train[i], \
        partitions_y_train[i]], server_model)
        client_models.append(client_model)

    # Update the server model using the client models with the server callback
    server_model = fl_cent_server_processing([], client_models)

    # Get the final aggregated model coefficients
    aggregated_b0, aggregated_b1 = server_model
     print(f"The final model coefficients: aggregated_b0={aggregated_b0}, \
     aggregated_b1={aggregated_b1}")

    # Test the aggregated model on the test data
    y_pred, accuracy = evaluate(X_test, y_test, aggregated_b0, aggregated_b1)
    print(f"Accuracy = {accuracy}")

"""
Client callback function arguments format and usage.
Format: localData = [locb0, locb1], privateData = [X_train, y_train], msg = [srvb0, srvb1].
Usage: localData is not used, privateData contains client training data, and msg contains the server model, i.e., the initial coefficients.
"""
def fl_cent_client_processing(localData, privateData, msg):
    b0, b1 = logistic_regression(privateData[0], privateData[1], msg[0], msg[1])
    return [b0, b1]
```

```
"""
Server callback function arguments and usage.
Format: msgs = [[b00, b01], [b10, b11], ...]. In bij: i - partition (or client) index, j - coefficient index.
Usage: privateData is not used and msgs contains clients' models.
"""
def fl_cent_server_processing(privateData, msgs):
    aggregated_b0 = np.mean([model[0] for model in msgs])
    aggregated_b1 = np.mean([model[1] for model in msgs])
    return [aggregated_b0, aggregated_b1]
```

< *ChatGPT:* > The seq_horizontal_federated_with_callbacks function performs federated learning using callbacks.

< *Authors to ChatGPT:* > You made just one mistake in the function seq_horizontal_federated_with_callbacks, the server should calculate the aggregated model after all the clients updated their models, so the sever callback function should not be called within the for loop but after that loop. Please could you correct that mistake and give the corrected function?

< *ChatGPT:* > Apologies for the mistake. Here's the corrected version of the function. In this updated version, the server callback function fl_cent_server_processing is called after all the client models have been collected in the client_models list.

Questionnaire answers:

1. On a scale from 1 to 10, I would rate this assignment as a 6. It involves understanding the concept of federated sequential code with callbacks and implementing the necessary changes in the provided code. While it may require some familiarity with the concepts of federated learning and callback functions, the guidelines provided in the assignment help in completing the task.
2. The most important part of the context for completing the assignment is the understanding of federated sequential code and the purpose of using callbacks in the context of federated learning. The guidelines provided in the assignment describe the steps involved in transforming the original code into code with callbacks, which helps in understanding the expected changes.
3. The redundant part of the context is the mention of the specific logistic regression model and the dataset ("Social_Network_Ads.csv"). While they provide a concrete example for the assignment, the main focus is on transforming the code into a federated sequential code with callbacks. The specific model and dataset used can be considered irrelevant to the main objective of the assignment, as the transformation process would apply to other models and datasets as well.

4.3. Phase 4

The context for phase 4, see Table 5, comprises the revised general phase 4 guidelines, the description of the formats of the callback functions, the important note on the formats of messages, the assignment for ChatGPT with additional notes (added in the second version of the context, which was also the final version), the tweaked code from phase 3, and the placeholders for the code completion.

Table 5. The context for phase 4 of the four-phases paradigm

```
"""
General phase 4 guidelines:
< Here was a copy of the text from Section 3.2 modified according to the note in Section 3.3. >

The PTB-FLA API includes the following functions that are defind in the class PtbFla.
PtbFla(noNodes, nodeId, flSrvId=0), no return value.
fl_centralized(sfun, cfun, ldata, pdata, noIters=1), returns ldataUpdated.
Where:
sfun is the server callback function,
cfun the client callback function,
ldata the local data (typically a model),
pdata the private data (typically training data),
```

noIters the number of iterations (default value is 1), and
ldataUpdated is the updated ldata.

The client and server callback functions have the following format.
< *The description is the same as in Table 3.* >

IMPORTANT NOTE:
< *The note is the same as in Table 3.* >

Assignment:
Using the phase 4 guidelines, rewrite the federated sequential code with callbacks, which is given in the function seq_horizontal_federated_with_callbacks, into the PTB-FLA code in the function main, which is launched as noNodes processes. Important notes related to the assignment follow.
Note 1: Please do not introduce any new functions, just complete the function main!
Note 2: PTB-FLA code should use the same client and server callback functions as the function seq_horizontal_federated_with_callbacks!
Note 3: Since there is one server and noClinets clients, noNodes = noClients + 1 i.e., noClients = noNodes - 1.
"""
< *Here we provided the tweaked code from phase 3 see Table 4.* >

"""
The function main is executed by noNodes processes (or nodes) and each process gets its own command line arguments.
Assume that for all nodes noNodes is 3 and flSrvId is 2,
and individual nodes are assigned nodeId values 0, 1, and 2, respectively.
"""
```
def main():
    # Process command line arguments
    noNodes = int( sys.argv[1] )
    nodeId = int( sys.argv[2] )
    flSrvId = int( sys.argv[3] )
    """ Please complete this function """
```

The tweaked code and the feedback for phase 4 are given in Table 6. This session was a bit disappointing and time consuming and was exactly the reason why we changed the process to manually tweak the code in case when it contains a small number of obvious bugs rather than continuing a seemingly endless and diverging sessions.

In this case, it took us six attempts to guide ChatGPT with additional explanations and notes from our side so that it produces the correct code. However, even the code we got in the sixth attempt that at glance looked like right, had a single bug, which we discovered when executing the code, and which we decided to manually tweak to get it right (see TWEAK 2 in the code in Table 6). Overall, we may state that this effort paid off as the final tweaked code is of better quality than the manually written code in [4] (because it is generalized for any number of clients and has better usage of APIs).

Both the explanation of the generated code and the questionnaire answers are sound and human-like. ChatGPT rated the hardness of the assignment as 7. The third answer needs to be explained. Although the phase 3 code already comprised the client and server callback functions, ChatGPT introduced new callback functions (server_callback and client_callback) and used them instead of already provided functions (and this was not the only problem in the generated code). Perhaps it introduced these new callback functions because it remembered that it was asked is previous assignments to create callback functions. No matter what the reason was, these new functions that ChatGPT

introduced were redundant, and in its third answer it admitted it, however this does not compromise the context minimality.

Table 6. The tweaked code and the feedback for phase 4 of the four-phases paradigm

< To save space only the function main is shown, and only the most important TWEAK. >

```
"""
The function main is executed by noNodes processes (or nodes) and each process gets its own command line
arguments.
Assume that for all nodes noNodes is 3 and flSrvId is 2,
and individual nodes are assigned nodeId values 0, 1, and 2, respectively.
"""
def main():
    # Process command line arguments
    noNodes = int(sys.argv[1])
    nodeId = int(sys.argv[2])
    flSrvId = int(sys.argv[3])

    # Load the data
    data = pd.read_csv("Social_Network_Ads.csv")

    # Divide the data into training set and test set
     X_train, X_test, y_train, y_test = train_test_split(data['Age'], \
     data['Purchased'], test_size=0.20, random_state=42)

    # Split the training data into partitions
    noClients = noNodes - 1  # Number of clients
    partitions_X_train = np.array_split(X_train, noClients)
    partitions_y_train = np.array_split(y_train, noClients)

    # Create an instance of PtbFla
    ptb = PtbFla(noNodes, nodeId, flSrvId)

    # Prepare local data and private data for the current node
    localData = [0.0, 0.0]  # Initial local data (model coefficients)

     # TWEAK 2: the server doesn't have its private data, so the next 2 lines were -
     # added and the 3rd is indented
    privateData = None
    if nodeId != flSrvId:
        privateData = [partitions_X_train[nodeId], partitions_y_train[nodeId]]  # Private data

    # Perform federated learning using PTB-FLA
     updated_ldata = ptb.fl_centralized(fl_cent_server_processing, \
     fl_cent_client_processing, localData, privateData)

    # Get the final aggregated model coefficients
    aggregated_b0, aggregated_b1 = updated_ldata
     print(f"The final model coefficients: \
     aggregated_b0={aggregated_b0}, aggregated_b1={aggregated_b1}")

    # Test the aggregated model on the test data
    y_pred, accuracy = evaluate(X_test, y_test, aggregated_b0, aggregated_b1)
    print(f"Accuracy = {accuracy}")

    # TWEAK 3: this code was added to make it comparable with handwritten code
    # Must be
```

```
    if nodeId == flSrvId:
        refbs = seq_horizontal_federated_with_callbacks()
        assert refbs[2][0] == aggregated_b0 and refbs[2][1] == aggregated_b1, \
        "b0 and b1 must be equal to ref_b0 and ref_b1, respectively!"

    # Shutdown
    del ptb
    pkey = input('press any key to continue...')
```

< *Below is the ChatGPT explanation after the first attempt.* >
I have updated the main() function to include the PTB-FLA code. The client and server callback functions remain the same. The PtbFla instance ptb is created, and the local and private data are prepared. The server and client callback functions are defined as inner functions within main(). Finally, the callback function calls are replaced with the fl_centralized function call on the ptb object.
< *To save space our dialog with ChatGPT in the next five attempts is not shown here.* >

Questionnaire answers:

1. I would rate this assignment a 7 on the difficulty scale. The assignment involved understanding the concept of federated learning, callback functions, and their integration into the PTB-FLA code. It required careful reading and interpretation of the provided context to ensure the correct implementation.
2. The most important part of the context was the description of the PTB-FLA API functions, namely PtbFla and fl_centralized. Understanding their input arguments, return values, and how they should be used was crucial in completing the assignment.
3. The redundant part of the context was the introduction of new callback functions (server_callback and client_callback). These functions were not required and caused confusion. It was sufficient to use the provided callback functions fl_cent_server_processing and fl_cent_client_processing directly in the fl_centralized function call.

5. Two-Phases Paradigm Validation

This section presents the validation of the two-phases paradigm adapted for ChatGPT. As phase 1 is conducted manually, we immediately proceed to phase 2.

The context for phase 2, see Table 7, comprises the revised general guidelines for the two-phases paradigm, the description of the formats of the callback functions, the important note on the formats of messages, the assignment for ChatGPT with additional notes (added in the second version of the context, which was also the final version), the referent code from phase 1, and the placeholders for the code completion.

Table 7. The context for phase 2 of the two-phases paradigm

"""
Revised general guidelines for the two-phases paradigm:
< *Here was a copy of the text from Section 3.3.* >

The PTB-FLA API includes the following functions that are defined in the class PtbFla.
< *The description is the same as in Table 5.* >

The client and server callback functions have the following format.
< *The description is the same as in Table 3.* >
IMPORTANT NOTE:
< *The note is the same as in Table 3.* >

Assignment:
Rewrite the referent sequential code, which is given in the function seq_base_case, into the PTB-FLA code in the function main and supplementary client and server callback functions in the functions fl_cent_client_processing and fl_cent_server_processing, respectively. The PTB-FLA code is launched as

```
noNodes processes. Important notes related to the assignment follow.
Note 1: There is one server and noClinets clients, noNodes = noClients + 1 i.e., noClients = noNodes - 1.
Note 2: The server does not have its own private data.
"""
< Here we provided the referent sequential code from phase 1, see Table 1. >

< The prolog comment for the function below is the same as in Table 3. >
def fl_cent_client_processing(localData, privateData, msg):
    """ Please complete this function """

< The prolog comment for the function below is the same as in Table 3. >
def fl_cent_server_processing(privateData, msgs):
    """ Please complete this function """

< The prolog comment for the function below is the same as in Table 5. >
def main():
    # Process command line arguments
    noNodes = int( sys.argv[1] )
    nodeId = int( sys.argv[2] )
    flSrvId = int( sys.argv[3] )
    """ Please complete this function """
```

The tweaked code and the feedback for phase 2 are given in Table 8. This session started great as ChatGPT immediately produced perfect callback functions and the function main with just a single bug – it "forgot" that FL server doesn't have its own private data. This code was easily tweakable, and we consider this as a great achievement – it justifies both the paradigm/context effectiveness and comes close to an ideal solution where a developer gives the context and the input referent sequential code and gets back (almost) the perfect target PTB-FLA code.

Now because the code was easily tweakable, we wanted to see whether ChatGPT could tweak it on its own, so we added two additional notes related to the assignment (in the second version of the context in Table 7). Unfortunately, ChatGPT unexpectedly produced the code that was more far away from the target code. So, we entered a rather tedious session and finally in the sixth attempt ChatGPT produced the correct function main. Overall, we may again state that this effort paid off as the final tweaked code is of better quality than the manually written code in [4].

The dialog with ChatGPT was cumbersome but the questionnaire answers are sound (except the third one) and human-like. ChatGPT rated the hardness of the assignment as 7. The third answer came unexpectedly, because in this session ChatGPT did not introduce new callback functions at all, obviously it mixed up this session with the previous one however this does not compromise the context minimality.

Table 8. The tweaked code and the feedback for phase 2 of the two-phases paradigm

```
< To save space only the function main and the callback functions are shown. >

def fl_cent_client_processing(localData, privateData, msg):
    b0, b1 = logistic_regression(privateData[0], privateData[1], msg[0], msg[1])
    return [b0, b1]

def fl_cent_server_processing(privateData, msgs):
    aggregated_b0 = np.mean([model[0] for model in msgs])
    aggregated_b1 = np.mean([model[1] for model in msgs])
    return [aggregated_b0, aggregated_b1]
```

```
def main():
    # Process command line arguments
    noNodes = int(sys.argv[1])
    nodeId = int(sys.argv[2])
    flSrvId = int(sys.argv[3])
    print(noNodes, nodeId, flSrvId)

    # Load the data
    data = pd.read_csv("Social_Network_Ads.csv")

    # Divide the data to training set and test set. For reproducible set random_state=42.
     X_train, X_test, y_train, y_test = train_test_split(data['Age'], \
     data['Purchased'], test_size=0.20, random_state=42)

    # Split the training data into partitions
    noClients = noNodes - 1  # Number of clients
    partitions_X_train = np.array_split(X_train, noClients)
    partitions_y_train = np.array_split(y_train, noClients)

    # Initialize the PTB-FLA object
    ptb = PtbFla(noNodes, nodeId, flSrvId)

    # Prepare local data and private data for the current instance
    localData = [0.0, 0.0]  # Initial values of coefficients
     privateData = None if nodeId == flSrvId \
     else [partitions_X_train[nodeId], partitions_y_train[nodeId]]

    # Perform federated learning using PTB-FLA
     updated_ldata = ptb.fl_centralized(fl_cent_server_processing, \
     fl_cent_client_processing, localData, privateData)

    # Get the updated model coefficients
    aggregated_b0, aggregated_b1 = updated_ldata

    # Print the final model coefficients
     print(f"The final model coefficients: \
     aggregated_b0={aggregated_b0}, aggregated_b1={aggregated_b1}")

    # Test the aggregated model on the test data
    y_pred, accuracy = evaluate(X_test, y_test, aggregated_b0, aggregated_b1)
    print(f"Accuracy = {accuracy}")

    # Must be (TWEAK 2)
    if nodeId == flSrvId:
        refbs = seq_horizontal_federated_with_callbacks()
        assert refbs[2][0] == aggregated_b0 and refbs[2][1] == aggregated_b1, \
        "b0 and b1 must be equal to ref_b0 and ref_b1, respectively!"

    # Shutdown
    del ptb
    pkey = input('press any key to continue...')
```

< ChatGPT: > Please complete the fl_cent_client_processing and fl_cent_server_processing functions as per the assignment instructions.
< Authors to ChatGPT: > Please could you also complete the callback functions.
< ChatGPT (after flashing out the functions): > These callback functions implement the logistic regression training on each client partition and the aggregation of the models on the server side, respectively.
< To save space our dialog with ChatGPT in the next five attempts is not shown here. >

Questionnaire answers:

1. I would rate this assignment a 7 on the difficulty scale. It involves understanding and applying the concepts of federated learning, PTB-FLA, and callback functions to transform a sequential code into a distributed PTB-FLA code. Additionally, there were some details and modifications required to address specific issues and bugs.
2. The most important part of the context was understanding the PTB-FLA framework, including the functions PtbFla, fl_centralized, and the roles of the server and client callback functions. Additionally, understanding the structure and format of the local data and private data used in the example was crucial for completing the assignment.
3. The redundant part of the context was the introduction of new callback functions (server_callback and client_callback) in the earlier versions of the code. These functions were not required, and the provided callback functions (fl_cent_server_processing and fl_cent_client_processing) should have been used instead.

6. Adapted Development Paradigms Performance

In this section, we report on the adapted development paradigms performance in terms of human labour (in working hours) and size of ChatGPT context (in number of characters with spaces) needed to develop the logistic regression PTB-FLA code. We would like to emphasize that data on human labour should be treated as rough estimates because it is based on our freeform notes in private diaries. Data on ChatGPT context size are exact and we got them by the text editor.

Table 9 is organized as follows, the left part relates to the human labor and the right part relates to the ChatGP contexts size, the top part contains raw (input) data, and the bottom part contains calculated (output) data; some of the fields and not applicable (but this should be obvious, so we skip explaining these exceptions). There are two types of calculated data: (i) the working speed up that is defined as the ratio of working hours, and (ii) the ChatGTP context size reduction that is defined as the ratio of ChatGPT context sizes.

Table 9. Adapted paradigms performance data

	Human labour [h]			Context size [ch with spaces]	
Phase	4-Ph human	4-Ph GPT	2-Ph GPT	4-Ph GPT	2-Ph GPT
Phase 2	8	4	4	2462	2685
Phase 3	12	4		2593	
Phase 4	4	4		2334	
Total	24	12	4	7389	2685
	Speed up:	2	6	Reduction:	2.75

The calculated data in the bottom raw reveal encouraging results, the adapted 4-phases development paradigm for ChatGPT achieved the speed up of 2 times over the original 4-phases development paradigm for humans, whereas the adapted 2-phase development paradigm for ChatGPT achieved the speed up of 6 times over the original 4-phase development paradigm for humans. However, one should keep in mind that these impressive results are based on rough estimates of working hours.

Regarding the ChatGPT context sizes, the calculated data in the bottom row of Table 9 reveal that the adapted 2-phases development paradigm for ChatGPT achieves the context size reduction of 2.75 times over the adapted 4-phases development paradigm for ChatGPT i.e., it makes using ChatGPT 2.75 times cheaper.

7. Related Work

To make it easier to follow, we start this section with a taxonomy of the features and a classification of related works by the features, see Table 10. The taxonomy comprises the following features: (1) low code and no code paradigms, (2) code generation, (3) paradigm adaptations for ChatGPT, and (4) other (references not directly comparable with this paper). For references comparable with this paper (not in "Other"), we discuss their weakness and how this paper solves this weakness.

Table 10. Related works versus the taxonomy features

Paper	Low code No code	Code Generation	Paradigm Adaptation	Other (Comment)
This	Yes	Yes	Yes	
1				Yes (Umbrella EU project)
2				Yes (Target FL framework)
3				Yes (Logistic regression use case inputs)
4	Yes			
5	Yes	Yes		
6		Yes		
7				Yes (Text-bot usage modes by P.J. Denning)
8		Yes		
9		Yes		
10				Yes (Status of the ChatGPT popularity)
11				Yes (Overview of low code program. models)
12	Yes	Yes		
13	Yes			
14				Yes (Discussion on ChatGPT privacy issues)
15				Yes (Well-known ChatGPT hallucinations)
16				Yes (Strategies for ChatGPT hallucinations)
17				Yes (Strategies for ChatGPT hallucinations)
18				Yes (Strategies for ChatGPT hallucinations)
19				Yes (Assessment of program. assignments)
20				Yes (Abstract state machines, Fed. Networks)
21				Yes (Discussion thread summarising systems)
22				Yes (Decision-making for mental health)

Not long ago, various systems and frameworks have appeared that can automate code generation e.g., Amazon's CodeWhisperer, GitHub's Copilot, IBM's Project Wisdom, etc. Although it is unlikely AI will eliminate jobs for developers anytime soon, it is poised to revolutionize the way software is created. For instance, the OpenAI Codex generates software in many popular programming languages, including Python, Perl, Ruby, and PHP [6]. Discussion: from the three relevant features, [6] only features code generation, whereas this paper additionally features low code and paradigm adaptation.

P.J. Denning has seen three modes of interaction with text-bots: jump-start, provocateur, and appropriator. In this paper we use the *jump-start mode*, and this term means the bot helps the human complete a job faster e.g., a programmer asks a text-bot to generate initial code, then reviews and edits to make it error free [7]. This reference is classified as "Other" in Table 10.

Sarkar et al. explored how programming with large language models is similar and different from the prior conceptualisations of programmer assistance and found that while LLM-assisted programming shares some properties of compilation, pair programming, and programming via search and reuse, there are some fundamental differences [8]. Discussion: from the three relevant features, [8] only features code generation, whereas this paper additionally features low code and paradigm adaptation.

Bird et al. conducted three studies to understand how developers use GitHub Copilot: (i) an analysis of forum discussions from early Copilot users, (ii) a case study of Python developers using Copilot for the first time, and (iii) a large-scale survey of Copilot users to understand its impact on productivity [9]. Discussion: from the three relevant features, [9] only features code generation, whereas this paper additionally features low code and paradigm adaptation.

However, it seems that ChatGPT is still generally the most popular LLM with its 180 million users as of December 2023 [10] and therefore we decided to use it for validation of our adapted development paradigms in this paper. This reference is classified as "Other" in Table 10.

Martin Hirzel in his recent paper on low-code programming models [11] discussed the following three representative techniques for low-code programming: visual programming languages (VPLs), programming by demonstration (PBD), and programming by natural language (PBNL), and all three languages use some domain-specific language (DSL) that is typically embedded in a general-purpose programming language (like Python). In the context of [11], the approach used in this paper is close to low-code approach based on PBNL. This reference is classified as "Other" in Table 10.

Busch et al. in [12] extended their language driven engineering environment (LDE) for low-code/no-code development via dedicated DSLs to also support specification in natural language with the following benefits: (i) the tasks to be solved via LLM-based code generation can be tailored in size and conceptual complexity and (ii) the overall heterogeneously constructed system can be directly validated at system level using automata learning and model analysis. The main difference between this paper and [12] is that in this paper the final PTB-FLA code is correct by construction and is semantically equivalent to the referent sequential code (both produce the same results). Discussion: from the three relevant features, [12] features low-code/no-code and code generation, whereas this paper additionally features paradigm adaptation.

An interesting approach to low-code implementation and no-code reuse is given in [13], where authors use Pyrus, a web-based, low-code/no-code graphical modelling environment for ML and data analytics applications, to solve the challenge of re-using AI/ML models that do not address data with imbalanced class by implementing ML algorithms in Python that do the re-balancing and then using these algorithms and the original models in the data processing pipelines in Pyrus. At present, this paper uses a low-code approach, and extending it by no-code reuse is an interesting direction for the future work. Discussion: of the three relevant features, [13] features only low-code/no-code, whereas this paper additionally features code generation and paradigm adaptation.

Regarding the discussion on data privacy issues with ChatGPT (issues with open code, who is owning the data when query is sent and what can OpenAI do with the data, etc.) [14], we would like to clarify that in our development process (see Section 4) ChatGPT never receives the private data (i.e., the data that is used to train the model), it only receives the program code. Therefore, in this respect, our development process is

completely aligned with the idea of federated learning. Other issues covered in [14] are out of the scope of this paper. This reference is classified as "Other" in Table 10.

In this paper, we accepted the well-known ChatGPT hallucinations [15] as a kind of tool imperfectness, because they were rarely happening in our experiments, and therefore we did not try to mitigate or counter them. Meanwhile, some strategies were proposed to tackle this issue [16-18], and we plan to conduct research in this direction in our future work. These references are classified as "Other" in Table 10.

At the end of this section, we would like to shortly mention the four interesting papers that are orthogonal to this paper and relevant to our longer-term future work.

First, Paiva et al. in [19] provided a thorough comparison of the most widespread semantic graph representations for the automated assessment of programming assignments. This and similar techniques and tools may be used to assess the program codes received from the ChatGPT. This reference is classified as "Other" in Table 10.

Second, Ghilezan et al. in [20] designed the architecture of the overlay network federation using abstract state machines, and they showed that the federation can be used as a formal model of forward contact tracing in the context of Covid-19 pandemic. Abstract state machines or similar formal methods perhaps may be used to formally verify the program codes received from ChatGPT. This reference is classified as "Other" in Table 10.

Third, Khan et al. in [21] we introduce the long short-term memory (LSTM) auto-encoder as a sentence embedding technique to improve the performance of the discussion thread summarizing system (DTS), which creates a candid view of the entire discussion of a query. DTS and similar techniques and tools may be used to independently analyze series of longer sessions with ChatGPT when developing more complex PTB-FLA algorithms. This reference is classified as "Other" in Table 10.

Fourth, Yu and Lin in [22] provided a scientific decision-making basis for mental health education in colleges and universities by extracting the classification rules from the optimal decision tree model. Similar techniques and tools may be used to analyze ChatGPT responses. This reference is classified as "Other" in Table 10.

8. Conclusion and Future Work

In this paper, we present the four-phase and the two-phase PTB-FLA development paradigms adapted to be used by ChatGPT. The former paradigm comprises the four phases: (1) the sequential code, (2) the federated sequential code, (3) the federated sequential code with callbacks, and (4) the PTB-FLA code, whereas the latter comprises the two phases: (1) the sequential code and (2) the PTB-FLA code. The PTB-FLA development paradigms were adapted and validated using the logistic regression case study, where centralized FL algorithms were developed.

The main original contributions of this paper are: (1) the PTB-FLA development paradigms adapted for ChatGPT, (2) our contexts (i.e., prompts) for the individual phases, the tweaked program codes, and the questionnaire, (3) ChatGPT generated program codes, the explanations, and the questionnaire answers, and (4) the report on adapted development paradigms performance in terms of human labour and ChatGPT context size.

The main advantages of all the PTB-FLA development paradigms over ad hoc approach are: (1) developed FLAs are correct by construction (they produce the same results as their referent sequential counterparts) and (2) the development of FLAs is easier than by ad hoc approach.

The main advantage of the two PTB-FLA development paradigms adapted for ChatGPT over the original development paradigm for humans is that they are faster and cheaper, because they require less human labour, and it is reasonable to assume that ChatGPT costs will be negligible compared to human labor (at this time it is for free).

When comparing the two PTB-FLA development paradigms adapted for ChatGPT, the two-phase paradigm: (1) aids easier development that is close to an ideal solution where a developer gives the context and the input referent sequential code and gets back (almost) the perfect target PTB-FLA code, and (2) it is cheaper to use as it requires smaller ChatGPT context.

The main shortcoming of all the PTB-FLA development paradigms (both for humans and ChatGPT) is that they still have not been tested during the development of some real applications.

The main limitation of this paper is that only ChatGPT was used in the experimental validation.

The main directions of the future work are: (1) use the PTB-FLA development paradigms to develop other more complex FLAs and real applications, (2) continue improving the PTB-FLA development paradigms based on the feedback from developing more complex FLAs and real applications, (3) research adapting and specifying the PTB-FLA development paradigms for other emerging AI tools e.g., other general purpose LLMs like Bard, as well as other models that are specialized for coding, such as Code LLama, StarCoder, WizardCoder, and Safurai 001, etc., and (4) research techniques to mitigate ChatGPT hallucinations.

Acknowledgement. Funded by the European Union (TaRDIS, 101093006). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.
Authors are grateful to anonymous reviewers for their valuable comments and suggestions.

References

1. TaRDIS: Trustworthy and Resilient Decentralised Intelligence For Edge Systems. [Online]. Available: https://www.project-tardis.eu/ (current March 2024)
2. Popovic, M., Popovic, M., Kastelan, I., Djukic, M., Ghilezan, S.: A Simple Python Testbed for Federated Learning Algorithms. In Proceedings of the 2023 Zooming Innovation in Consumer Technologies Conference. IEEE Xplore, New Jersey, USA, 148-153. (2023)
3. Logistic Regression. [Online]. Available: https://colab.research.google.com/drive/1qmdfU8tzZ08D3O84qaD11Ffl9YuNUvlD (current December 2023)
4. Popovic, M., Popovic, M., Kastelan, I., Djukic, M., Basicevic, I.: A Federated Learning Algorithms Development Paradigm. In: Kofron, J., Margaria, T., Seceleanu, C. (eds.): Engineering of Computer-Based Systems. Lecture Notes in Computer Science, Vol. 14390. Springer, Cham, 26–41. (2024)

5. Popovic, M., Popovic, M., Kastelan, I., Djukic, M., Basicevic, I.: Developing Elementary Federated Learning Algorithms Leveraging the ChatGPT (2023). In Proceedings of the 31st Telecommunications Forum (TELFOR). IEEE Xplore, New Jersey, USA, 1-4. (2023)
6. Greengard, S.: AI Rewrites Coding. Communications of the ACM, Vol. 66, No. 4, 12–14. (2023)
7. Denning, P.J.: The Profession of IT: Can Generative AI Bots Be Trusted? Communications of the ACM, Vol. 66, No. 6, 24–27. (2023)
8. Sarkar, A., Gordon, A.D., Negreanu, C., Poelitz, C., Ragavan, S.S., Zorn, B.: What is it like to program with artificial intelligence? (2023). [Online]. Available: https://arxiv.org/abs/2208.06213 (current March 2024)
9. Bird, C., Ford, D., Zimmermann, T., Forsgren, N., Kalliamvakou, E., Lowdermilk, T., Gazit, I.: Taking Flight with CopilotTaking Flight with Copilot. Communications of the ACM, Vol. 66, No. 6, 56–62. (2023)
10. Duarte, F.: Number of ChatGPT Users (2023). [Online]. Available: https://explodingtopics.com/blog/chatgpt-users (current March 2024)
11. Hirzel, M.: Low-Code Programming Models. Communications of the ACM, Vol. 66, No. 10, 76–85. (2023)
12. Busch, D., Bainczyk, A., Steffen. B.: Towards LLM-Based System Migration in Language-Driven Engineering. In: Kofron, J., Margaria, T., Seceleanu, C. (eds.): Engineering of Computer-Based Systems. Lecture Notes in Computer Science, Vol. 14390. Springer, Cham, 191–200. (2024)
13. Singh, A., Minguett, O.: IDPP: Imbalanced Datasets Pipelines in Pyrus. In: Kofron, J., Margaria, T., Seceleanu, C. (eds.): Engineering of Computer-Based Systems. Lecture Notes in Computer Science, Vol. 14390. Springer, Cham, 60–69. (2024)
14. Liesenfeld, A., Lopez, A., Dingemanse, M.: Opening up ChatGPT: Tracking openness, transparency, and accountability in instruction-tuned text generators. In Proceedings of the 5th international conference on conversational user interfaces. ACM, New York, NY, Article No. 47, 1-6 (2023)
15. Rawte, V., Sheth, A., Das, A.: A Survey of Hallucination in Large Foundation Models. (2023). [Online]. Available: https://arxiv.org/abs/2309.05922 (current March 2024)
16. Tonmoy, S. M., Zaman, S. M., Jain, V., Rani, A., Rawte, V., Chadha, A., Das, A.: A Comprehensive Survey of Hallucination Mitigation Techniques in Large Language Models. (2024). [Online]. Available: https://arxiv.org/abs/2401.01313 (current March 2024)
17. Rawte, V., Priya, P., Tonmoy, S. M., Zaman, S. M., Sheth, A., Das, A.: Exploring the Relationship between LLM Hallucinations and Prompt Linguistic Nuances: Readability, Formality, and Concreteness. (2023). [Online]. Available: https://arxiv.org/abs/2309.11064 (current March 2024)
18. Martino, A., Iannelli, M., Truong, C.: Knowledge Injection to Counter Large Language Model (LLM) Hallucination. In: Pesquita, C., et al. (eds.): The Semantic Web: ESWC 2023 Satellite Events. Lecture Notes in Computer Science, Vol. 13998. Springer, Cham, 182-185. (2023).
19. Ghilezan, S., Kašterović, S., Liquori, L., Marinković, B., Ognjanović, Z., Stefanović, T.: Federating Digital Contact Tracing using Structured Overlay Networks. Computer Science and Information Systems, Vol. 19, No. 3, 1261-1282. (2022)
20. Paiva, J. C., Leal, J. P., Figueira, Á.: Comparing Semantic Graph Representations of Source Code: The Case of Automatic Feedback on Programming Assignments. Computer Science and Information Systems, Vol. 21, No. 1, 117–142. (2024)
21. Khan, A. W., Al-Obeidat, F., Khalid, A., Amin, A., Moreira, F.: Sentence embedding approach using LSTM auto-encoder for discussion threads summarization. Computer Science and Information Systems, Vol. 20, No. 4. (2023)
22. Yu, J., Lin, J.: Data mining technology in the analysis of college students' psychological problems. Computer Science and Information Systems, Vol. 19, No. 3, 1583-1596. (2022)

Miroslav Popovic received his Dipl. Eng., M.Sc., and Ph.D. degrees from the Faculty of Technical Sciences, University of Novi Sad, Serbia, in 1984, 1988 and 1990, respectively. He has been Full Professor at the University of Novi Sad since 2002. Currently he is giving courses on parallel and systems programming and on computer networks. His research interests are engineering of computer-based systems and intelligent distributed systems. He has authored or co-authored more than 165 scientific papers, and the book Communication protocol engineering, 2nd Edition (CRC Press).

Marko Popovic received his B.Sc., M.Sc., and PhD degrees from the Faculty of Technical Sciences, University of Novi Sad, Serbia, in 2015, 2017, and 2020, respectively. Currently he is Scientific Researcher affiliated with the RT-RK Institute of Computer Based Systems, Novi Sad, Serbia. His research interests are in the areas of engineering of computer-based systems and intelligent distributed systems. He has authored or co-authored more than 23 scientific papers.

Ivan Kastelan received Dipl. Eng. M.Sc, and PhD degrees from the Faculty of Technical Sciences, University of Novi Sad, Serbia, in 2008, 2009, and 2014, respectively. Currently he is Associate Professor at the University of Novi Sad, teaching courses on algorithms and computer engineering. His research interests include digital and computer systems design, design of algorithms, and engineering education. He is the author of one textbook, more than 90 scientific papers, and 13 patents.

Miodrag Djukic received Dipl. Eng. and M.Sc degree in 2008., from the Faculty of Technical Sciences, University of Novi Sad, Serbia, and PhD degree in 2015 from the same faculty. Currently he is an Assistant Professor at the University of Novi Sad, teaching courses on compilers and programming. His research interests include compilers, software tools, formal methods, and engineering education. He has authored or co-authored more than 50 scientific papers.

Ilija Basicevic received Dipl. Eng. M.Sc, and PhD degrees from the Faculty of Technical Sciences, University of Novi Sad, Serbia, in 1998, 2001, and 2009, respectively. Currently he is Full Professor at the University of Novi Sad, teaching courses on computer networks. His research interests are in the areas of Internet protocols and network security. He has authored or co-authored more than 100 scientific papers and one textbook.

Received: December 24, 2023; Accepted: May 28, 2024.

Computer Science and Information Systems 21(4):1293–1320 https://doi.org/10.2298/CSIS240327039D

TRL-PROTAC: A pre-trained generator of PROTACs targeting specific proteins optimized by reinforcement learning

Yuhao Dai and Fei Zhu*

School of Computer Science and Technology, Soochow University
215006 Suzhou, Jiangsu, China
20224227033@stu.suda.edu.cn
zhufei@suda.edu.cn

Abstract. Proteolysis-targeting chimeras (PROTACs) introduce a novel paradigm in drug development, incorporating three essential components: the warhead, the E3 ligand, and the linker. The complexity of the ternary structure, particularly the intricate design of the linker, presents a significant challenge in PROTACs drug design. Here an integrated protocol for design and evaluation of PROTACs targeting specific proteins, called TRL-PROTAC is proposed. TRL-PROTAC is focused on the de novo design of complete PROTACs by effectively joining the designed ligands targeting the proteins of interest (POI) with linkers. The ligands for POIs and E3 ligases are generated by a molecular generation model for targeting proteins, and the linker design is generated by a sequence-to-sequence model consisting of a transformer variant and the policy-based reinforcement learning method which is employed to optimize the reward values for generating PROTACs. The three components are then integrated and optimized based on their pharmacokinetic (PK) and degradation (DEG) properties. The experimental results have strongly confirmed that TRL-PROTAC is superior in optimizing properties. For existing PROTACs, TRL-PROTAC improves DEG scores by 0.45 and lowers PK scores by 1.20. Moreover, TRL-PROTAC enhances binding affinity by 2.15 in PROTACs generated from scratch.

Keywords: proteolysis-targeting chimeras, transformer, reinforcement learning, drug design, protein-ligand interaction.

1. Introduction

Proteolysis-targeting chimeras (PROTACs) are a promising new class of drugs [1], [2], [3]. The PROTAC molecule consists of three parts: one ligand (warhead) binding to the POI, one ligand (E3 ligand) for recruiting an E3 ubiquitin ligase and a chemical bridge (linker) that links the two ligands. This enables the target protein and ubiquitin E3 ligase to be recognized and induced to undergo polyubiquitination, leading to target protein degradation and disease treatment [4]. Unlike traditional occupancy-driven inhibitors that require sufficient binding affinity to the druggable site of the target protein to exert their effects, PROTACs utilize event-driven mechanisms and only need to bind briefly to

* Corresponding author

the target protein to induce its ubiquitination and degradation. The advantages of event-driven drugs such as PROTACs are that they enable the efficacy of drugs even at sub-stoichiometric concentrations [5]. Moreover, an equivalent PROTAC can degrade multiple equivalent target proteins and avoid off-target effects caused by high drug concentrations. Additionally, many proteins play important roles in cancer development but lack druggable pockets, making them "undruggable" by small molecule drugs [6]. However, the emergence of PROTAC technology has made most protein targets in human cells "druggable", greatly expanding the therapeutic prospects of difficult-to-treat diseases such as metabolic syndrome. Moreover, the ability to persistently block downstream signals and delay kinase recombination reduces the likelihood of target compensation, as demonstrated in multiple biological evaluations of PROTACs [7].

The process of traditional drug development is notoriously low in success rate, given that it involves target validation, high-throughput screening, optimization of lead compounds, preclinical trials, and clinical trials, all of which are expensive and time-consuming [8]. However, the development of artificial intelligence (AI) has had a significant impact on the pharmaceutical industry, reducing the time and cost of drug development and shortening the development cycle [9], [10], [11]. AI has emerged as an important tool in computational-aided drug design [12], [13], [14], [15], [16]. Currently, AI-assisted drug design methods can be primarily classified into three categories: structure-based methods, ligand-based methods [17], and hybrid methods [18]. Among the structure-based methods, those focusing on protein structure and the dynamic properties of proteins are emerging as a promising direction for development [19]. Amongst, the representative AlphaDrug model was proposed by Qian et al. [20] for the de novo generation of molecules targeting specific target proteins. The Monte Carlo tree search (MCTS) algorithm was used to search for candidate drugs in all possible molecular spaces. This approach enabled candidate drugs to correctly dock with protein targets in an exclusive manner. Another representative model was the reinforcement learning (RL) model proposed by Olivecrona et al. [21]. This model utilized Recurrent Neural Networks (RNNs) [22] in combination with the reinforcement learning method for de novo drug design and introduced an augmented likelihood function for training.

Initially developed in different domains, machine learning (ML) techniques are increasingly being applied to drug discovery due to their potential to expedite the entire drug development process. As a result, a wide range of ML methods, including Artificial Neural Networks (ANNs) [23], RNNs [22], Long Short-Term Memory (LSTM) networks [24], and Transformer models [25], are being applied in drug design, particularly in addressing the challenges of predicting compound bioactivity. In contrast, deep learning in PROTACs is still in its infancy. PROTACs are distinct from conventional small-molecule drugs in that they consist of three components. When considering the generation of PROTACs, not only the effectiveness of the generated individual small molecules should be considered, but also the pharmacological properties of the overall molecule, particularly the linker generation [26]. Due to the complex and dynamic nature of the ternary structure, linker design remains a significant challenge.

Previous studies have shown that representing PROTACs as SMILES strings is feasible for linker generation. Yang et al. [27] transformed linker design into a sentence completion task and introduced a language model SyntaLinker to generate novel linkers. Zheng et al. [26] used a pair of warhead and E3 ligand as input and outputted the

designed PROTACs with favorable properties that are chemically feasible. To generate PROTACs with better pharmacokinetic (PK) properties, a Proformer model [26] was developed through pre-training and fine-tuning. This model was then integrated into a memory-augmented reinforcement learning framework with an experience reward function, which facilitated the generation of optimized PROTACs. Li et al. [8] proposed the DeepPROTACs model to help design effective PROTACs. Ligands and binding pockets were represented as graphs, and a graph convolutional network was applied to extract their features. As for linkers, they were fed into the model as SMILES strings and features were extracted using a bidirectional LSTM layer.

Zheng et al. [26] made remarkable contributions in the field of PROTACs, but their model only considered the PK property of generated PROTACs, where a smaller PK value indicates a higher oral availability of the molecule. However, in some cases, optimizing for a smaller PK value can lead to a smaller DEG value, which reduces the molecule's degradation efficacy, and this is not desirable. Furthermore, when more complex warheads and E3 ligands were input into their model, beyond the scope of the training and testing datasets, the model was unable to generate valid PROTAC molecules. The likely reason was that their molecular generation task involved first generating complete PROTAC molecules and then checking for the presence of warheads and E3 ligands. This task required the model to generate accurate warheads and E3 ligands as a prerequisite. However, the ultimate goal was to generate linkers, which made the generation of warheads and E3 ligands an additional burden for the model.

In this work, a rational strategy (TRL-PROTAC) targeting specific proteins for the design and evaluation of PROTACs with both better pharmacokinetic (PK) properties and higher degradation (DEG) efficacy was proposed for the first time. To address the issues raised in the preceding paragraph, we proposed a novel molecular generation task in which linkers were directly generated from the given warheads and E3 ligands using a pre-trained generator, the Lmser Transformer. Subsequently, these generated linkers were concatenated with the original components to obtain complete PROTACs. To acquire warheads and E3 ligands that possess higher binding affinity to target proteins, we employed the AlphaDrug model [20] to generate novel warheads and E3 ligands. To obtain PROTACs with desired properties, a policy-based reinforcement learning method is employed. Additionally, the probabilities predicted by the DeepPROTACs model were incorporated as DEG scores into the reward function of the reinforcement learning method along with PK scores, and a reward function was creatively proposed that simultaneously optimized both PK and DEG properties during the molecule generation process.

Objectives. The objectives of our work can be summarized as follows:

- Designing complete PROTACs targeting the POI. For this objective, we introduced a novel approach to design and optimize PROTACs targeting the POI (Section 3.1).
- Solving the model training problem for PROTACs with small amounts of data. For this objective, we employed a pre-training and fine-tuning approach on the Lmser Transformer model, which has been demonstrated to possess superior capabilities in drug design [20], aimed to address challenges stemming from the limited dataset (Section 3.2).
- Optimizing the properties of model-generated PROTACs. For this objective, we utilized a reinforcement learning approach with an augmented likelihood function to

optimize the Lmser Transformer, aiming to generate molecules with desirable properties (Section 3.3).
- Enhancing binding affinity between the warhead and the POI and between the e3 ligand and the e3 ligase. For this objective, the AlphaDrug model [20] was employed to generate warheads and E3 ligands with enhanced binding affinity to the POI (Section 2.1).

Contributions. Our contributions can be summarized as follows:

- We proposed an effective strategy for the rational design of PROTACs targeting specific proteins, rather than designing PROTACs based on existing warheads and E3 ligands pairs.
- We introduced a novel approach to design PROTACs, which involves generating linkers with attachment points first, and then connecting these linkers to warheads and E3 ligands.
- We considered the pharmacokinetic and degradation properties of PROTACs in the process of molecular generation by defining an appropriate reward function.
- We meticulously collected and provided structural data for the protein pockets targeted by PROTACs.
- We conducted extensive experiments and visually presented the results through illustrations.

The structure of the remaining sections of this paper is outlined as follows. Section 2 presents a review of related work and the underlying motivation. In Section 3, we delve into the specifics of our method. Section 4 elaborates on the experimental setups involving the use of the model, including data preparation. Section 5 outlines the experimental validation of the model and the subsequent analysis of the results. Section 6 discusses the potential applications of our method. Finally, Section 7 concludes and summarizes the findings.

2. Related work and motivation

This section encompasses related work and the underlying motivation. Subsection 2.1 introduces the AlphaDrug model used for de novo design of conventional small molecule drugs, highlighting the insights it brings to this study. This subsection also provides details about the AlphaDrug model. Subsection 2.2 presents the methodology employed by the DeepPROTACs model and its implications for this endeavor. Subsections 2.3 to 2.5 elaborate on the details of the three methods: Beam Search, Temperature Sampling and the PROTAC-RL model.

2.1. AlphaDrug

AlphaDrug [20] is an advanced deep learning model that possesses the capability to design ligands tailored to target proteins, taking into consideration the binding affinity between the protein and ligand during the molecule generation process. Proteins are represented as sequences of amino acids, and ligands are represented as SMILES strings. To facilitate the inefficiency which is due to the information transfer bottleneck from the top layer

of the encoder to various levels of decoder layers, a transformer variant called Lmser Transformer is utilized [20]. By integrating AlphaDrug into the TRL-PROTAC framework, we were empowered to initiate the design of PROTACs from scratch. The process commenced with the identification of high-affinity warheads specific to target proteins and high-affinity E3 ligands for specific E3 ligases. Subsequently, attachment points were established to link the designed ligands, resulting in the formation of warheads and E3 ligands pairs, pivotal for the subsequent generation of PROTACs.

Additionally, the AlphaDrug model was trained on the PROTACs dataset and compared with our model. The SMILES forms of warheads and E3 ligands are used as input to the Lmser Transformer, while the SMILES forms of linkers are used as the output. Furthermore, the MCTS algorithm is employed to optimize the properties of generated PROTACs. It plays a crucial role in AlphaDrug. The MCTS algorithm is pivotal in reinforcement learning and decision-making tasks, extensively applied to explore vast state spaces and determine optimal actions [28]. It finds applications in domains such as board games, game AI, and resource allocation [29].

MCTS's core idea is employing Monte Carlo simulations to estimate action values and guiding the search process through a constructed tree. In this work, we have introduced modifications to the MCTS process by incorporating the pre-trained Lmser Transformer for simulation purposes. The algorithm comprises four key steps:

Step 1. Selection: Starting from the root node, nodes are traversed using a certain strategy until an unexpanded node or a terminal state is reached. A modified version of the Predictor with Upper Confidence bounds applied to Trees (PUCT) algorithm [30] is used, which prioritizes nodes with fewer visits. The PUCT algorithm equation for selecting child nodes is:

$$\tilde{a}_{\mathrm{t}} = \underset{a \in \mathcal{A}}{\arg\max} \left(\mathrm{Q}\left(\tilde{C}_{\mathrm{t}-1}, a\right) + U\left(\tilde{C}_{\mathrm{t}-1}, a\right) \right) \tag{1}$$

where $\tilde{a}_{\mathrm{t}}$ is a child node after t selections, $\tilde{C}_{\mathrm{t}-1}$ represents the context with the pair of warhead and E3 ligand, and $Q\left(\tilde{C}_{t-1}, a\right) = W_a/N_a$ represents the expected reward for choosing symbol a in context $\tilde{C}_{\mathrm{t}-1}$, with W_a and N_a representing the total reward and visit count for the node; $U\left(\tilde{C}_{t-1},a\right) = c_{\text{puct}} P\left(a \,|\, \tilde{C}_{t-1}\right) \sqrt{N_t} / \left(1+N_t(a)\right)$, with c_{puct} controlling the degree of exploration.

Step 2. Expansion: If the chosen node has not been expanded, one or more child nodes are added, representing potential actions.

Step 3. Simulation: Expanded child nodes are simulated via Lmser Transformer, generating outcomes based on reward or scoring rules.

Step 4. Backpropagation: Simulation rewards propagate back along the path from the simulated node to the root, updating statistics such as visit counts and cumulative rewards.

MCTS has widespread applications in reinforcement learning, game strategy, and optimization problems. Its successful use in computer programs like AlphaGo underscores its potent search and decision-making capabilities.

2.2. DeepPROTACs

DeepPROTACs [8] is a deep learning model specifically designed to predict the degradability of PROTACs. In the DeepPROTACs model, ligands and ligand binding pockets are

generated and represented as graphs. These graph representations are then inputted into Graph Convolutional Networks (GCNs) to extract relevant features. On the other hand, SMILES representations of linkers are passed through a Bidirectional Long Short-Term Memory (BiLSTM) layer to generate their respective features [8]. In our research, we integrated the predicted probability from DeepPROTACs into the TRL-PROTAC model to guide the determination of PROTAC degradability. By incorporating this probability within the reward function of the policy-based reinforcement learning method, we were able to consider the overall degradability of PROTACs during the process of linker generation. This integration facilitated the generation of PROTACs with enhanced degradability, optimizing their effectiveness in targeted protein degradation.

2.3. Beam Search

Beam Search is a widely used decoding algorithm in sequence generation tasks, often applied to tasks like machine translation and text generation [31]. It aims to find the most likely sequence of tokens given a trained generative model. Unlike greedy decoding, which selects the token with the highest probability at each step, Beam Search maintains a fixed-size set of candidate sequences, known as the beam width.

The core idea of Beam Search is to explore multiple possible sequences simultaneously by considering a limited set of promising candidates. At each decoding step, the algorithm expands each candidate sequence by generating all possible next tokens and calculates their probabilities using the model. Then, it selects the top-k candidates with the highest probabilities based on a scoring function. These candidates become the new set of candidates for the next step. The scoring function combines the probability of the current sequence with the probability of the new token, which can be defined as:

$$\text{Score}(s, y) = \log P(y|s) \tag{2}$$

where s is a candidate sequence, y is a new token, and $P(y|s)$ is the probability assigned by the generative model.

The algorithm repeats this process iteratively until the sequences reach the desired length. One key parameter is the beam width, denoted as k, which determines how many candidate sequences are kept at each step. Larger k values allow for more exploration but may increase computation time. While Beam Search is efficient and can improve output quality compared to greedy decoding, it may suffer from generating repetitive sequences and missing out on diverse alternatives. Researchers have proposed variations like length normalization and diverse beam search to address these issues.

In summary, Beam Search strikes a balance between exploration and exploitation, leveraging a fixed-size set of candidate sequences to find high-probability solutions in sequence generation tasks.

2.4. Temperature Sampling

When discussing the outputs of generative models, "Temperature Sampling" [32] stands as a commonly used technique to modulate the diversity and determinism of model-generated results. It introduces a temperature parameter τ to reweight the output probability distribution, thereby influencing the generated outcomes. A higher temperature value

τ smoothens the distribution, encouraging the model to sample from lower probability options and enhancing result diversity. Conversely, a lower temperature value τ narrows the distribution, prompting the model to favor high probability options and generating relatively deterministic outcomes.

This temperature adjustment process can be expressed using the following equation:

$$\mathrm{Softmax}(z)_i = \frac{\exp\left(z_i/\tau\right)}{\sum_{j=1}^{n} \exp\left(z_j/\tau\right)} \tag{3}$$

where z_i represents the logits prior to softmax transformation, n denotes the possible number of symbols, and i indicates the index of a specific symbol. By dividing logits by the temperature parameter τ and then performing softmax normalization, a new probability distribution $\mathrm{Softmax}(z)$ is obtained, where values reflect the probabilities of selecting each symbol.

2.5. PROTAC-RL

The PROTAC-RL model [26] is a deep learning model designed to generate complete PROTACs based on existing warheads and E3 ligands. This model utilizes reinforcement learning to optimize the PK property of PROTACs. The model begins by one-hot encoding the source sequence (comprising warheads and E3 ligands sequences) and subsequently embedding it into a latent representation denoted as $\mathcal{H}$. The decoder leverages $\mathcal{H}$ to generate a probability distribution for the tokens, which are sampled iteratively. Sampling continues until the model encounters the ending token "</s>", ultimately forming the output sequence (the complete PROTACs sequence) $Y = (y_1, \ldots, y_m)$. The model's objective is to minimize the cross-entropy loss $\mathcal{L}$, computed between the target sequence $M_t = (t_1, \ldots, t_k)$ and the generated sequence Y:

$$\mathcal{L}(Y, M) = -\sum_{i=1}^{k} y_i \log t_i \tag{4}$$

The PROTAC-RL model optimizes the generation of PROTACs attributes through a Markov decision process. In this process, the action $A = (a_1, a_2, \ldots, a_m)$ represents the sampling of tokens from a dictionary. The state $S = (s_1, s_2, \ldots, s_m)$ is a hidden state, representing intermediate molecules sampled in previous steps. The policy π refers to the probability distribution of tokens learned by the model from the data. The reward $R(Y)$ is defined using the following equation [26]:

$$R(Y) = \begin{cases} \max\left(0, 1 - \frac{1}{\alpha}|f(Y) - \text{ Target }|\right), & \text{if valid;} \\ 0, & \text{if invalid.} \end{cases} \tag{5}$$

where α is a scalar coefficient, $f(Y)$ is a function for calculating attributes, and Target represents the custom target attribute value.

3. Methodology

In this section, we present the concrete methodology employed. Subsection 3.1 demonstrates the overarching design approach for complete PROTACs. Subsection 3.2 details

the training of ProLinker using pre-training and fine-tuning methods, along with the process of designing from warheads and E3 ligands to linkers and merging the three components. Subsection 3.3 elucidates the training of the policy network using policy-based reinforcement learning.

3.1. Architecture of TRL-PROTAC

An integrated protocol for design and evaluation of PROTACs (Figure. 1) is proposed to generate PROTACs with ideal properties (PK and DEG). As presented in Figure. 1, we first utilized the AlphaDrug model to generate warheads and E3 ligands with high binding affinity to the target proteins. To generate valid linkers, a novel end-to-end generation task was proposed for the rational design of linkers from warheads and E3 ligands. In pursuit of this goal, we initially pre-trained the Lmser Transformer, a transformer variant from the AlphaDrug model, on a large dataset of quasi-PROTAC molecules. Subsequently, the model was fine-tuned on a dataset of actual PROTAC molecules. The quasi-PROTAC dataset and the actual PROTACs dataset are two different datasets. The actual PROTACs dataset contains the PROTACs molecule, which is a ternary complex consisting of the warhead, the linker and the e3 ligand. Whereas, the quasi-PROTAC dataset is a small molecule dataset, where each data entry consists of only one small molecule. The purpose of the model being trained on the quasi-PROTAC dataset first is to allow the model to learn the syntactic information of the SMILES sequences. As a result, the trained model named ProLinker was able to generate linkers. Following that, the reinforcement learning method was utilized for step-by-step molecule generation to generate PROTACs with ideal properties.

Our model took warhead, POI pocket, E3 ligand and E3 pocket as input. In addition, the ligands are encoded by the model in the form of SMILES strings, while the pockets are encoded using mol2 files. All SMILES strings have already been tokenized to construct a vocabulary. ProLinker was applied for sampling and predicting the next token based on intermediate molecules. The reinforcement learning method generated complete molecules by performing sampling and subsequently calculated the reward score by RDKit [33] of each molecule. Formally,

$$\mathrm{PK}(X) = \mathrm{AB-MPS} = |\mathrm{LogP}(X) - 3| + \mathrm{NAR} + \mathrm{NRotB} \tag{6}$$

$$\mathrm{Reward}(X) = \mathrm{DEG} * \frac{1}{|\mathrm{LogP}(X) - 3| + \mathrm{NAR} + \mathrm{NRotB}} * 10 \tag{7}$$

where X is a PROTAC represented in SMILES format, LogP is commonly used to indicate the lipophilicity of a compound, NAR represents the number of aromatic rings, NRotB denotes the number of rotatable bonds, and DEG is the probability of degradation predicted by the DeepPROTACs model.

During the step-by-step generation process, the algorithm selects the optimal token for the current step by taking into consideration the feedback score obtained from the previous steps. According to the research conducted by DeGoey et al. [34], a notable negative correlation (correlation coefficient $r = -0.41$) exists between AB-MPS and oral availability. It was observed that a lower AB-MPS score is indicative of a higher oral availability of the drug. Apart from that, the DEG score predicted by DeepPROTACs represented the

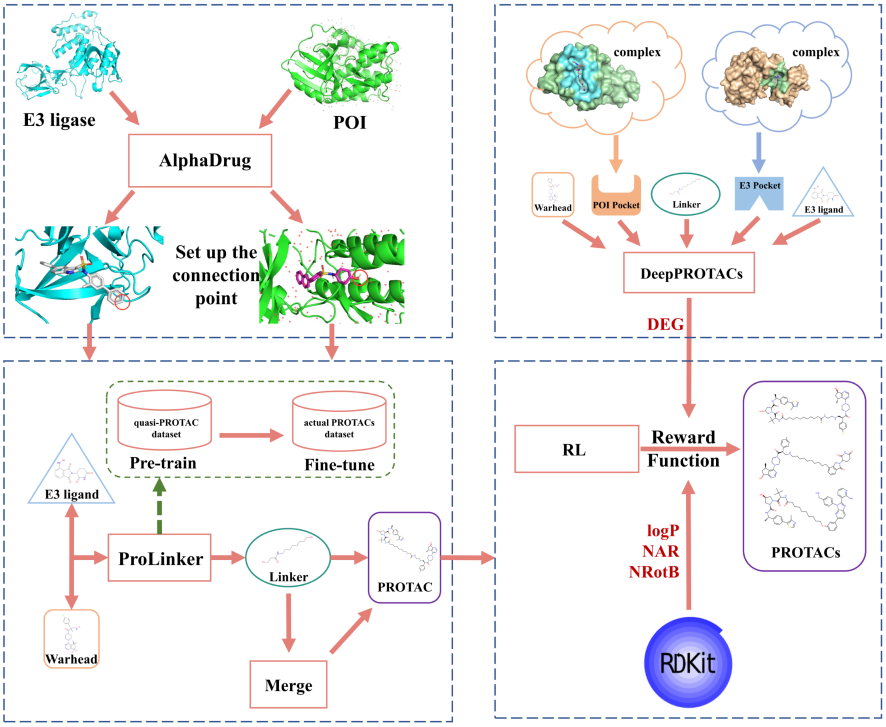


Fig. 1. Flowchart of TRL-PROTAC which is a rational strategy for design and evaluation of PROTACs targeting specific proteins. It encompasses three distinct components: the generation of warheads and E3 ligands, the generation of linkers, and the optimization of PROTAC properties

degradation efficacy of given PROTACs. A higher DEG score indicates a greater probability of degradation. The final reward function is derived from the integration of the two aforementioned scores. Due to the fact that the resulting final score obtained by dividing the DEG value by the PK value is relatively small, it has been decided to amplify the final score by a factor of 10 and utilize it as the reward value.

3.2. ProLinker

ProLinker is a crucial step in the TRL-PROTAC workflow. Building upon the validated efficacy of the Lmser Transformer architecture in the AlphaDrug model [20], ProLinker adopted the same framework for its training process. Given the inherent scarcity of available datasets specific to PROTACs, our approach involved an initial pre-training phase on a substantial collection of quasi-PROTAC molecules, followed by a fine-tuning stage utilizing actual PROTACs. Detailed information about important parameters of ProLinker is obtained in Section 4.3. Detailed procedures regarding pre-training and fine-tuning can be found in Algorithm 1 and Algorithm 2.

Algorithm 1 Pre-training with Lmser Transformer

Require: Pre-training data: quasi-PROTAC molecules D_{quasi}, learning rate: lr
Ensure: Pre-trained model parameters: $\theta_{\text{pre-trained}}$
Initialize Lmser Transformer model with random weights: $model \leftarrow$ Lmser_Transformer()
Initialize optimizer: $optimizer \leftarrow$ Adam($\theta_{\text{pre-trained}}$, lr)
for $epoch \leftarrow 1$ to num_epochs **do**
    for $batch \in D_{quasi}$ **do**
        $input_batch, target_batch \leftarrow$ prepare_batch($batch$)
        $out \leftarrow model(input_batch)$
        $loss \leftarrow$ Calculate the nll loss of out and $target_batch$
        Backward propagation of the $loss$
        Update parameters of the $optimizer$
    end for
end for
$\theta_{\text{pre-trained}} \leftarrow$ parameters of the $model$

Algorithm 2 Fine-tuning with Lmser Transformer

Require: Fine-tuning data: actual PROTACs D_{actual}, pre-trained model: $\theta_{\text{pre-trained}}$, learning rate: lr
Ensure: Fine-tuned model parameters: $\theta_{\text{fine-tuned}}$
Initialize Lmser Transformer model with $\theta_{\text{pre-trained}}$: $model \leftarrow$ Lmser_Transformer($\theta_{\text{pre-trained}}$)
Initialize optimizer: $optimizer \leftarrow$ Adam($\theta_{\text{fine-tuned}}$, lr)
for $epoch \leftarrow 1$ to num_epochs **do**
    for $batch \in D_{actual}$ **do**
        $input_batch, target_batch \leftarrow$ prepare_batch($batch$)
        $out \leftarrow model(input_batch)$
        $loss \leftarrow$ Calculate the nll loss of out and $target_batch$
        Backward propagation of the $loss$
        Update parameters of the $optimizer$
    end for
end for
$\theta_{\text{fine-tuned}} \leftarrow$ parameters of the $model$

Algorithm 1 is designed to initialize a Lmser Transformer model by training it on a dataset of quasi-PROTAC molecules. The objective is to equip the model with meaningful representations of molecular structures. The training loop consists of iterating over a predefined number of epochs, with each epoch comprising batches of quasi-PROTAC molecules. For each batch, the "input_batch" refers to the SMILES sequence of warhead and e3 ligand with attachment points. And the "target_batch" refers to the SMILES sequence of the linker with attachment points on both sides. Then the model predicts molecular structures. The negative log-likelihood (nll) loss is then calculated to quantify the disparity between the model's predictions and the actual targets. Through backpropagation, the model's parameters are adjusted to minimize this loss. This iterative optimization process is facilitated by the Adam optimizer. Algorithm 2 is designed for training on actual PROTAC molecules. This process involves refining a pre-trained Lmser Transformer model using a dataset of actual PROTAC molecules (D_{actual}). The pre-trained model ($\theta_{\text{pre-trained}}$), previously obtained through Algorithm 1, serves as the starting point for fine-tuning.

It is worth noting that we took a completely new approach to generate PROTACs. We generated linkers sequences with attachment points from the sequences of warheads and E3 ligands pairs. These generated linkers were then connected to warheads and E3 ligands using the attachment points to obtain complete PROTACs sequences. This differed from the conventional method of directly generating complete PROTACs sequences from the pairs of warheads and E3 ligands sequences. The input format for warheads and E3 ligands pairs was structured as "*N1CCN(c2ccc(Nc3ncc4cc(C(=O)N(C)C)n(C5CCCC5)c4n3)nc2)CC1.*Nc1cccc2c1C(=O)N(C1CCC(=O)NC1=O)C2=O", where "*" denoted the attachment point and "." represented the boundary between the warhead and the E3 ligand. The input sequence is initially tokenized into individual tokens from the vocabulary, resulting in a sequence of tokens. Subsequently, the token sequence is mapped to numerical vectors based on the positions of different tokens in the vocabulary. Vectors were then input into the model for encoding by the encoder. The model received information from the encoder for processing and outputted a probability distribution over different tokens in the vocabulary through the decoder. Afterward, a sequence was generated by sampling based on this probability distribution. The output format for linkers was represented as "[1*]CCOCCOCC[2*]", where "[1*]" denoted the attachment point between the linker and the warhead, and "[2*]" denoted the attachment point between the linker and the E3 ligand. By adopting this generation approach, the model was still capable of generating valid linkers and assembling them into functional PROTACs, even in cases where the structures of warheads and E3 ligands were complex.

We trained the Lmser Transformer model on prepared PROTACs datasets with the objective of minimizing the following loss function:

$$J(\Theta) = -\sum_{(S,m)\in\mathcal{D}}\sum_{\tau=1}^{t}\sum_{a\in\mathcal{V}} y_a \ln P\left(a \mid C_\tau(S,m)\right) \tag{8}$$

where Θ denotes the parameters of the Lmser Transformer network, $\mathcal{D}$ is the training dataset, $\mathcal{V}$ is the vocabulary, S is the input sequence, $m = (a_1, a_2, \ldots, a_t)$ is the target sequence, $C_\tau = \{S, (a_1, a_2, \ldots, a_\tau)\}, \tau = 1, \ldots, t$, represents a sequence of contexts, $P(a \mid C_\tau)$ is the probability of the next token $a_{\tau+1}$, and y_a is a binary label indicating whether a is the next token.

3.3. Optimizing policy network with reinforcement learning

A policy-based reinforcement learning method is employed to optimize the policy network. The policy-based approach constitutes a significant methodology within reinforcement learning [35], [36]. Policy-based approaches enable the explicit learning of an optimal stochastic policy [37]. Figure. 2 provides a comprehensive depiction of the procedural steps involved in training the agent through the utilization of reinforcement learning. This method shares the same concept as that of [21].

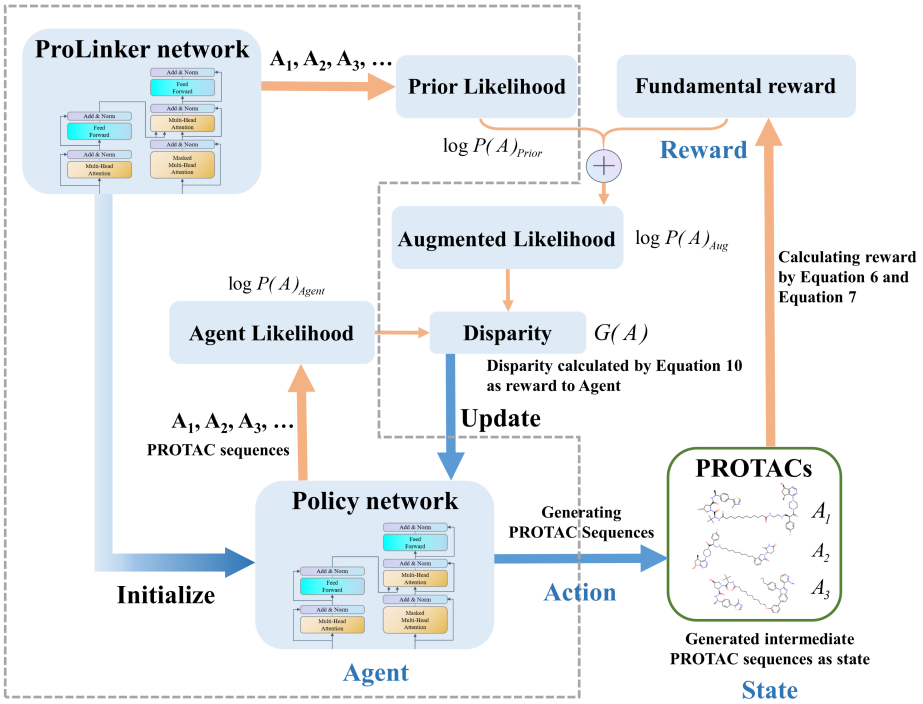


Fig. 2. The illustration of constructing the policy-based reinforcement learning model. Continuous actions are taken by the policy network until a complete PROTAC sequence is generated. Subsequently, the fundamental reward is calculated based on Formula 6 and Formula 7. The ProLinker network computes the prior likelihood for the generated PROTAC sequence. The augmented likelihood is obtained by combining the fundamental reward with the prior likelihood. The disparity between the augmented likelihood and the agent likelihood is then used as the final reward to update the agent

The reinforcement learning training process usually entails equipping an agent with the task of choosing an action, denoted as $a \in A(s)$, based on a specific state, $s \in S$, where S encompasses all possible states [38]. Here, $A(s)$ stands for the set of actions available in those states. The agent's policy, denoted as $\pi(a \mid s)$, indicates the probability

of selecting action a in state s, and $r(a \mid s)$ represents the resulting outcome or reward following a specific action. In this work, the reinforcement learning modeling is outlined as follows.

Reinforcement learning modeling.

- **Agent:** The policy network responsible for generating PROTAC sequences.
- **State** $s \in S$: The intermediate molecule in the process of PROTAC generation.
- **Action** $a \in A(s)$: Selection of the next token in the vocabulary based on the intermediate molecule.
- **Reward** $G(A)$: Accumulated reward obtained by calculating molecular properties after generating a complete molecule.

We intend to utilize the Lmser Transformer architecture as the agent to proficiently generate PROTACs represented in the form of SMILES strings, while ensuring the incorporation of desired attributes. This task is framed within the context of a partially observable Markov decision process, where the agent's objective is to determine the subsequent token a in the vocabulary, based on the given intermediate sequence s. An episodic task refers to a task with a well-defined endpoint after undergoing step T [39]. This task starts sampling from the "&" symbol and concludes upon sampling "$" making it an episodic task. In this task, the action, denoted as a, represents the selection of a token from the vocabulary $\mathcal{V}$ to be added to the generated sequence. The state s refers to the currently generated intermediate sequence. The policy $\pi(a \mid s)$ signifies the probability distribution for choosing the next token in the intermediate sequence. $r(a \mid s)$ represents the reward value obtained after selecting the next token. In the molecular generation task, as the properties of a molecule are only meaningful once the entire molecule is generated, we calculate the cumulative reward $G(A)$ after generating the complete molecule $A = a_1, a_2, ..., a_T$. The product of action probabilities $P(A) = \prod_{t=1}^{T} \pi\left(a_t \mid s_t\right)$ signifies the likelihood of the generated sequence by the model.

The agent is characterized by a structural configuration akin to that of the ProLinker network, and it is initialized using ProLinker network parameters that have undergone meticulous fine-tuning. The agent policy π is securely tethered to the pre-existing ProLinker policy, skillfully amalgamating the rewards acquired from the attribute values of the sampled PROTACs. This leads to a novel augmented likelihood $\log P(A)_{\text{Aug}}$. Formally,

$$\log P(A)_{\text{Aug}} = \log P(A)_{\text{Prior}} + \sigma Reward(A) \tag{9}$$

where $P(A)_{\text{Prior}}$ represents the prior likelihood calculated by the ProLinker model after step T, σ is a scalar coefficient, and $Reward(A)$ represents the property value of the generated molecule and is calculated using equation 7. The objective of the agent is to acquire a policy that effectively mitigates the disparity between the agent likelihood $\log P(A)_{\text{Agent}}$ and the augmented likelihood $\log P(A)_{\text{Aug}}$. This disparity will be represented as the reward $G(A)$. Formally,

$$G(A) = -\left[\log P(A)_{\text{Agent}} - \log P(A)_{\text{Aug}}\right]^2 \tag{10}$$

The current objective of the agent is to optimize the reward value $G(A)$, which is attained by minimizing the loss function $J(\theta) = -G$. The detailed specifics of the policy-based optimization algorithm can be found in Algorithm 3.

Algorithm 3 is devised for optimizing the policy network through reinforcement learning. It utilizes the fine-tuned model ($\theta_{\text{fine-tuned}}$) to initialize both the prior network ($Prior$) and the policy network ($Agent$). The parameters of the fine-tuned model are retained as frozen prior model parameters (θ_{prior}). The optimization process involves sampling a batch of molecules using the agent, computing likelihoods for both the agent and prior networks, calculating scores, and then updating the parameters based on the loss function. The aim is to obtain updated agent model parameters (θ_{agent}) over a specified number of epochs.

Algorithm 3 Optimizing policy network with reinforcement learning

Require: Fine-tuned model: $\theta_{\text{fine-tuned}}$, batch size: $batch_size$, learning rate: lr
Ensure: Updated agent model parameters: θ_{agent}, Frozen prior model parameters: θ_{prior}
Initialize prior network ($Prior$) with $\theta_{\text{fine-tuned}}$: $Prior \leftarrow$ Lmser_Transformer($\theta_{\text{fine-tuned}}$)
Initialize policy network ($Agent$) with $\theta_{\text{fine-tuned}}$: $Agent \leftarrow$ Lmser_Transformer($\theta_{\text{fine-tuned}}$)
Initialize optimizer: $optimizer \leftarrow$ Adam($\theta_{\text{fine-tuned}}$, lr)
for $epoch \leftarrow 1$ to num_epochs **do**
    Initialize the set of molecules $\mathbf{D_s}$
    for $i \leftarrow 1$ to $batch_size$ **do**
        $a_0 \leftarrow$ the start symbol
        for $t \leftarrow 1$ to T **do**
            Select action $a_t \leftarrow \arg\max_a \pi_{agent}(a_{t-1}|s_{t-1})$
        end for
        $A_i \leftarrow a_0 a_1 \dots a_T$
        Add A_i to $\mathbf{D_s}$
    end for
    $agent_likelihood \leftarrow$ Compute the likelihood of $\mathbf{D_s}$ on $Agent$
    $prior_likelihood \leftarrow$ Compute the likelihood of $\mathbf{D_s}$ on $Prior$
    $score \leftarrow$ calculate scores of $\mathbf{D_s}$
    $augmented_likelihood \leftarrow prior_likelihood + \sigma score$
    $loss \leftarrow (augmented_likelihood - agent_likelihood)^2$
    Backward propagation of the $loss$
    Update parameters of the $optimizer$
end for
$\theta_{\text{agent}} \leftarrow$ parameters of the $Agent$

4. Experimental settings

This section encompasses pertinent details of the experimental setup. Subsection 4.1 illustrates the data preparation process for the subsequent experiments. Subsection 4.2 provides a detailed overview of the comparative methods employed. Subsection 4.3 presents the hyperparameter settings for pre-training and fine-tuning. Subsection 4.4 outlines the hyperparameters for the reinforcement learning method used in training the policy network.

4.1. Data preparation

4.1.1 Data for training ProLinker

The datasets used for training and validating ProLinker were curated by Zheng et al. [26] The pre-training dataset consisted of a collection of quasi-PROTAC small molecules carefully selected from the ZINC dataset. These molecules were chosen based on their similarity in chemical space to PROTACs. The fine-tuning data, on the other hand, comprised actual PROTACs obtained from the PROTAC-DB database.

These datasets consist of input data (src) and target data (tgt). The input data comprises pairs of warheads and E3 ligands, while the target data consists of complete PROTACs. To facilitate the proposed novel task, we have retained the original input data while modifying the target data from complete PROTACs molecules to individual linkers. Finally, a total of 204,840 quasi-PROTAC small molecules and 6,556 actual PROTACs were acquired. Following the methodology of previous investigations, these datasets were partitioned into training and validation sets in a ratio of 8:1.

4.1.2 Data for generating PROTACs based on existing warheads and E3 ligands

To validate the effectiveness and superiority of the ProLinker and reinforcement learning method combination, we first employed them to generate PROTACs on the existing dataset. The preprocessing of the dataset is depicted in Figure. 3. Due to the fact that the ProLinker model has already been trained on the training and validation sets, only the test set (consisting of a total of 165 pairs of Warhead and E3 Ligand molecules) was selected for linker generation. By using the structure search tool provided by the PROTAC-DB database [40], the corresponding information for the PROTACs in the database (a total of 162 molecules) can be retrieved. The PROTAC-DB database provides information such as the ID, name, target name, and E3 Ligase name for the molecules. Additionally, detailed information for both the Warhead and E3 Ligand molecules can be viewed, and links to external databases such as the ChEMBL database [41] are provided. In the ChEMBL database, information about the three-character representation of the ligand molecule in the PDBeChem database [42], [43], [44] can be obtained. Using the Uniprot [45] ID for the target protein and the three-character representation of the ligand molecule, the receptor protein-ligand small molecule complex structure can be retrieved from the RCSB PDB database [46]. A total of 54 pairs of warheads and E3 ligands have been obtained, all of which have well-defined complex structures with the target proteins. Detailed information about these 54 pairs of warheads and E3 ligands is obtained in Supplementary Table S1.

In the subsequent step, using the ProDy [47] package in Python, the protein pocket structure within 5 Å of the ligand molecule was obtained based on the name of the complex structure and the three-character representation of the ligand molecule. Subsequently, the protein pocket structure was converted from PDB format to MOL2 format using the Open Babel (obabel) tool [48]. Finally, the warheads, E3 ligands, and corresponding pockets were input into the model for PROTACs generation.

4.1.3 Data for de novo design of PROTACs targeting specific proteins

Subsequently, the ProDy package [47] was utilized to select individual chains from complex structures and separate the protein and ligand. The protein structure was saved in PDB format, while the ligand structure was saved in SDF format. A total of 28 unique

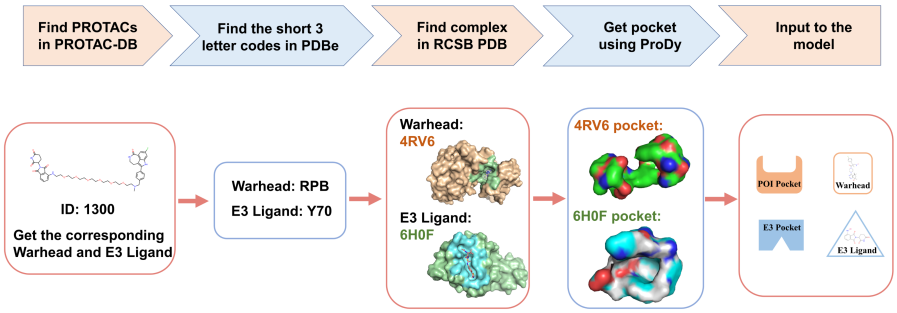


Fig. 3. The comprehensive workflow for data processing.

protein-ligand pairs were obtained. These two types of structures were then input into the AlphaDrug model for the generation of ligands with higher binding affinity. The generated molecules with the highest binding affinity will be selected as the ligands for the corresponding target proteins and E3 ligases for linker generation. The generated linkers will then be further processed and combined to obtain the final PROTACs.

4.2. Comparative methods

We compared our method with other methods, including Beam Search, Temperature Sampling, the AlphaDrug model and the PROTAC-RL model.

- Beam Search is an algorithm commonly employed in tasks involving the generation of sequences, like language translation or text generation. It operates by concurrently exploring several potential solutions, maintaining a restricted set of the most promising options (referred to as the "beam"), with the goal of identifying the most probable sequence based on a scoring mechanism.
- Temperature Sampling is a method commonly employed in probabilistic models, particularly in generative models like language models. It entails manipulating the temperature parameter during the sampling process, which in turn influences the diversity of generated outputs: higher values amplify randomness, while lower values induce a greater degree of predictability in the generated results.
- The AlphaDrug model is a deep learning model designed for creating custom ligands that are specific to target proteins, considering the binding affinity between proteins and ligands during molecule generation. It consists of a transformer variant known as the Lmser Transformer and the MCTS algorithm. The Lmser Transformer is used to address the inefficiency caused by information transfer bottlenecks in the encoder-decoder architecture. And MCTS is a sophisticated algorithmic strategy frequently applied for making decisions in intricate domains with multiple choices. MCTS systematically constructs a decision tree, intelligently exploring and exploiting possibilities, resulting in improved decision-making especially in situations marked by ambiguity and extensive options [30].

- PROTAC-RL is built upon the SyntaLinker [27] language model, enabling the generation of complete PROTACs sequences from existing warheads and E3 ligands sequences. The model consists of a pre-trained SyntaLinker model for learning the representation of PROTAC molecules and the reinforcement learning approach to optimize the PK property of generated PROTACs.

For specific descriptions of these methods, refer to Section 2.1 and Sections 2.3 to 2.5.

4.3. Hyperparameters in pre-training and fine-tuning

As the pre-training and fine-tuning employed the same model architecture, it was essential to maintain consistency in hyperparameters, as outlined in Table 1. Here, "fragsVoc" represents the vocabulary for input warheads and E3 ligands, while "LinkerVoc" represents the vocabulary for output linkers. "fragsMaxLen" and "LinkerMaxLen" correspondingly indicate the maximum lengths of input and output sequences. "fragsPaddingIdx" and "LinkerPaddingIdx" signify the indices of the start characters in input and output sequences. "frags_voc_len" and "Linker_voc_len" denote the vocabulary sizes. The following are common parameters used for training the Transformer model.

4.4. Hyperparameters in optimizing policy network

The hyperparameters for the policy-based optimization approach are detailed in Table 2. Here, "batch_size" indicates the number of samples used for training the policy network in each iteration. "epoch" and "lr" respectively denote the training epochs and learning rate for the policy network. "σ" represents the constant in Equation 9. "σ" is set to 2000 to balance the difference in magnitude between the prior likelihood computed by the ProLinker model and the reward scores.

5. Results and analysis

In the initial phase, we commenced with the pre-training of the ProLinker model on the quasi-PROTAC dataset, which was subsequently fine-tuned on the actual PROTACs dataset. Afterwards, utilizing the ProLinker model in conjunction with the reinforcement learning method, we generated 54 sets of PROTACs derived from corresponding pairs of warheads and E3 ligands (a detailed description of the data can be found in Section 4.1). Each set contains PROTACs with different PK values, DEG values, and reward values. We visualized the distribution of the scores of the generated PROTACs (Section 5.1), compared them with the original PROTACs (Section 5.2), and compared them with PROTACs generated by Beam Search, Temperature Sampling, AlphaDrug and PROTAC-RL (Section 5.3). Lastly, we embarked on de novo designs of PROTACs targeting the 33 specific proteins (after removing duplicates). And we selected the top 5 reward scores of PROTACs generated by AlphaDrug and TRL-PROTAC for comparison (Section 5.4). Moreover, we employed visualizations to enhance the analysis and evaluation of the designed PROTACs.

Table 1. Important hyperparameters of training ProLinker

hyperparameters	pre-training/fine-tuning
fragsVoc	"#", "$", "&", "(", ")", "*", "-", ".", "/", "1", "2", "3", "4", "5", "6", "7", "=", "B", "Br", "C", "Cl", "F", "I", "N", "O", "P", "S", "[B-]", "[C+]", "[C-]", "[C@@H]", "[C@@]", "[C@H]", "[C@]", "[C]", "[N+]", "[N-]", "[N@+]", "[N@@+]", "[O-]", "[O]", "[P+]", "[P@@]", "[P@]", "[S+]", "[S@@]", "[S@]", "[Si]", "[Sn]", "[c-]", "[n+]", "[nH]", "[s+]", "\\", "^", "c", "n", "o", "p", "s"
LinkerVoc	"#", "$", "&", "(", ")", "-", ".", "/", "1", "2", "3", "4", "5", "6", "7", ":", "=", "B", "Br", "C", "Cl", "F", "I", "N", "O", "P", "S", "[1*]", "[2*]", "[B-]", "[B@-]", "[BH-]", "[C+]", "[C-]", "[C@@H]", "[C@@]", "[C@H]", "[C@]", "[C]", "[IH2]", "[N+]", "[N-]", "[N@@+]", "[NH3+]", "[O-]", "[O]", "[P+]", "[P@@]", "[P@]", "[S+]", "[S@@]", "[S@]", "[Si]", "[Sn]", "[c-]", "[n+]", "[nH]", "[s+]", "\\", "^", "c", "n", "o", "s"
fragsMaxLen	202
LinkerMaxLen	91
fragsPaddingIdx	54
LinkerPaddingIdx	59
frags_voc_len	60
Linker_voc_len	64
batch_size	32
epoch	500
lr	0.0001
d_model	96
dim_feedforward	256
num_layers	4
nhead	4

Table 2. Important hyperparameters of training policy network

hyperparameters	optimizing policy network
batch_size	32
epoch	1000
lr	0.005
σ	2000

5.1. Horizontal comparison of all generated PROTACs

The ProLinker model is initially pre-trained on a quasi-PROTAC dataset containing a large number of small molecules to learn the chemical language of molecules represented by SMILES sequences. Subsequently, the ProLinker model undergoes fine-tuning on an actual PROTAC dataset to learn the specific chemical language of PROTACs. The detailed data of pre-training and fine-tuning processes is obtained in Supplementary Tables S2 and S3. The fine-tuned ProLinker is further optimized using a policy-based reinforcement learning approach to generate PROTACs with specific properties. We conducted 54 sets of experiments using existing warheads and E3 ligands to validate the effectiveness and superiority of the ProLinker model combined with the reinforcement learning method.

To provide a more intuitive representation of the distribution of scores for all generated PROTACs across multiple dimensions, we utilized the CMplot package [49] in R to generate the Manhattan plot for multiple scores. The Manhattan plot was originally designed to visualize the summary statistics results of genome-wide association studies (GWAS), especially p-values [50]. It can display the different p-values of multiple tracks at different positions of the genome. Here, we used three different scores as three different tracks and drew a circular scatter plot with PROTAC IDs as genomic positions. This plot enables us to visualize the distribution of the generated PROTAC scores from multiple dimensions, and also allows for the horizontal comparison between different PROTAC IDs.

As shown in Figure. 4, the outermost circle of the figure represents different IDs of PROTACs in PROTAC-DB, while the three trajectories from top to bottom represent the scores of PK, DEG, and reward score, respectively. This figure intuitively shows the score distribution of the model during the molecule generation process in three dimensions. It can be observed that most of the samples show long bar-shaped distributions for the three scores, indicating that the scoring of the model is not single-valued, but distributed from low to high. This further confirms that the reinforcement learning method does guide the generation of molecules with desired properties. Many molecules in the figure have PK values reaching around 15.75, and a considerable portion of molecules have DEG values above 0.5. The figure also allows for visual comparison between different molecules, showing that different molecules can be optimized to varying degrees. For example, the molecule with ID 939 has a reward value that is only presented as two dots. Because its DEG value is relatively low, and the space for optimizing this molecule is relatively small. On the other hand, the molecule with ID 1399 has a wide range of PK value changes, and can even reach a minimum of around 15, while its DEG value also changes significantly, ranging from 0 to 1. This indicates that the space for optimizing this molecule is quite large. Detailed information about these scores of these generated PROTACs is obtained in Supplementary Table S4.

5.2. Comparison with original PROTACs

In order to elucidate the substantial enhancement of our model-generated PROTACs over the original PROTACs, the original PROTACs were utilized to compute the DEG values, PK values, and reward values. As there exists only one initial PROTAC for each pair of warhead and E3 ligand, the top 3 PROTACs generated by our model ranked by reward

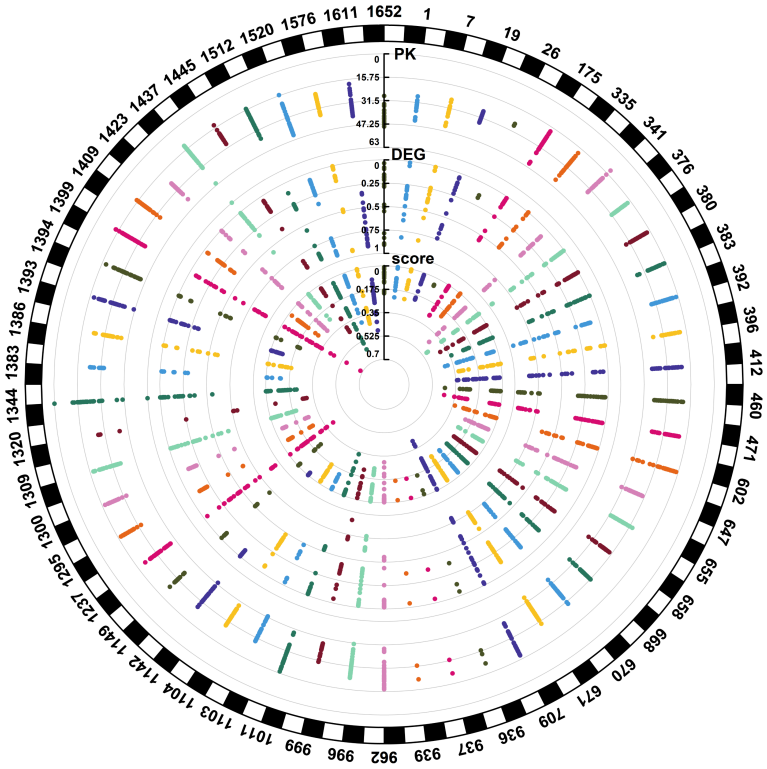


Fig. 4. The three scores, namely reward score, DEG score, and PK score, are visually represented on three distinct tracks in the Manhattan plot

values were averaged and contrasted with the original PROTACs. To present the comparison more effectively, a dumbbell plot using the ggplot2 package [51] in R was illustrated as depicted in Figure. 5. In the figure, blue indicates the original PROTACs, while green represents the PROTACs generated by our model. It is visually evident from the figure that the DEG values of most PROTACs generated by our model are significantly higher than those of the original PROTACs. At the same time, a considerable proportion of PROTACs generated by our model have lower PK values than the original PROTACs. Based on the final reward values, almost all PROTACs generated by our model have higher reward values than the original PROTACs. Our model relies on the reward value to generate PROTACs, and overall, it has successfully optimized the reward value for generating PROTACs. The DEG and PK values of these PROTACs generated by our model are relatively optimal. Therefore, the PROTACs generated by our model provide a comprehensive improvement over the original PROTACs. Detailed information about these scores of original PROTACs is obtained in Supplementary Table S5.

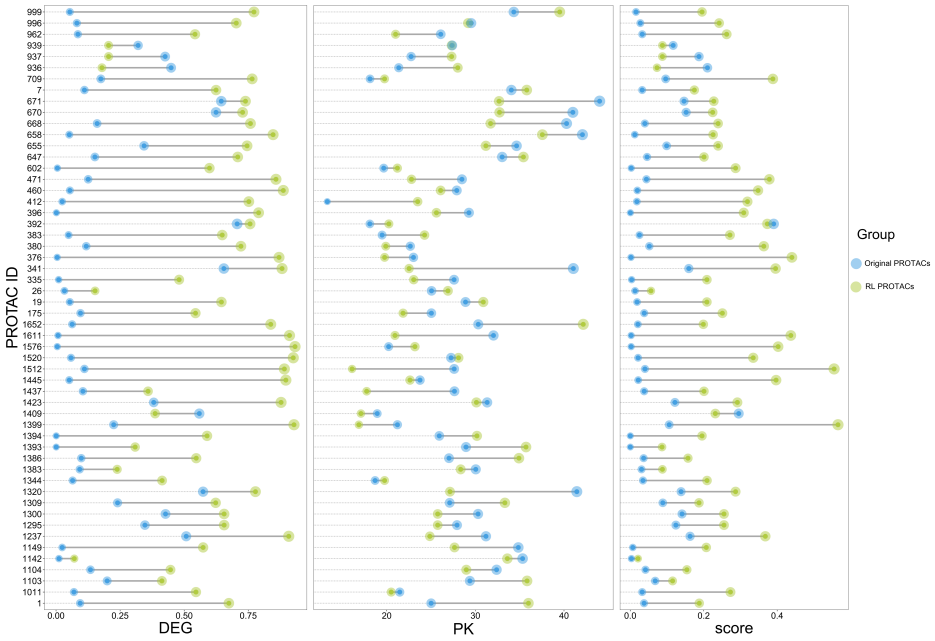


Fig. 5. Comparison between original PROTACs and PROTACs generated by TRL-PROTAC. The blue dots in the figure indicate the original PROTACs, and the green dots indicate the PROTACs generated by our model. The vertical axis indicates the different PROTAC IDs, and the horizontal axis demonstrates the three scores of the PROTACs

5.3. Comparison with other methods

To further demonstrate the superiority of our model (TRL-PROTAC), we used the classic Beam Search algorithm (beam_size=20, n_best=5), Temperature Sampling, the AlphaDrug model and the PROTAC-RL model to generate PROTACs. Here, the top 5 best PROTACs were obtained and compared with the top 5 PROTACs with the best reward scores generated by our model, as shown in Figure. 6. The figure shows that a considerable number of the PROTACs generated by TRL-PROTAC (red) have better PK values compared to those generated by the Beam Search algorithm (blue), temperature sampling (deep blue), the AlphaDrug model (orange) and the PROTAC-RL model(green). Moreover, TRL-PROTAC has a significant advantage in terms of DEG scores and reward scores for almost all PROTACs. Therefore, it can be concluded that the combination of the ProLinker model and reinforcement learning method is quite successful, as it can comprehensively consider PK and DEG scores to optimize the generated PROTACs. Detailed information about these scores of PROTACs is obtained in Supplementary Tables S6, S7 and S8.

To further perform statistical analysis, the scores of different PROTAC IDs for the same model were aggregated and subjected to significance tests (Wilcoxon test), as shown in Figure 7. The results of the significance tests are indicated by statistical significance markers in the figure. A single asterisk "*" denotes a p-value less than 0.05, two aster-

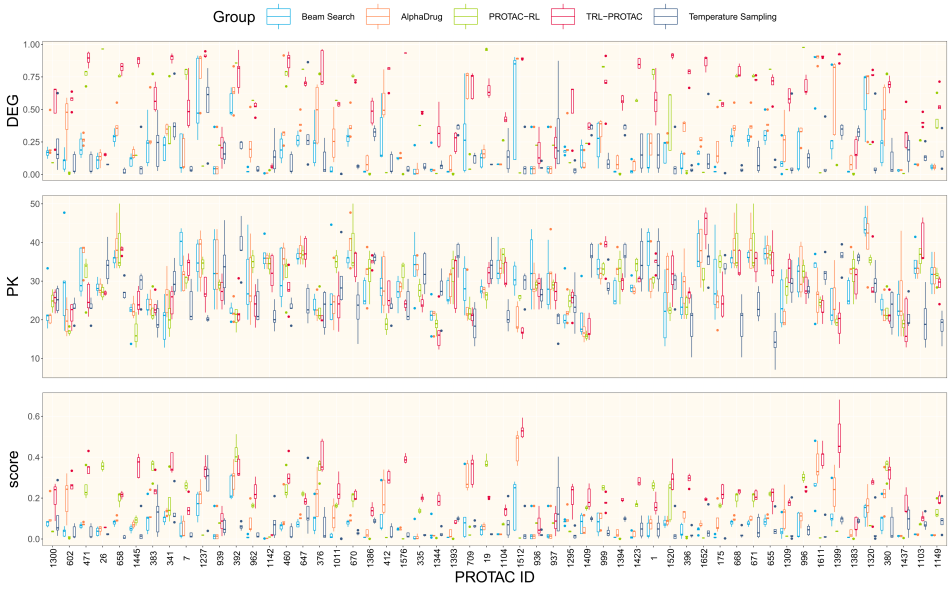


Fig. 6. Comparison between PROTACs generated by Beam Search, Temperature Sampling, AlphaDrug, PROTAC-RL and TRL-PROTAC. The horizontal axis of the figure represents the different PROTAC IDs and the vertical axis represents the three scores of PROTACs. The different coloured boxes indicate the PROTACs generated by different models, where the red boxes are the scores of the PROTACs generated by our model

isks "**" indicate a p-value less than 0.01, three asterisks "***" signify a p-value less than 0.001, four asterisks "****" indicate a p-value less than 0.0001, and "ns" represents a p-value greater than 0.05. From the figure, it can be observed that DEG scores of TRL-PROTAC, in the significance tests against other models, are marked with four asterisks, indicating an extremely significant difference between the DEG scores of our model and those of other models. Furthermore, considering the data distribution in the boxplot, TRL-PROTAC achieves higher DEG scores compared to the other models, indicating a significant superiority of TRL-PROTAC in terms of DEG scores. Regarding PK scores, the significance tests reveal that TRL-PROTAC exhibits a significant difference when compared to Beam Search and AlphaDrug, but no significant difference is observed in comparison to PROTAC-RL and Temperature Sampling. Looking at the data distribution in the boxplot, PK scores of TRL-PROTAC are slightly lower than those of Beam Search and AlphaDrug (lower PK scores are better), while they are similar to the distributions of PROTAC-RL and Temperature Sampling. This could be attributed to the PROTAC-RL model's emphasis on optimizing PK scores, while the Temperature Sampling method enhances sample diversity, making it easier to obtain molecules with better PK scores. However, TRL-PROTAC stands out because it can optimize both PK and DEG scores. Therefore, the final reward score is significantly better than that of other models.

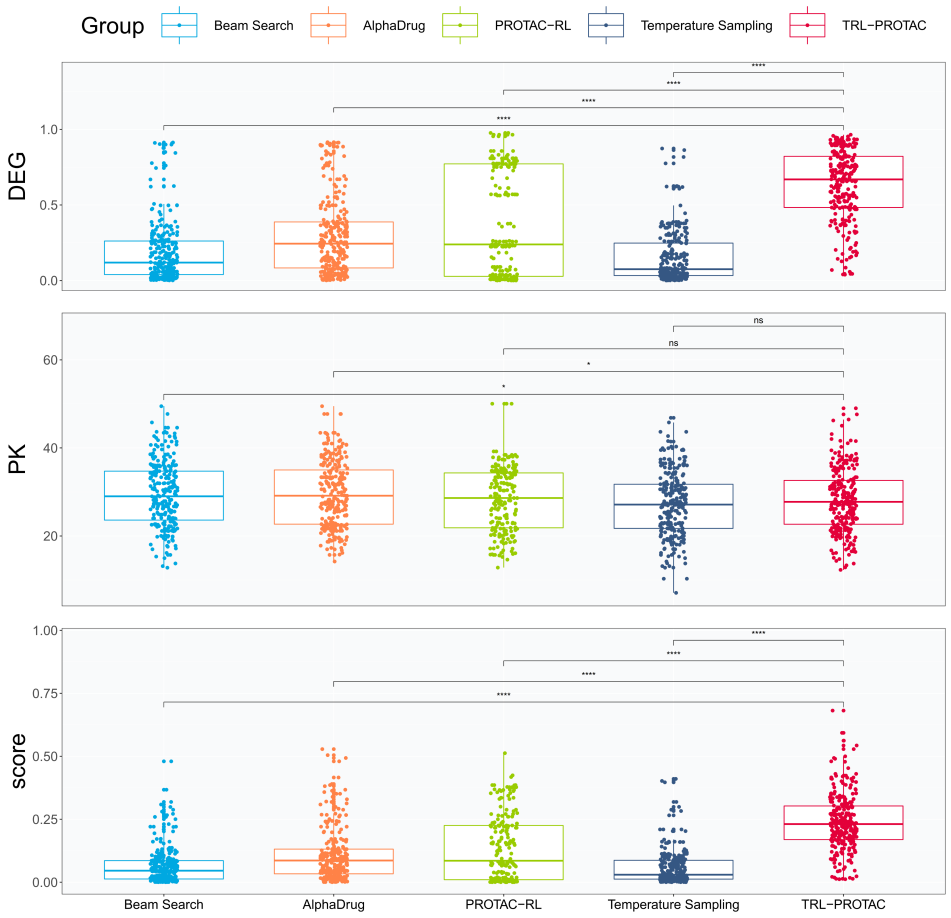


Fig. 7. Comparison between Beam Search, Temperature Sampling, AlphaDrug, PROTAC-RL and TRL-PROTAC after aggregating the scores of different PROTAC IDs. Different coloured boxes indicate PROTACs generated by different models, and the asterisks above the boxes are significance notations for significance tests between the corresponding models

5.4. Comprehensive analysis of the de novo design of PROTACs

As illustrated in Figure. 8a, through the utilization of AlphaDrug, we employed a targeted approach to design warheads and E3 ligands tailored specifically to the protein and E3 ligase, respectively. The resulting ligands demonstrated significantly enhanced binding affinities to their respective proteins. In fact, the majority of the generated ligands exhibited higher binding affinities than their original counterparts, with an impressive average increase of 2.15. Continuing from there, we proceeded to design PROTACs based on the generated ligands. As illustrated in Figure. 8b-c, the resulting PROTACs exhibited notably elevated DEG scores, with the majority exceeding the 0.5 mark. Furthermore, compared

to the MCTS algorithm, our model has demonstrated superiority across all three scores. This outcome underscores the efficacy of our model in optimizing the degradability of the generated PROTACs—an achievement that has proven elusive for other models. However, it is worth noting that these PROTACs displayed relatively high PK scores in both of these methods. This can be attributed to the fact that the ligands devised by AlphaDrug, while exhibiting enhanced binding affinities for the proteins, often showcased intricate structures such as heterocyclic motifs. Consequently, the inherent PK scores of the warheads and E3 ligands were inherently higher, thereby yielding a correspondingly elevated PK score for the generated PROTACs. Detailed information about the binding affinity scores of protein-ligand pairs is obtained in Supplementary Table S9. The scores of de novo designed PROTACs using the TRL-PROTAC model can be found in Supplementary Table S10, while the scores of de novo designed PROTACs using the AlphaDrug model are available in Supplementary Table S11.

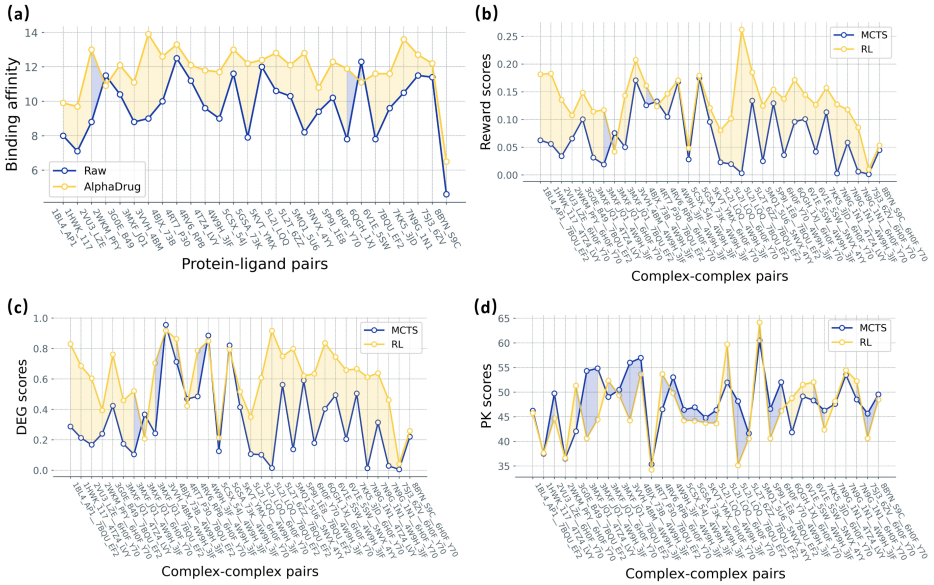


Fig. 8. Results of de novo designs of PROTACs. (a) Comparison between the binding affinity of the generated ligands and the original ligands. (b-d) The distribution of three scores (reward score, DEG score, and PK score) for PROTACs generated by AlphaDrug and TRL-PROTAC

6. Potential applications

In this section, we will discuss the potential applications of the TRL-PROTAC model. Firstly, when designing PROTACs starting from known warheads and E3 ligands, the TRL-PROTAC model demonstrates superior performance compared to other methods

(Beam Search, Temperature Sampling, the AlphaDrug model and the PROTAC-RL model), yielding more favorable PK and DEG properties. Therefore, the TRL-PROTAC model holds significant promise in discovering PROTACs drugs with enhanced attributes. Secondly, in comparison to the original PROTACs, PROTACs generated by the TRL-PROTAC model also exhibit superior characteristics, highlighting the model's potential in optimizing and replacing existing PROTACs medications. Finally, when designing PROTACs targeting specific proteins (considering binding affinity), the TRL-PROTAC model outperforms the MCTS method, showcasing its superiority in molecular de novo design.

7. Conclusion

Previous studies on generating PROTACs really did a good job, but they only considered a single property, such as PK, and then used a classification model to calculate the degradation and filter the generated PROTACs. This approach is both time-consuming and unable to generate valid PROTACs for complex inputs. We have innovatively proposed a de novo design strategy for PROTACs. Our model not only coordinates the PK property and the DEG property of PROTACs during the generation process but also avoids the issue of greedy algorithms not considering long-term benefits. Experimental results show that nearly all of the PROTACs designed using our model have better degradation and most of the molecules have better PK property than the original PROTACs, indicating the feasibility of our model. Compared to Beam Search, Temperature Sampling, the AlphaDrug model and the PROTAC-RL model, our model generates PROTACs with better properties, demonstrating the superiority of TRL-PROTAC in PROTAC generation. Furthermore, the TRL-PROTAC method enables the design of PROTACs targeting specific proteins, offering the advantage of further considering the binding affinity between PROTACs and the target proteins. To sum up, our strategy takes into account the integration of binding affinity, degradation, and pharmacokinetic properties for PROTACs.

Compared to traditional small molecules, the complexity of PROTACs design lies in their ternary structure. Naturally, two approaches can be considered for generating PROTACs. One is to directly generate a complete PROTAC molecule and then divide it into warhead, E3 ligand, and linker parts. Subsequently, the interaction between the warhead and E3 ligand with the respective target proteins is checked. However, the challenge with this approach is how to divide the PROTAC molecule and validate its ability to interact with the target protein after division. The other approach involves designing warheads and E3 ligands separately for the target proteins and then using a deep learning model to generate the linker that connects these two parts and optimizes the entire PROTAC. The difficulty here lies in how to connect these two components effectively. TRL-PROTAC follows the latter approach and provides a novel solution to these challenges, making it a remarkable exploration of PROTAC generation from scratch.

While our approach allows for PROTAC design against target proteins, it still has limitations. It can only be applied to design PROTACs for protein targets with existing ligand interactions and cannot be used for protein targets with no prior ligand interactions. Additionally, the warheads and E3 ligands generated by the AlphaDrug model, while exhibiting a high binding affinity to proteins, contain numerous heterocyclic structures, resulting in elevated PK properties. Further investigations will focus on exploring methodologies to

reduce the structural complexity of the ligands while preserving their heightened binding affinities to target proteins.

References

1. Guenette, R.G., Yang, S.W., Min, J., Pei, B., Potts, P.R.: Target and tissue selectivity of protac degraders. Chemical Society Reviews (2022)
2. Guedeney, N., Cornu, M., Schwalen, F., Kieffer, C., Voisin-Chiret, A.S.: Protac technology: A new drug design for chemical biology with many challenges in drug discovery. Drug Discovery Today 28(1), 103395 (2023)
3. Chen, Y., Tandon, I., Heelan, W., Wang, Y., Tang, W., Hu, Q.: Proteolysis-targeting chimera (protac) delivery system: Advancing protein degraders towards clinical translation. Chemical Society Reviews 51(13), 5330–5350 (2022)
4. Cao, C., He, M., Wang, L., He, Y., Rao, Y.: Chemistries of bifunctional protac degraders. Chemical Society Reviews (2022)
5. Chirnomas, D., Hornberger, K.R., Crews, C.M.: Protein degraders enter the clinic—a new approach to cancer therapy. Nature Reviews Clinical Oncology 20(4), 265–278 (2023)
6. Ocaña, A., Pandiella, A.: Proteolysis targeting chimeras (protacs) in cancer therapy. Journal of Experimental & Clinical Cancer Research 39(1), 189 (2020)
7. Zhou, Q.Q., Xiao, H.T., Yang, F., Wang, Y.D., Li, P., Zheng, Z.G.: Advancing targeted protein degradation for metabolic diseases therapy. Pharmacological Research p. 106627 (2022)
8. Li, F., Hu, Q., Zhang, X., Sun, R., Liu, Z., Wu, S., Tian, S., Ma, X., Dai, Z., Yang, X., et al.: Deepprotacs is a deep learning-based targeted degradation predictor for protacs. Nature Communications 13(1), 7133 (2022)
9. Yang, J., Li, Z., Wu, W.K.K., Yu, S., Xu, Z., Chu, Q., Zhang, Q.: Deep learning identifies explainable reasoning paths of mechanism of action for drug repurposing from multilayer biological network. Briefings in Bioinformatics 23(6), bbac469 (2022)
10. Baptista, D., Ferreira, P.G., Rocha, M.: Deep learning for drug response prediction in cancer. Briefings in bioinformatics 22(1), 360–379 (2021)
11. Krentzel, D., Shorte, S.L., Zimmer, C.: Deep learning in image-based phenotypic drug discovery. Trends in Cell Biology (2023)
12. Deng, J., Yang, Z., Ojima, I., Samaras, D., Wang, F.: Artificial intelligence in drug discovery: applications and techniques. Briefings in Bioinformatics 23(1) (2022)
13. Shao, D., Dai, Y., Li, N., Cao, X., Zhao, W., Cheng, L., Rong, Z., Huang, L., Wang, Y., Zhao, J.: Artificial intelligence in clinical research of cancers. Briefings in Bioinformatics 23(1), bbab523 (2022)
14. Lv, H., Shi, L., Berkenpas, J.W., Dao, F.Y., Zulfiqar, H., Ding, H., Zhang, Y., Yang, L., Cao, R.: Application of artificial intelligence and machine learning for covid-19 drug discovery and vaccine design. Briefings in Bioinformatics 22(6), bbab320 (2021)
15. Zeng, X., Wang, F., Luo, Y., Kang, S.g., Tang, J., Lightstone, F.C., Fang, E.F., Cornell, W., Nussinov, R., Cheng, F.: Deep generative molecular design reshapes drug discovery. Cell Reports Medicine p. 100794 (2022)
16. Cerchia, C., Lavecchia, A.: New avenues in artificial-intelligence-assisted drug discovery. Drug Discovery Today p. 103516 (2023)
17. Ma, X.H., Jia, J., Zhu, F., Xue, Y., Li, Z.R., Chen, Y.Z.: Comparative analysis of machine learning methods in ligand-based virtual screening of large compound libraries. Combinatorial chemistry & high throughput screening 12(4), 344–357 (2009)
18. Ye, Q., Hsieh, C.Y., Yang, Z., Kang, Y., Chen, J., Cao, D., He, S., Hou, T.: A unified drug–target interaction prediction framework based on knowledge graph and recommendation system. Nature communications 12(1), 6775 (2021)

19. Chen, Z., Zhao, P., Li, F., Leier, A., Marquez-Lago, T.T., Wang, Y., Webb, G.I., Smith, A.I., Daly, R.J., Chou, K.C., et al.: ifeature: a python package and web server for features extraction and selection from protein and peptide sequences. Bioinformatics 34(14), 2499–2502 (2018)
20. Qian, H., Lin, C., Zhao, D., Tu, S., Xu, L.: Alphadrug: protein target specific de novo molecular generation. PNAS Nexus 1(4), pgac227 (2022)
21. Olivecrona, M., Blaschke, T., Engkvist, O., Chen, H.: Molecular de-novo design through deep reinforcement learning. Journal of cheminformatics 9(1), 1–14 (2017)
22. Yi, Z., Li, S., Yu, J., Tan, Y., Wu, Q., Yuan, H., Wang, T.: Drug-drug interaction extraction via recurrent neural network with multiple attention layers. In: Advanced Data Mining and Applications: 13th International Conference, ADMA 2017, Singapore, November 5–6, 2017, Proceedings 13. pp. 554–566. Springer (2017)
23. Ghaffari, A., Abdollahi, H., Khoshayand, M., Bozchalooi, I.S., Dadgar, A., Rafiee-Tehrani, M.: Performance comparison of neural network training algorithms in modeling of bimodal drug delivery. International journal of pharmaceutics 327(1-2), 126–138 (2006)
24. Karim, M.R., Cochez, M., Jares, J.B., Uddin, M., Beyan, O., Decker, S.: Drug-drug interaction prediction based on knowledge graph embeddings and convolutional-lstm network. In: Proceedings of the 10th ACM international conference on bioinformatics, computational biology and health informatics. pp. 113–123 (2019)
25. Huang, K., Xiao, C., Glass, L.M., Sun, J.: Moltrans: molecular interaction transformer for drug–target interaction prediction. Bioinformatics 37(6), 830–836 (2021)
26. Zheng, S., Tan, Y., Wang, Z., Li, C., Zhang, Z., Sang, X., Chen, H., Yang, Y.: Accelerated rational protac design via deep learning and molecular simulations. Nature Machine Intelligence 4(9), 739–748 (2022)
27. Yang, Y., Zheng, S., Su, S., Zhao, C., Xu, J., Chen, H.: Syntalinker: automatic fragment linking with deep conditional transformer neural networks. Chemical science 11(31), 8312–8322 (2020)
28. Tolpin, D., Shimony, S.: Mcts based on simple regret. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 26, pp. 570–576 (2012)
29. Perez, D., Samothrakis, S., Lucas, S.: Knowledge-based fast evolutionary mcts for general video game playing. In: 2014 IEEE Conference on Computational Intelligence and Games. pp. 1–8. IEEE (2014)
30. Rosin, C.D.: Multi-armed bandits with episode context. Annals of Mathematics and Artificial Intelligence 61(3), 203–230 (2011)
31. Grechishnikova, D.: Transformer neural network for protein-specific de novo drug generation as a machine translation problem. Scientific reports 11(1), 321 (2021)
32. Holtzman, A., Buys, J., Du, L., Forbes, M., Choi, Y.: The curious case of neural text degeneration. In: International Conference on Learning Representations (2019)
33. Landrum, G., et al.: Rdkit: A software suite for cheminformatics, computational chemistry, and predictive modeling. Greg Landrum 8 (2013)
34. DeGoey, D.A., Chen, H.J., Cox, P.B., Wendt, M.D.: Beyond the rule of 5: lessons learned from abbvie's drugs and compound collection: miniperspective. Journal of Medicinal Chemistry 61(7), 2636–2651 (2017)
35. Sewak, M., Sewak, M.: Policy-based reinforcement learning approaches: Stochastic policy gradient and the reinforce algorithm. Deep Reinforcement Learning: Frontiers of Artificial Intelligence pp. 127–140 (2019)
36. Zhang, J., Koppel, A., Bedi, A.S., Szepesvari, C., Wang, M.: Variational policy gradient method for reinforcement learning with general utilities. Advances in Neural Information Processing Systems 33, 4572–4583 (2020)
37. Sutton, R.S., Barto, A.G.: Reinforcement learning: An introduction. MIT press (2018)
38. Alegre, L.N., Bazzan, A.L., et al.: Knowledge transfer in multi-objective multi-agent reinforcement learning via generalized policy improvement. Computer Science and Information Systems (00), 71–71 (2023)

39. Fu, Q., Kang, Y., Gao, Z., Wu, H., Hu, F., Chen, J., Zhong, S.: Multi-threading parallel reinforcement learning. International Journal of Computer Applications in Technology 61(4), 278–286 (2019)
40. Weng, G., Shen, C., Cao, D., Gao, J., Dong, X., He, Q., Yang, B., Li, D., Wu, J., Hou, T.: Protac-db: an online database of protacs. Nucleic acids research 49(D1), D1381–D1387 (2021)
41. Gaulton, A., Bellis, L.J., Bento, A.P., Chambers, J., Davies, M., Hersey, A., Light, Y., McGlinchey, S., Michalovich, D., Al-Lazikani, B., et al.: Chembl: a large-scale bioactivity database for drug discovery. Nucleic acids research 40(D1), D1100–D1107 (2012)
42. Dimitropoulos, D., Ionides, J., Henrick, K.: Using msdchem to search the pdb ligand dictionary. Current protocols in bioinformatics 15(1), 14–3 (2006)
43. Golovin, A., Dimitropoulos, D., Oldfield, T., Rachedi, A., Henrick, K.: Msdsite: a database search and retrieval system for the analysis and viewing of bound ligands and active sites. Proteins: Structure, Function, and Bioinformatics 58(1), 190–199 (2005)
44. Golovin, A., Oldfield, T.J., Tate, J.G., Velankar, S., Barton, G.J., Boutselakis, H., Dimitropoulos, D., Fillon, J., Hussain, A., Ionides, J.M., et al.: E-msd: an integrated data resource for bioinformatics. Nucleic Acids Research 32(suppl_1), D211–D216 (2004)
45. Consortium, U.: Uniprot: a hub for protein information. Nucleic acids research 43(D1), D204–D212 (2015)
46. Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H., Shindyalov, I.N., Bourne, P.E.: The protein data bank. Nucleic acids research 28(1), 235–242 (2000)
47. Bakan, A., Meireles, L.M., Bahar, I.: Prody: protein dynamics inferred from theory and experiments. Bioinformatics 27(11), 1575–1577 (2011)
48. O'Boyle, N.M., Banck, M., James, C.A., Morley, C., Vandermeersch, T., Hutchison, G.R.: Open babel: An open chemical toolbox. Journal of cheminformatics 3(1), 1–14 (2011)
49. Yin, L.: Cmplot: circle manhattan plot. r package version 3.6 (2020)
50. Wang, J., Yu, J., Lipka, A.E., Zhang, Z.: Interpretation of manhattan plots and other outputs of genome-wide association studies. In: Genome-Wide Association Studies, pp. 63–80. Springer (2022)
51. Wickham, H., Wickham, H.: Data analysis. ggplot2: elegant graphics for data analysis pp. 189–201 (2016)

Yuhao Dai is a graduate student at School of Computer Science and Technology, Soochow University. He is mainly engaged in deep generative models, bioinformatics and molecular generation.

Fei Zhu is an associate professor at School of Computer Science and Technology, Soochow University. He is a senior member of China Computer Federation. His main research interests include deep learning, reinforcement learning, protein interaction, and pattern recognition.

Received: March 27, 2024; Accepted: June 10, 2024.

Computer Science and Information Systems 21(4):1321–1334 https://doi.org/10.2298/CSIS240122037Z

Psychological Effect Computation of Courtroom Arguments: A Deep Learning Approach of EEG Signal Data

Xuan Zhou [1], Yaming Liu [2], Baoqian Jiao [1, *], Hanzhen Ouyang[3], and Weihui Dai [3, *]

[1] Guanghua Law School, Zhejiang University,
Hangzhou 310008, China
{zhoushelley, jiaobaoqian}@ zju.edu.cn
[2] Department of Electrical and Computer Engineering, University of Toronto,
Toronto M5S3G4, Canada
yaming.liu@mail.utotonto.ca
[3] School of Management, Fudan University,
Shanghai 200433, China
{ouyanghanzhen, whdai}@fudan.edu.cn

Abstract Previous studies have shown that the attorney's speeches can exert significant impacts on the cognition and judgment of the jury in court arguments. However, the psychological effects induced by these speeches are intricately tied to subconscious brain states, making them challenging to accurately and comprehensively describe through subjective self-reports. This study aims to explore a neural reaction observation method for psychological effect analysis of the attorney's speeches in courtroom scenarios. We utilized a corpus of courtroom arguments from legal movies and television series as source material. Participants' psychological responses to these speeches were monitored using wearable electroencephalography (EEG) devices. Building upon this data, we employed a deep learning model based on Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks to compute attention intensity, cognitive load, and emotional changes. Our test results demonstrate that this approach enables continuous and dynamic computation within courtroom argument contexts, providing a more accurate assessment of attorneys' language skills.

Keywords: Courtroom argument, attorney' Speech, psychological effect, EEG, CNN-LSTM.

1. Introduction

Courtroom argument is the vital part of the trial process, its language skills have important impacts on the cognition and adjudication of the trialed case [1-4], and have attracted the common interest and attention as an interdisciplinary research domain of law, linguistics, and psychology [5, 6]. Crucially, the attorneys need to utilize various logic and emotional expression skills according to the designed defense strategies, to guide the cognitions of the trial personnel and persuade them to make the judgmental

* Corresponding authors

decisions in favor of their clients [7-10]. Besides, the criterion as to whether justice is fulfilled had to consider the usual judgements of ordinary people, especially in most western countries which have a jury system. The jury represents the consensus of the society or a local social community, not the thinking of legal professionals. Therefore, the courtroom arguments should be presented and accepted in the light of the cognition and judgment of the public. Conley et al. analyzed the impact of changes in the presentational style of courtroom language on the decision makers [7]. Hahn and Clayton evaluated the effect of attorneys' narrative style and gender on decisions of the jury [8]. They all concluded that the excellent language skills of the attorneys play an important role subtly in striving for acquittals for their clients.

As for the language skills in courtroom arguments, the language act theory and pragmatic studies regard it as an issue of linguistic psychological effects, which are caused by the following features embedded in the argument speeches: legal compliance, logical validity, and psychological effects [9-11]. Hu expounded the legal compliance from five perspectives of jurisprudential norms, discourse power norms, procedural norms, evidentiary norms, and debating norms [9]. Shalmanova and Shumov made a further study on the precision problem of court language [10]. Based on the corpus analysis of approximately 1000 sentences extracted from the case records of the US Supreme Court, Liu investigated the lawyers' arguments from the perspective of linguistic communication dynamics, and revealed the language logic and the characteristics of the emotional expression [11]. Yu analyzed the factors affecting the acceptance of courtroom arguments from the viewpoint of legal thinking and logical methodology, and emphasized the roles of situational cognition and persuasive utility under the above specific contexts [12]. This utility can be understood as the resulting valence of psychological effects for the trial personnel or the public.

The psychological effects caused by languages usually include cognitive and emotional aspects dominated by human brain mechanism [13-16], which have mutual influence and involve a series of descriptive indicators. However, this study mainly focuses on the three indicators of attention intensity, cognitive load, and emotional change, which are of great significance to the analysis of language skills under the scenarios of courtroom arguments. Attention intensity usually reflects how much attention is paid to the speeches presented by the arguer, cognitive load represents how easy it is for the listener to understand what is being said by the arguer, and emotional change indicates how emotionally affected the listener is by the arguer's speeches [17, 18]. Nowadays, a lot of computational methods have been presented for psychological effect analysis of human speeches [20-22]. Nevertheless, the induced psychological effects by speeches are closely pertinent to the brain's subconscious perception, which are difficult to be described and expressed in subjective narration at the state of consciousness [25, 26]. In recent years, the development of experimental observation techniques in neuroscience such as electroencephalography (EEG) has made it possible to observe the brain neural activity in the subconscious or even unconscious states, thereby EEG data has been applied to emotion recognition and affective computation for achieving more accurate results [25-28]. Different from the above existing researches, this study aims to explore the method for continuous and dynamic psychological computation of attention intensity, cognitive load, and emotional changes induced by the attorney's speeches based on EEG signal data. The above psychological effects and neural mechanism are also the great concerned issue in the feild of Neurolaw [29].

2. Data Acquisition and Preprocessing

In the experiment, a total of 22 qualified subjects (10 males and 12 females) aged between 18 and 55 were selected as members of a mock jury or public representatives. Among them, 12 were undergraduate and graduate students while the remaining 10 were teachers and employees of government agencies and companies. Further, three had a background in legal profession, 8 in humanities and social sciences, 6 in engineering and technology, and 5 in mathematics, physics, and medical science. Prior to the experiment, each subject was fully informed on the experimental purpose, process, and potential impacts on human body in accordance with the ethical norms, and signed an informed consent form.

In order to avoid the influence of the subjects' preconceptions about the case and characters, the courtroom argument scenarios extracted from the famous movie "And Justice for All" and the television series "American Crime Story: The People v. O.J. Simpson" were used as the test materials in the study, as shown in Fig. 1.



Fig. 1. The courtroom argument scenarios extracted from the movie "And Justice for All" (left), and the television series "American Crime Story: The People v. O.J. Simpson" (right)

The experiment used a specifically designed computer control system for EEG testing, which can collect the data from EEG device according to the following experimental procedure and paradigm.

Stage 1: Preparation for testing. Put on the electrode cap for the subject and debug the EEG device; prepare the subjects in the resting and calm state while the control computer sends the instruction to begin the test.

Stage 2: Resting-state data acquisition. Start the EEG to record the resting-state brain electrical signals in the first 30s, and then to transmit the data to the control computer in which a special software reads and stores them in the database.

Stage 3: Synchronous data acquisition. The control computer issues the EEG testing instructions, which is then conditioned by the synchronous measurement and control unit. The output signals are sent out and the variation of the EEG signal is recorded and transmitted to the control computer for storage. In the meanwhile, the psychological effects of the attorneys' speeches are evaluated by the subjects with a group of rotary mechanical levers. Based on the position changes, the levers can convert the psychological effects into the score values between intervals of attention intensity [0,5], cognitive load [0,5], and emotional change [-5,5], and the results are stored in the database.

Stage 4: Test completion. The control computer sends out instructions to complete the test and to restore the device to its initial state.

The study adopted the international standard of 10-20 double-lead 8-channel EEG signal acquisition, in which the electrode distribution of the 8 channels was as follows: C1: Fp1-T3, C2: Fp2-T4, C3: T3-O1, C4: T4-O2, C5: Fp1-C3, C6: Fp2-C4, C7: C3-O1, C8: C4-O2. During the playback of the test materials, we collected 18 paragraphs of EEG signals with total 26 minutes 20 seconds containing only the attorneys' speeches through computer synchronous control. For example, the structure of EEG raw data for one scenario of the attorneys' speeches shows as in Table 1.

Table 1. The structure of EEG raw data for one scenario of the attorneys' speeches

scenario	mi	se	c1	c2	c3	c4	c5	c6	c7	c8	bs
	1.391	83.438	-19.386	-31.445	-17.582	-31.784	-14.201	-26.374	-11.496	-26.036	-738.123
	1.391	83.445	-12.172	-24.345	5.748	5.748	10.144	5.748	9.467	8.791	-738.123
	1.391	83.453	-13.863	1.014	-16.230	-10.144	-7.777	-5.410	-24.683	-.338	-738.123
	1.391	83.461	-32.347	-14.201	-25.359	1.014	-18.597	-1.352	-.676	5.072	-738.123
	1.391	83.469	-24.007	-7.439	-.338	10.482	-4.396	7.101	6.762	27.050	-738.123
	1.391	83.477	-23.105	-1.014	-11.496	-3.719	-13.525	-4.734	-2.029	-2.367	-738.123
	1.391	83.484	-26.712	-15.216	-6.086	-.3380	-9.129	-2.029	.338	-7.777	-738.123
	1.392	83.492	-24.007	.338	-10.82	-12.172	-13.863	-4.057	-12.511	1.352	-738.123
	…	…	…	…	…	…	…	…	…	…	…

The EGG signals are collected once per 8ms and therefore composing a time series. In the data processing, the study first filters out the EEG data before and after 2s of the changing process of each lever, because those data may be affected by manual operations and could not accurately reflect the psychological changes of the subjects. In addition, a cubic spline interpolation method is used to interpolate the score values of psychological effects to form the time series synchronously with EEG data for machine learning.

The EEG signals can be processed by means of wavelet decomposition and reconstruction to yield typical brain wave frequencies such as δ (0.5-3Hz), θ (4-7Hz), α (8-13Hz), β (14-30Hz), and γ (>30Hz). Among them, the θ and α waves are mostly linked to the brain's subconscious and unconscious states. In order to get more accurate computation results by machine learning, the study divides the signals into the following nine subdivided bands: δ (0.5-3Hz), θ (4-7Hz), low α (8-10Hz), high α (11-13Hz), low β (14-15Hz), midrange β (16-20Hz), high β (21-30Hz), low γ (31-40Hz), and midrange γ (41-50Hz) , and pre-processes the EEG signals by filtering and wavelet transform to obtain the values of each subdivided band for the subsequent machine learning.

3. Machine Learning Method

3.1. CNN-LSTM Model

The CNN-LSTM model combines with the advantages of LSTM (Long Short-Term Memory) model and CNN (Convolutional Neural Networks) model, which has been successfully applied to the emotion recognition from EEG signals in recent years [30]. Therefore, we design the CNN-LSTM model architecture and employ it for the

correlation assessment between EGG signals and indicators of attention intensity, cognitive load, and emotional changes. We mainly focus on the use of LSTM since it has several advantages on dealing with the problem in the study. Firstly, LSTM has a chain-like structure which makes it suitable for time series analysis. Secondly, as a non-linear model, LSTM can be used as a complex non-linear unit for constructing larger deep neural networks. Thirdly, LSTM can remember information for extended periods of time, helping it to handle long-term dependencies.

In the proposed CNN-LSTM model architecture, CNN is used for further feature extraction, and LSTM is applied to exploit temporal relationships in the data. These two sections are linked together in order to perform higher accuracy. Specifically, 1D Convolutional Neural Network is adopted as a feature extractor and data encoder, providing more features for the LSTM to train and calculate. Compared with traditional fully-connected neural networks, 1D-CNNs can better handle local relationships in sequential data, and therefore perform better in tasks such as speech recognition, natural language processing, and time series prediction. It can automatically extract important features from the data, thus reducing the workload of manual feature extraction, while providing better generalization performance.

A model based on LSTM is designed to control the accumulation rate of information by introducing gating mechanisms as shown in Fig. 2, including selectively adding new information and selectively forgetting previously accumulated information, thereby improving the long-range dependency problem of recurrent neural networks (RNNs) and alleviating the gradient disappearance problem during the training of long sequences.

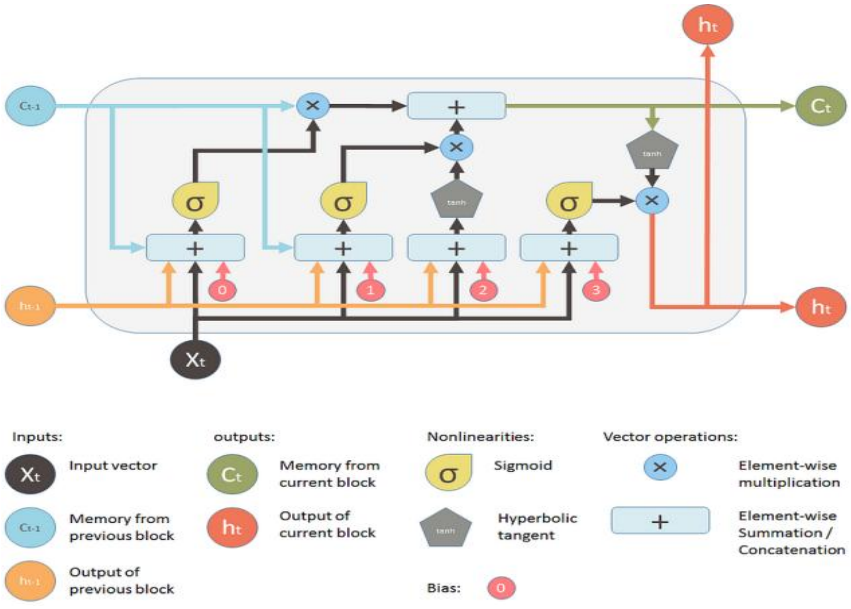


Fig. 2. LSTM built block

The first step of the LSTM is to decide what information to discard or keep. This decision is made by a "forget gate" which selectively forgets by sigmoid function.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{1}$$

The second step which consists two parts are taken to update the cells status. The "input gate" uses sigmoid to determine which values to update and tanh to generate new candidates. The sigmoid output is then multiplied by the tanh output candidates, i.e., the sigmoid output determines which of the tanh output values are important to keep.

$$i_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_i) \tag{2}$$

$$\widetilde{C_t} = tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{3}$$

Finally, the output values are calculated by output cell state. The output is first obtained through the sigmoid layer, and then the new memory cell state obtained in the previous stage is deflated using tanh and then multiplied pair by pair with the output obtained from sigmoid to determine the information which the hidden state should carry. The new hidden state is used as the current output, thereby, the new information is passed to the next time step.

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C_t} \tag{4}$$

The model consists two CNN layers, a one-layer LSTM network and a full connected layer as shown in Fig. 3. A dam optimizer together with a learning rate of 0.001 which would be divided by 2 in every 15 epochs is applied. The architecture is programmed in Pytorch. All the figures are trained in a batch size of 2.

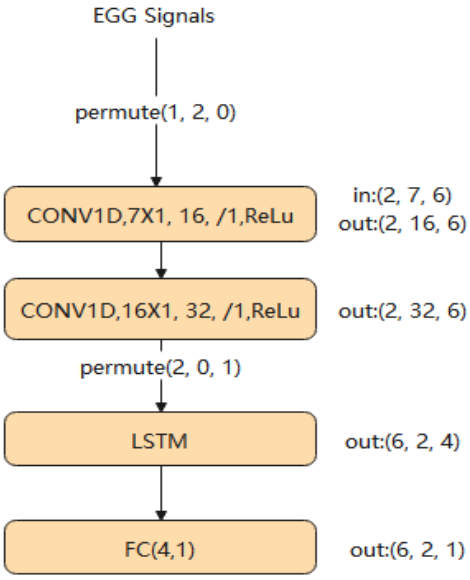


Fig. 3. Model architecture

3.2. Training and Evaluation

The model for three different output features, namely attention intensity, cognitive load, and emotional change, are trained respectively. The model architecture and hyperparameters for these features are the same. Because the induced psychological effects by speeches are closely related to the perception at the brain's subconscious states, which can't be described accurately and comprehensively through a subjective self-report way. However, the psychological effects under brain's subconscious states and conscious states have relationships in different band waves of EEG signals for the same subject [24], and existing studies indicated that PSD (Power Spectral Density) parameters of the EEG signals have good performance in emotion recognition [24, 27, 28]. Therefore, we use the following adjusted formula to calculate the full psychological effects, and take which as the target values for machine learning.

$$P(i) = P_e(i)(1 + PSD_m/PSD_e), i = 1,2,3 \tag{5}$$

where, P_e(i) stands for the explicit part of psychological effects that can be evaluated by the subjects under conscious states in Step 3 of the experimental procedure, while PSD_m and PSD_e are the power spectral densities of band 0.5-7 Hz (δ and θ waves) and band 8-50 Hz (α, β, low γ, and midrange γ waves) respectively for the same subject. Using the adjusted target values for machine learning can better consider the subconscious psychological effects [24].

The study first numbers the data sequentially, and then randomly divides these data into three parts, namely learning data set, validation data set and test data set, with a proportion of 8: 1: 1, and RobustScaler is used for the normalization of the input data so as to reduce the impact of outliers. The training process of the model architecture is as follows. First, EEG Signals are permuted to the size of (2, 7, 6) and sent into a 2-layer CONV1D network to extract features from the raw signals. The first CONV1D layer has 16 1x1 kernels and the stride of convolution kernels is 1. The second one has 32 kernels, with the same size and stride. Each CONV1D layer is followed with a ReLU unit which can introduce non-linearity to the proposed model. The output features, after passing through the CONV1D layers, are then fed into the LSTM layers, which have the capability to circumvent the long-term dependency problem inherent in standard RNNs. Upon passing through the LSTM layers, the output features are subsequently directed into a FC layer. Ultimately, a softmax output layer is appended to the proposed model to facilitate final recognition.

The learning performances are evaluated by a MAPE (Mean Absolute Percentage Error) curve. Equation (6) shows the calculation of MAPE, where, n is the total number of predict values, A_t is the actual value at time t and C_t is the calculated value.

$$\text{MAPE} = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{A_t - C_t}{A_t} \times 100\%\right| \tag{6}$$

The CNN-LSTM model performances of attention intensity, cognitive load, and emotional change are shown in Fig. 4, Fig. 5, and Fig. 6 respectively.

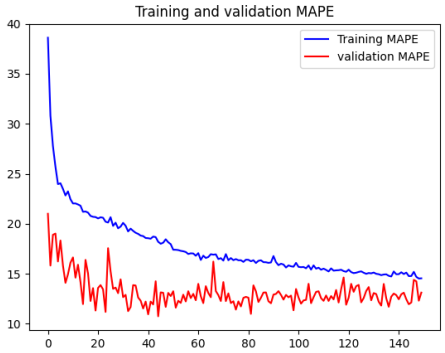


Fig. 4. MAPE curve of attention intensity

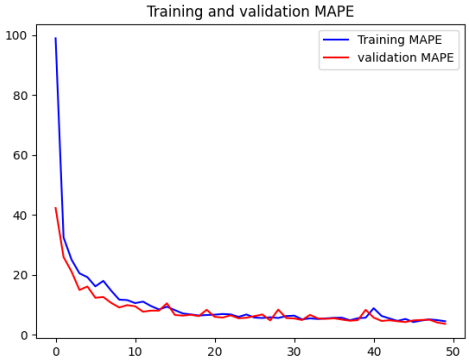


Fig. 5. MAPE curve of cognitive load

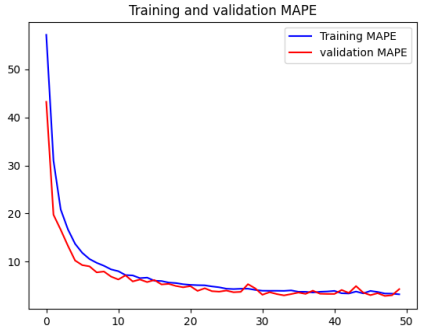


Fig. 6. MAPE curve of emotional change

The final trained MAPEs of CNN-LSTM model, CNN model, and LSTM model are shown contrastively in Table 2.

Table 2. Final trained MAPEs of different models

Model Name	MAPE		
	Attention intensity	Cognitive load	Emotional changes
CNN-LSTM	7.802%	3.188%	14.544%
CNN	22.672%	9.387%	41.053%
LSTM	9.411%	4.123%	20.134%

As seen from Table 2, the CNN-LSTM model can achieve more higher accuracy than a single CNN model or LSTM model, which provides an appropriate and valid approach for the continuous and dynamic computation of psychological effects from EEG data.

4. Psychological Effect Analysis

The US courts deal with cases mostly in adversarial proceedings, for which the main idea is that the court relies on the evidence presented by the litigation parties to develop the judgement. The main speech roles in the courtroom include the judge, the prosecutor, the attorneys, the witnesses, and the defendant. In criminal cases, the prosecutor and the attorneys participate in the proceedings as agents for the state and the defendant, respectively. In the common law jury trial, the jurors act as the fact finders and decide whether the defendant is guilty or not. If found guilty, further sentencing is to be made by the judge who is required to assume the role of presiding over the entire case.

In order to illustrate how the attorneys' linguistic skills would affect the psychology of the jury, this study takes the famous Simpson case as example. Two audio clips are extracted from the attorneys' speeches in the television series of "American Crime Story: The people v. O.J. Simpson", and are used as the source material for analysis. The psychological effects of attention intensity, cognitive load, and emotion changes are computed by the trained CNN-LSTM model. In order to facilitate comparison, the average values of all subjects are uniformly converted into the range between interval [-1,1], as shown in Fig. 7 and Fig. 8.

Segment 1: The interrogation conversation between the defense attorney F. Lee Bailey and the prosecuting witness Detective Mark Fuhrman regarding whether he had utilized the term "nigger."

The attorney's speeches: I will ask a different question. In describing people, Detective Fuhrman, do you use the word "Nigger"? Detective Fuhrman, have you ever used the word "nigger" when addressing someone? // Have you ever used the word "Nigger" in the past ten years? // You mean if you called someone a nigger, you have forgotten it? // Let me put it simply. Are you saying, under oath, that you have not addressed any black person as a nigger or spoken about black people as niggers in the past ten years, Detective Fuhrman? // So, then anyone who comes to this court and quotes you as using that word in dealing with African-Americans would be a liar. Would they not, Detective Fuhrman? // All of them, correct? // Thank you, no further questions.

Psychological effects: As shown in Fig. 7.

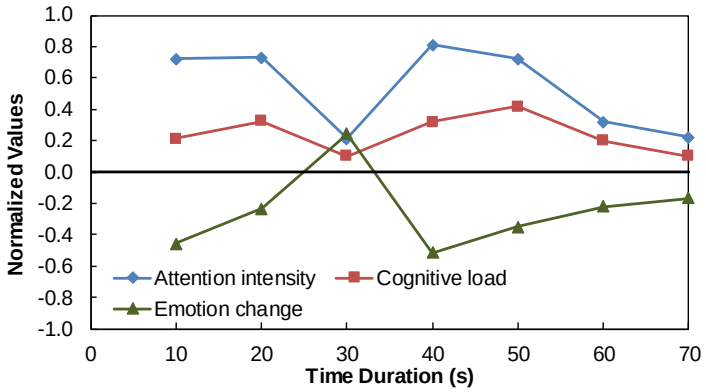


Fig. 7. Psychological effects of the attorney's speeches in Segment 1

Language skills: The attorney threw out four consecutive rhetorical questions to induce the witness to make an affirmative and clear statement that he never used the word "nigger". The purpose of these questions was not to create confusion, but to set the stage for subsequent testimony that Fuhrman frequently used insulting words such as "nigger," thereby arousing the jury's or the public's distrust on him and rendering his testimony ineffective. As shown in Fig. 7, the attention density of the mock jury (or the public members) was at a high level, the cognitive load was moderate, and the emotion change varied in the negative region, indicating that the attorney's questions triggered a high degree of attention and formed negative perceptions with relative ease. In short, his linguistic skills produced highly desirable results.

Segment 2: The defense attorney Barry Scheck questioned the criminal examination expert Dennis Fung on the key evidence of blood and hair.

The Attorney's Speeches: Mr. Fung, you received from Detective Lange, the blood sample taken from Mr. Simpson, correct? // And where did you take possession of that blood sample? // Well, maybe I can help. The blood was handed to you at the Rockingham scene while you were examining that scene, correct? // So, Mr. Simpson's blood was literally handed to you by LAPD at the very location where you found evidence of his blood on the carpet in the driveway, in his socks? // This might explain a huge unanswered question in this case. Are you aware, sir, that 1.9 milliliters or one quarter of the blood collected from Mr. Simpson is missing?! // Mr. Fung, when did you realize that the blanket covering Nicole Brown's body was actually from inside her own residence? // Would you agree, sir, that a blanket taken from inside her residence placed by law enforcement over her dead body could be thought of as a contamination of that scene? // And that if Mr. Simpson had been in that home previously, sitting or lying on that blanket, his hairs could be on that blanket? Would be on that blanket? And thus, would have been in the crime scene? // That is a terrible mistake for a criminalist to make. Isn't it? // Mr. Fung, have you made some bad choices in this case? // No further questions.

Psychological effects: As shown in Fig. 8.

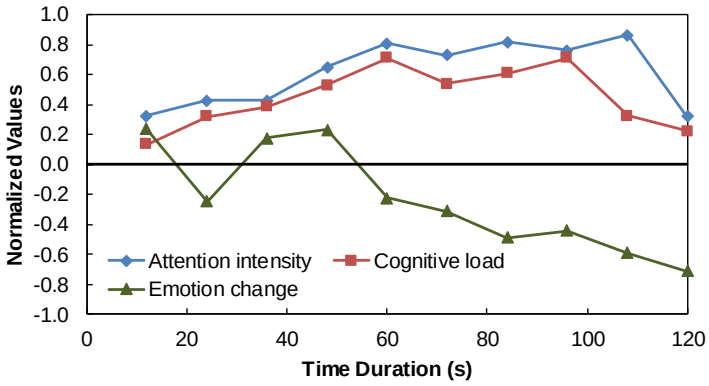


Fig. 8. Psychological effects of the attorney's speeches in Segment 2

Language skills: The attorney used a barrage of questions that caused the witness to gradually begin to doubt the legitimacy of the evidence. Stammering and ambiguous in his answers, the expert witness not only was interrupted several times by the attorney, but also agreed that "that is a terrible mistake for a criminalist to make." Using quick and frequent questioning, the attorney effectively consolidated his dominant voice and controlled the direction of the whole conversation, making the jury (or the public) naturally doubt the credibility of the procurators' key evidences. According to Figure 8, the attention density of the mock jury (or the public members) was always at a high level, indicating that the attorney's questions had triggered a high level of attention. The cognitive load level was also high, suggesting that the questions had prompted a deep thinking. The overall rapid reduction of emotion changes into the negative region reflected that the jurors (or the public) had developed strong doubts and dissatisfaction. In short, the attorney's language skills were again highly successful in producing the desired effects.

5. Summary and Discussion

The study of courtroom arguments is becoming a hot topic in the multidisciplinary fields. It is of important significance for the in-depth analysis of the jurors' (or the publics') cognitions and affecting factors, and also for improving the language skills of the litigation parties and maintaining the impartiality in legal adjudication for the society.

Especially, the attorney's speeches in court arguments can produce subtle cognitive and emotional effects on the jurors, and significantly affect their judgment. Those psychological effects are closely related to the subconscious perception of human brains, and difficult to be reflected in a subjective self-report way. This study explored the psychological computation method based on EEG signal data for coping with the problem. A valid deep learning approach with CNN-LSTM model architecture was presented for evaluating the full psychological effects induced by attorney's speeches, in terms of attention density, cognitive load, and emotion changes.

In real trials, the courtroom arguments are conducted in the form of dialogues between two or more parties with complex contexts and involving multi-faceted expertise. Therefore, further exploration is needed in the accurate extraction of EEG signals and on the cognitive discrepancies between the jury or the public members with different genders and professional backgrounds. In addition, the neural mechanisms underlying these psychological effects have yet to be systematically investigated in conjunction with various neuro-experimental equipment such as functional magnetic resonance imaging (fMRI) and event-related potentials (ERPs). Besides, the main purpose of this study is to explore the valid approach to compute the above psychological effects, and the complicated factors causing those effects are not thoroughly considered.

Acknowledgements. This research has been supported in part by National Natural Science Foundation of China (No.71971066), National Key Research and Development Program of China (No. 2023YFC3304602), and Major Project of Philosophy and Social Science Research, Ministry of Education of China (No.19JZD010). Weihui Dai and Baoqian Jiao are the joint corresponding authors of this paper.

References

1. O' Barr, W. M.: Linguistic Evidence: Language, Power and Strategy in the Courtroom. New York, USA: Academic Press. (1982)
2. Conley, J. M., O'Barr, W. M.: Just Words: Law, Language, and Power; London, England: The University of Chicago Press. (1998)
3. Nuez, N., Schweitzer, K., Chai, C. A., et al.: Negative emotions felt during trial: the effect of fear, anger, and sadness on juror decision making. Applied Cognitive Psychology, Vol.29, No.2, 200-209. (2015)
4. Estrada-Reynolds, V., Schweitzer, K., Nunez, N.: Emotions in the Courtroom: How Sadness, Fear, Anger, and Disgust Affect Jurors' Decisions. Wyoming Law Review, Vol.16, No.2, Article 5. (2016)
5. Correa, M.: Forensic linguistics: an overview of the intersection and interaction of language and law. Studies About Languages, Vol.2013, No.23, 5-13. (2013)
6. Monica, K. M., Logan, A. Y., Matthew, T. H., Jason A. C.: The Cambridge Handbook of Psychology and Legal Decision-Making. Cambridge, England: Cambridge University Press. (2024)
7. Conley, J. M., O'Barr, W. M., Lind, W. M.: The power of language: presentational style in the courtroom. Duke Law Journal, Vol.27, No.6, 1375-1399. (1978)
8. Hahn, P. W., Clayton, S. D.: The effects of attorney presentation style, attorney gender, and juror gender on juror decisions. Law and Human Behavior, Vol.20, No.5, 533-554. (1996)
9. Hu, S. W. A.: Research on court language based on speech act theory. Journal of Guangdong Communications Polytechnic, Vol.12, No.3, 125-128. (2013)
10. Shalmanova, E., Shumov, P.: The problem of court language precision as a quality criterion of the judicial act. Bulletin of Science and Practice, Vol.5, No.6, 459-463. (2019)
11. Liu, L. The Analysis of Linguistic Features Between English Law Text and Counsel's Statement of Defense Based on the Thematic Theory. Tianjin, China: Tianjin University of Technology. (2008)
12. Yu, F.: Discussion on Logic and Methodology in Argumentation Acceptance in Court. Wuhan, China: Central China normal University. (2019)
13. Frackowiak, R. S. J., Friston, K. J., Frith, C. D., et al.: Human Brain Function: Second Edition. New York, USA: Academic Press. (2004)

14. Huang, S., Zhou, X., Xue, K., Wan, X. Q., Yang, Z. Y., Xu, D. R., Ivanovic, M., Yu, X.: Neural cognition and affective computing on cyber language. Computational Intelligence and Neuroscience, Vol.2015, Article ID 749326, 1-10. (2015)
15. Zhang, Q. Y., Lu, J. M.: The concept and mechanism of emotional contagion. Advances in Psychological Science, Vol.21, No.9, 1596-1604. (2013)
16. Wang, Y. H., Hu, X. H., Dai, W. H, Zhou, J.; Kuo, T. Z.: Vocal emotion of humanoid robots: a study from brain mechanism. The Scientific World Journal, Vol. 2014, Article ID 216341, 1-7. (2014)
17. Wei, P. J., Fan, X. H.: Research of feature extraction in voice orientation analysis. Application Research of Computers, Vol.31, No.12, 3580-3583. (2014)
18. Sweller, J.: Cognitive load during problem solving: Effects on learning. Cognitive science, Vol.12, No.2, 257-285. (1988)
19. Horlings, R.; Rothkrantz, L. Emotion recognition using brain activity. Proc. Int. Conf. Comput. Syst. Technol 2008, 1-6.
20. Li, H. F., Chen, J., Ma, L., Bo, H., Xu, C., Li, H.: Dimensional speech emotion recognition review. Journal of Software, Vol.31, No.8, 2465−2491. (2020)
21. Lieskovská, E., Jakubec, M., Jarina, R., Chmulík, M.: A review on speech emotion recognition using deep learning and attention mechanism. Electronics, 2021, 10, 1163. https://doi.org/10.3390/electronics10101163. (2021)
22. Mehrish, A., Majumder, N., Bharadwaj, R., et al.: A review of deep learning techniques for speech processing. Information Fusion, 99, 101869, 1-55. (2023)
23. Shaari, A., Syafiq, M. M. J., Mikami, O. ,et al.: Electroencephalography (EEG) application in neuromarketing-exploring the subconscious mind. Journal of Advanced Manufacturing Technology, Special Issue iDECON 2018, 81-92. (2018)
24. Dai, W. H., Liu, L., Li, M. L., et al.: Olfactory psychological computation and olfactory environment for human-machine collaboration. Proceeding of ChineseCSCW 2021, 590-598. (2021)
25. Soroush, M. Z., Maghooli, K., Setarehdan, S. K., et al.: A review on EEG signals based emotion recognition. International Clinical Neuroscience Journal, Vol.4, No.4, 118-129. (2017)
26. Chaudhary, R., Jaswal, R. A. A review of emotion recognition based on EEG using DEAP dataset. International Journal of Scientific Research in Science Engineering and Technology, Vol.8, No.3, 298-303. (2021)
27. Li, W., Zhang, Z., Hou, B., et al.: A novel spatio-temporal field for emotion recognition based on EEG signals. IEEE sensors journal, Vol.21, No.23, 26941- 26950. (2021).
28. Liang, Z., Fang, F., Wei, W., et al.: Emotion recognition based on Granger causality feature between EEG signals. Journal of Electronic Measurement and Instrumentation, Vol.32, No.6, 87-95. (2018)
29. Spranger, T. M.: International Neurolaw: A Comparative Analysis. Springer: Dordrecht, Heidelberg, London, New York. (2012)
30. Iyer, A., Das, S. S., Teotia, R., et al.: CNN and LSTM based ensemble learning for human emotion recognition using EEG recordings. Multimedia Tools and Applications, Vol.82, No.4, 4883-4896. (2023)

Xuan Zhou is currently a Ph.D. candidate at Guanghua Law School, Zhejiang University, China. She is also a qualified lawyer and a psychological counselor, and serves as a member of the Neuroeconomic Management Committee of China Society of Technical Economics. Her research interests include Neurolaw, Linguistic Psychology, and Arts of Movie and Television.

Yaming Liu is currently a master degree candidate of Electrical and Computer Engineering, University of Toronto, Canada. He received his Bachelor's Degree in

Mechatronic Engineering from South China University of Technology, China in 2024. His research interests include machine learning, and Human-machine Hybrid Intelligence.

Baoqian Jiao is currently a professor at Guanghua Law School, Zhejiang University, China. He received his doctor's degree in Legal Theory from Shandong University, China in 2005. His research interests include Legal Method, Judicial Theory, and Legal Rhetoric.

Hanzhen Ouyang is currently a post-doctoral researcher at School of Management, Fudan University, China. She received her Ph.D. in Business Administration from Wonkwang University, Korea in 2020. Her research interests include Technological Innovation, Behavioral Analysis, and Artificial Intelligence Application.

Weihui Dai is currently a professor at School of Management, Fudan University, China. He received his Ph.D. in Biomedical Engineering from Zhejiang University, China in 1996. He serves as the chair of Neuromanagement and Neuroengineering Branch, Society of Management Science and Engineering of China, and an executive member of Collaborative Computing Professional Committee, China Computer Federation (CCF). His research interests include Neuroartificial Intelligence, Neuro-management, and Human-machine Hybrid Intelligence.

Received: January 22, 2024; Accepted: June 22, 2024.

Computer Science and Information Systems 21(4):1335–1357 https://doi.org/10.2298/CSIS231129035C

A Method for Solving Reconfiguration Blueprints Based on Multi-Agent Reinforcement Learning

Jing Cheng[1,⋆], Wen Tan[1], Guangzhe Lv[2], Guodong Li[3], Wentao Zhang[4], and Zihao Liu[5]

[1] School of Computer Science and Engineering, Xi'an Technological University, Xi'an 710016, China
chengjing@xatu.edu.cn, tanwen@st.xatu.edu.cn
[2] Xi'an Institute of Aeronautical Computing Technology (XICT), Xi'an 710065, China
guangzhe_lv@163.com
[3] Aviation Industry First Aircraft Design and Research Institute, Xi'an 710089, China
guodonglee@163.com
[4] School of Software, Northwestern Polytechnical University, Xi'an 710129, China
wentaoz1223@gmail.com
[5] Leihua Electronic Technology Research Institute of Aviation Industry Corporation of China (AVIC), Wuxi, China
2020264414@mail.nwpu.edu.cn

Abstract. Integrated modular avionics systems primarily achieve system fault tolerance by reconfiguring the system configuration blueprints. In the design of manual reconfiguration, the quality of reconfiguration blueprints is influenced by various unstable factors, leading to a certain degree of uncertainty. The effectiveness of reconfiguration blueprints depends on various factors, including load balancing, the impact of reconfiguration, and the time required for the process. Solving high-quality reconfiguration configuration blueprints can be regarded as a type of multi-objective optimization problem. Traditional algorithms have limitations in solving multi-objective optimization problems. Multi-Agent Reinforcement Learning (MARL) is an important branch in the field of machine learning. It enables the accomplishment of more complex tasks in dynamic real-world scenarios through interaction and decision-making. Combining Multi-Agent Reinforcement Learning algorithms with reconfiguration techniques and utilizing MARL methods to generate blueprints can optimize the quality of blueprints in multiple ways. In this paper, an Improved Value-Decomposition Networks (VDN) based on the average sequential cumulative reward is proposed. By refining the characteristics of the integrated modular avionics system, mathematical models are developed for both the system and the reconfiguration blueprint. The Improved VDN algorithm demonstrates superior convergence characteristics and optimization effects compared with traditional reinforcement learning algorithms such as Q-learning, Deep Q-learning Network (DQN), and VDN. This superiority has been confirmed through experiments involving single and continuous faults.

Keywords: Integrated modular avionics system, Multi-Agent Reinforcement Learning, reconfiguration blueprint, multi-objective optimization problem.

⋆ Corresponding author

1. Introduction

The modular avionics system offers several advantages, including system integration, a layered structure, network unification, scheduling flexibility, and centralized maintenance. It finds widespread applications in aerospace fields, including fighter aircraft, civil aircraft, and satellites [1,2,3]. With the continuous expansion of the functionality of integrated electronic systems, the probability of system failures has also increased dramatically, posing significant challenges to the safety and reliability of the systems. Consequently, fault-tolerant mechanisms for integrated electronic systems [4,5,6] are critical tasks.

Reconfiguration is a crucial mechanism for fault tolerance in integrated electronic systems. Reconfiguration in integrated electronic systems refers to the process of reallocating system resources and loading system software to ensure complete system functionality when faults occur in system partitions or software resources [7]. The reconfiguration of integrated electronic systems is primarily achieved through the generation of reconfiguration blueprints [8,9]. There are two main approaches to generating reconfiguration blueprints: manual design and traditional algorithms [10,11]. Manual design is influenced by practical experience and individual differences, while traditional algorithms emphasize feasibility but have limitations in solving multi-objective optimization problems [12,13,14]. Multi-agent reinforcement learning is a crucial branch of machine learning that can handle more complex tasks in dynamic real-world scenarios [15,16,17]. Generating reconfiguration blueprints using a multi-agent reinforcement learning approach allows for optimizing blueprints from various perspectives. Therefore, this paper proposes an intelligent method for generating reconfiguration blueprints for integrated electronic systems based on multi-agent reinforcement learning.

The main contributions of this paper include:

1) The establishment of mathematical models for integrated electronic systems and reconfiguration blueprints. This involves creating models for system resources, system partition, and system software based on the characteristics of integrated electronic system reconfiguration. Additionally, models for system faults, reconfiguration configuration blueprints, and system migration are developed considering the characteristics of reconfiguration blueprints.

2) The proposal of an Improved VDN algorithm based on average sequential cumulative rewards. Grounded in the VDN algorithm in multi-agent reinforcement learning, this algorithm is designed in conjunction with a comprehensive electronic system. It includes reconfiguration states, actions of agent, overall and local reward functions, and algorithmic strategies. The introduction of an adaptive exploration strategy based on the average sequence cumulative reward enhances the efficiency and accuracy of algorithmic reconfiguration. This builds upon the original exploration strategy of the VDN algorithm. Finally, an analysis and verification of the results of the Improved VDN algorithm are conducted, comparing it with reconfiguration algorithms implemented based on Q-learning [18], DQN [19], and VDN [20], confirming the superiority of the proposed Improved VDN algorithm.

2. Related Works

Reconfiguration techniques play a crucial role in maintaining the security, reliability, survivability, maintainability, and scalability of integrated modular avionics systems, holding substantial practical significance and economic value. Extensive research has been conducted on reconfiguration techniques by professionals both domestically and internationally. Housseyni et al. [15] proposed a distributed reconfiguration method based on a multi-agent model. The method considers the functional module as the smallest unit of reconfiguration, representing each device as an agent. Each agent undergoes reconfiguration at three levels, with the first level responsible for adding and removing functional modules, the second level for integrating functional modules, and the third level for updating functional modules with data. The method ensures the security of distributed reconfiguration through agent communication in eXtensible Markup Language (XML) format. Zhou et al. [21] introduced a functional architecture framework for the Distributed Integrated Modular Avionics (DIMA) platform, utilizing the Architecture Analysis and Design Language (AADL) action model to construct the functional action characteristics of DIMA's dynamic reconfiguration. This framework provides a design basis for implementing DIMA's dynamic refactoring. Saadi A et al. [22] proposed a methodology for validating software refactoring using the Communicating Sequential Processes (CSP) language, refactoring techniques, and the FDR model, aiming to guide the process of software refactoring. Ensuring the consistency of dynamic software refactoring involves conducting a comprehensive analysis before making any modifications to the software. The ability to safely execute software refactoring at runtime is crucial for the feasibility of integrated modular avionics system refactoring. Chen et al. [23] proposed an AADL model based on the reconfiguration process of an integrated modular avionics system. Using the accessibility analysis method of Petri nets, the danger of IMA reconfiguration function is analyzed. This method overcomes the weaknesses of the traditional static analysis method, making it more capable of analyzing hazardous behavior during runtime. It provides a solution to the state explosion problem that arises during reachability analysis.

Literature [24,25] has confirmed the effectiveness of reinforcement learning in optimizing planning. Zhang et al. [26] applied sequential game multi-agent reinforcement learning to the field of integrated modular electronic system reconfiguration. They utilized a policy gradient Monte Carlo search tree algorithm incorporating bias estimation to expedite the convergence of the conventional algorithm. However, the method does not address the adaptation of the exploration factor during the competition and cooperation phases of the agent. Liu et al. [27] proposed a reliability analysis method based on the AADL model, using Petri nets to analyze the reliability of the reconfiguration method. This method assists in designing reconfiguration models at the early stage of integrated modular avionics system development, ultimately improving the reliability of the entire integrated modular avionics system. Hollow et al. [28] proposed a reconfiguration scheme for integrated modular avionics systems based on a simulated annealing algorithm. The scheme finds an equivalent reconfiguration scheme with a search function for the integrated modular avionics system by calculating the fitness. While it has theoretical significance, the method exhibits certain defects, such as the inability to meet timeliness. Cui et al. [10] proposed a reconfiguration technique for integrated modular avionics systems based on distributed techniques. This technique enables the system to adapt to the reconfiguration task, quickly detect and locate faults, and establish a blueprint for reconfiguration. Suo et

al. [12] introduced an integrated modular avionics system reconfiguration safety detection technique based on STPA. The method focuses on coordinating human-computer interactions and detecting system failures based on the severity of the failure, human factors, and time constraints. The analysis results can be utilized for system development, system operation, and project revision, ensuring the safety of reconfigurable integrated modular avionics systems.

3. Research Methodology

3.1. Integrated Modular Avionics System Modeling

The integrated modular avionics system consists of hardware resource modules, operating system modules, and software modules. A single module satisfies functional independence, while multiple modules provide isolation. Additionally, the overall system ensures security and reliability [29,30,31]. The simplified schematic of the integrated modular avionics system is shown in Figure 1.

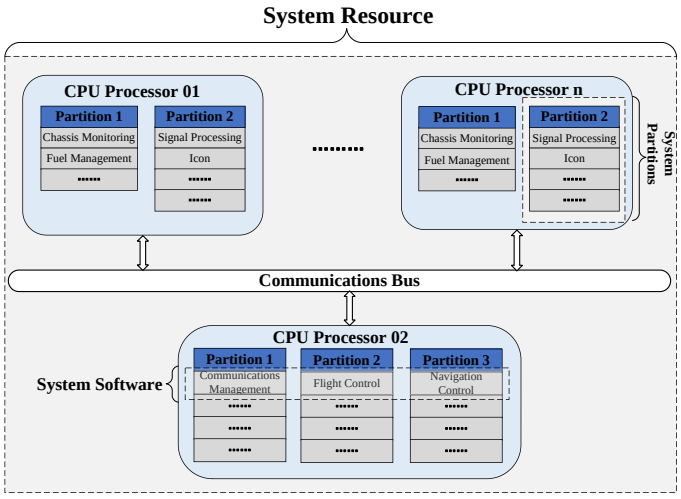


Fig. 1. Schematic diagram of the simplified model of the integrated modular avionics system

System resource model. In this paper, the system hardware resources are abstracted into a system hardware module. This module should include attributes such as CPU resource list, partition resource list, software resource list, and operation status to realize the function of easy maintenance and management of partition collection. The hardware resources

are represented using a quaternion model as shown in Equation (1).

$$\begin{gathered} Resource = \langle CPU_{list}, Patition_{list}, Application_{list}, state \rangle \\ CPU_{list} = \{CPU_1, CPU_2, ..., CPU_S\} \\ Partition_{list} = \{Partition, Partition_2, ..., Partition_M\} \\ Application_{list} = \{App_1, App_2, ..., App_N\} \\ state = \{0, 1\} \end{gathered} \tag{1}$$

Where $Resource$ denotes the system resources. CPU_{list} denotes the set of central processor resources. $Partition_{list}$ denotes the set of partitions configured on the processor. $Application_{list}$ denotes the set of software resources configured on partitions. $state$ denotes the health state of the system resources. 0 is the system resource policy. 1 is the system resource failure. S denotes the total number of system processors. M denotes the total number of system partitions. N denotes the total number of system software applications.

System partition model. The modeling of system partitions is carried out in terms of both temporal and spatial isolation. Combining the principles of temporal scheduling and spatial mapping of partitions, the model for constructing partitions for temporal and spatial attributes is shown in Equation (2) below.

$$\begin{gathered} Partition_i = \langle time_{start}, time_{duration}, \\ mem_{offset}, memsize_{Partition_i}, Application_{list}, state \rangle \end{gathered} \tag{2}$$

Where $time_{start}$ denotes the start execution time of the partition. $time_{duration}$ denotes the duration of execution of the partition. mem_{offset} denotes the offset address of the partition's virtual memory. $memsize_{Partition_i}$ denotes the size of the virtual memory of partition i. $Application_{list}$ denotes the set of applications deployed in the partition. $state$ indicates the health status of the system partition, with 0 being a healthy partition and 1 being a faulty partition.

System software model. The system software model is constructed in terms of both software spatial resources and software temporal resources. The hexadecimal model of software resource modeling is depicted in Equation (3).

$$App_{ij} = \langle pid, rank, run_{cycle}, WCET, time_{deadline}, mem, state \rangle \tag{3}$$

Where App_{ij} denotes the application software model, where the partition number is i and the software number is j. pid denotes the process ID of the application software. $rank$ denotes the task priority of the application software. run_cycle denotes the execution cycle of the application software. $WCET$ denotes the worst-case execution time of the process. $time_deadline$ denotes the application software deadline. mem indicates the space resources required by the application software. $state$ indicates the health status of the system software, with 0 being software health and 1 being software failure.

3.2. Reconfiguration Blueprint Model

System Failure Model. There are two main types of faults in the integrated modular avionics system: processor faults and processor partition faults [32,33]. When any one of these faults occurs, or when multiple faults occur in succession, the integrated modular avionics system enters a fault state [34]. This paper presents a model called the System Fault Abstraction Model (SFAM) for abstracting system faults. To uniquely identify faults and their occurrence locations, two attributes, namely system fault type and system fault location, are required to create a two-tuple model for the system fault model (4).

$$Fault = \langle type, location \rangle \tag{4}$$

Where $Fault$ denotes the system failure model, $type$ denotes the system failure type, and $location$ denotes the system fault location.

It is necessary to establish the following constraints to distinguish between the two types of faults and their respective locations in the integrated modular avionics system.

$$Fault_{ij} = \begin{cases} type =\subseteq \{F_i, F_{ij}\}, (1 \le i \le S, 1 \le j \le M) \\ location_i = C_i, \quad type = F_i \\ location_{ij} = C_i P_j, type = F_{ij} \end{cases} \tag{5}$$

F_i indicates that a processor failure has occurred in the system, and the faulty processor is numbered i. F_{ij} indicates that a processor partition failure has occurred in the system. The faulty partition is located in the jth partition of the CPU number i. C_i indicates the location of a system processor failure with faulty processor number i. $C_i P_j$ indicates the location of a system processor partition failure.

Reconfiguring the configuration blueprint model. The reconfig blue print (RBP) model consists of the following set of five Equations (6) and (7).

$$\begin{gathered} RBP = \langle CPU_{list}, Partition_{list}, Application_{list}, Fault_{list}, Map \rangle \\ Fault_{list} = Fault_1, Fault_2, \ldots, Fault_K \end{gathered} \tag{6}$$

$$\begin{gathered} Map = \left\{ application_i \overset{CPU_s}{\longrightarrow} Partition_j \right\} \\ (i \in [1, |Application_{list}|], j \in [1, |Partition_{lsit}|]) \end{gathered} \tag{7}$$

The reconfiguration blueprint model above illustrates the mapping relationship between processors, partitions, and software in the reconfiguration blueprint. For example: $Map = \left\{ App_1 \overset{C_1}{\longrightarrow} Partition_1 \right\}$ denotes that the application software numbered 1 is deployed on the partition numbered 1. The partition numbered 1 is deployed on the processor numbered 1. $Fault_{list}$ is a list of faults, and the number of faults is K.

Refactoring Migration Blueprint Model. Reconfigure Migrate Blueprint (RMB) consists of a reconfigured initial state, a set of reconfigured intermediate states, a reconfigured end state, and their corresponding transfer relationships. The Reconfigure Migration Blueprint model definition consists of five tuples, as shown in Equation (8) below.

$$\begin{gathered} RMB = \langle S_{init}, S_{end}, Fault_{list}, S_{list}, Move \rangle \\ S_{list} = \{S_1, S_2, ..., S_n\} \\ Move = \left\{ S_{src} \overset{Fault}{\longrightarrow} S_{des} \right\} \end{gathered} \tag{8}$$

Where RMB denotes the refactoring migration blueprint model. S_{init} denotes the refactoring initial state. S_{end} denotes the refactoring end state. S_{list} denotes the set of refactoring intermediate states. $Move$ denotes the refactoring transfer relation. S_{src} denotes the source state in the refactoring transfer relation. S_{des} denotes the destination state of the refactoring transfer relation.

Blueprint Quality. To improve the quality of the filtered feasible reconfiguration blueprints, this paper introduces optimization objectives to assess their quality.

A.Load balancing

Rationalizing the allocation of resources across multiple partitions in an integrated modular avionics system will improve the quality of reconfiguration blueprints. In this paper, we use the term "load" to refer to the evenly distributed allocation of partition resources. Load is calculated as the weighted sum of CPU utilization and memory utilization for each partition. The calculation formula for establishing load balancing is provided in Equation (9) below.

$$\begin{cases} LB = 1 - 2 * \sqrt{\frac{1}{M} \sum_{i=1}^{M} \left(Load_{Pi} - \overline{Load} \right)^2} \\ Load_{pj} = \mu_1 * C_{use}^{P_j} + \mu_2 * M_{use}^{P_j} \\ C_{use}^{P_j} = \frac{\sum_{j=1}^{|Application_{list}|} WCET_{app_{ij}}}{Partition_{durtime}} \\ M_{use}^{P_j} = \frac{\sum_{j=1}^{|Application_{list}|} mem_{app_{ij}}^{Partition_i}}{memsize_{Partition_i}} \\ \mu_1 + \mu_2 = 1 \end{cases} \tag{9}$$

where LB denotes load balancing and $Load_{P_j}$ denotes denotes the load on the partition numbered i. $\overline{Load}$ denotes the average resource load on all partitions. μ_1 denotes the weight of CPU utilization, and denotes the weight of memory utilization. $C_{use}^{P_j}$ denotes the partition p_j of CPU utilization, the $M_{use}^{P_j}$ denotes the weight of the partition's p_j of the partition. The memory utilization of the partition.

B.Reconfiguration fault tolerance

Refactoring Fault Tolerance (RFT) is used to measure the impact of refactoring blueprints on the system. The success rate of migrating the software set to the remaining available partitions is determined by the weighted proportion of the software that is successfully migrated. This paper emphasizes the importance and criticality of integrated modular avionics system software. The integrated modular avionics system software is categorized into five levels. Level 5 is the highest level of importance, while Level 1 is the lowest level of importance. According to the definition of migration success rate, the indicator for migration success rate is established as shown in Equation (10) below:

$$RFT = \frac{\sum_{i=1}^{n_M} G_{M_i}}{\sum_{i=1}^{N_M} G_{M_i}} \tag{10}$$

C.Reconfiguration time rate

During the reconfiguration migration process, each software that requires reconfiguration migrates in parallel, following the sequence outlined in the reconfiguration blueprint. The individual processor refactoring time is defined as the total time required to reload

all the software in the respective partition of the processor. The lower the percentage of time spent on refactoring, the higher the fault tolerance of the refactoring blueprint, and the lesser impact it has on system functionality. The reconfiguration time rate metric is established as shown in Equation (11) below.

$$\begin{cases} RTR = 1 - \frac{Max(T_{CPU_s})}{T_{max}} \\ T_{CPU_s} = \sum_{i=1}^{n} T_{App_i} \end{cases} \tag{11}$$

RTR denotes the reconfiguration time. T_{CPU_s} denotes the reconfiguration recovery time for processor numbered s. T_{max} denotes the maximum reconfiguration time. N_{re} denotes the processor c_k number of software in the processor that needs to be reconfigured. The T_{M_i} denotes the application software M_i reconfiguration migration time of the application software.

D.Load optimization function

To allocate more resources for future reconfiguration actions and enhance the success rate and efficiency of reconfiguration, a load optimization function is proposed. It is shown in Equation (12) below.

$$\begin{gathered} Minf_1 = \sum_{i=1}^{M} \sum_{j=1}^{N} \mu_1 C_{use}\left(P_i, App_{ij}\right) + \mu_2 M_{use}\left(P_i, App_{ij}\right) \\ \begin{cases} \mu_1 + \mu_2 = 1 \\ \forall Mem_{use}\left(P_i, App_{ij}\right) \leq Mem_{max} \\ \forall C_{use}\left(P_i, App_{ij}\right) \leq CPU_{max} \end{cases} \end{gathered} \tag{12}$$

f_1 denotes the load optimization function. $C_{use}(P_i, App_{ij})$ denotes the software to be migrated App_{ij} to be migrated to the partition P_i after being migrated to a partition. $\mu_2 M_{use}(P_i, App_{ij})$ denotes the CPU utilization of the software to be migrated App_{ij} to be migrated to the partition P_i memory utilization after being migrated to a partition. $Mem_{use}(P_i, App_{ij})$ represents the memory utilization of the software to be migrated, App_{ij}, after it has been migrated to partition P_i. Mem_{max} indicates the maximum memory available to the processor. And $C_{use}(P_j,D_i)$ Indicates the maximum memory available on the processor for the software to be migrated App_{ij} to be migrated to the partition P_i CPU resource consumption after being migrated to the partition.CPU_{max} indicates the maximum CPU resources available to the processor.

E.Multi-objective optimization function

The multi-objective optimization function is established as shown in Equation (13) below.

$$\begin{gathered} Maxf_2 = \lambda_1 LB + \lambda_2 RFT + \lambda_3 RTR \\ \begin{cases} 0 < \lambda_i < 1, i \in [1,3] \\ \lambda_1 + \lambda_2 + \lambda_3 = 1 \end{cases} \end{gathered} \tag{13}$$

Where f_2 denotes the multi-objective optimization function, λ_1 denotes the load balancing weights, and λ_2 denotes the reconfiguration fault tolerance weights, and λ_3 denotes the reconfiguration time share weights.

3.3. Reconfiguration method based on Improved VDN algorithm

Overview. In this paper, the algorithm for generating intelligent reconfiguration blueprints incorporates the design concepts of the VDN multi-agent reinforcement learning algorithm. It refines the characteristics of the reconfiguration task of a comprehensive electronic system and incorporates them into the design of agent, state design, action space

design, and payoff function design. Specifically, the software that requires reconfiguration is referred to as an agent. The decision sequences are organized based on the software's level of importance, which ranges from high to low. The partition state is represented by the weighted sum of time and space resources consumed by all partitions. The partitioned states, along with the observations made by the agent about their states, are utilized as the states in the reinforcement learning process. The assignment of each agent to an available partition or scheduling failure is abstracted as the action space. For the exploration and exploitation strategy, this paper adopts a strategy based on the cumulative reward of the average sequence. The cumulative rewards in the environment information are utilized to allow the algorithm to adaptively adjust the exploration factor during the training phase. For the reward function, two separate functions are designed for the agent competition phase and the agent cooperation phase. It is used to achieve the goal of balancing the optimization of agent systems and overall efficiency. When the algorithm reaches the termination round, the reconfiguration configuration blueprint is obtained.

Environmental design. Each software that requires reconfiguration is defined as a standalone agent. Each agent can independently sense the state of the environment, take actions, and receive feedback from the environment. All the agents compete and cooperate within the integrated modular avionics system environment. In this paper, a state matrix is used to describe the reconfiguration state of the integrated modular avionics system.

In conjunction with the integrated modular avionics system, the CPU information, partition information, and application software information are enumerated first. Let's consider an integrated modular avionics system that consists of three CPUs and six partitions. Each partition has 50 ms of available time slice resources and 50 KB of available memory resources. During the process of refactoring, the sequence of software refactoring significantly impacts the quality of the refactoring and the migration plan. This paper decomposes the fault granularity at the software level when defining the refactoring state. In other words, CPU2 faults are divided into software M3 faults and software M4 faults. At this point, two agents, M3 and M4, are generated according to the algorithm's design. As shown in Figure 2, the state matrix of agent M3 has been established.

The specific meaning of the state matrix is illustrated in the Equations. (14) and (15) below. The state matrix is divided into two rows. The first row represents the states of all partitions of the integrated modular avionics system, which correspond to the current observations of the agent in its environment. The weighting coefficients for the normal partitions are 0.5 for both the operation cycle and the operation memory, and -0.5 for the faulty partitions. The second row represents the state of the software that needs to be reconfigured. This pertains to the observation of agent within the framework of multi-agent reinforcement learning [35,36,37]. The observations of the agent about the environment

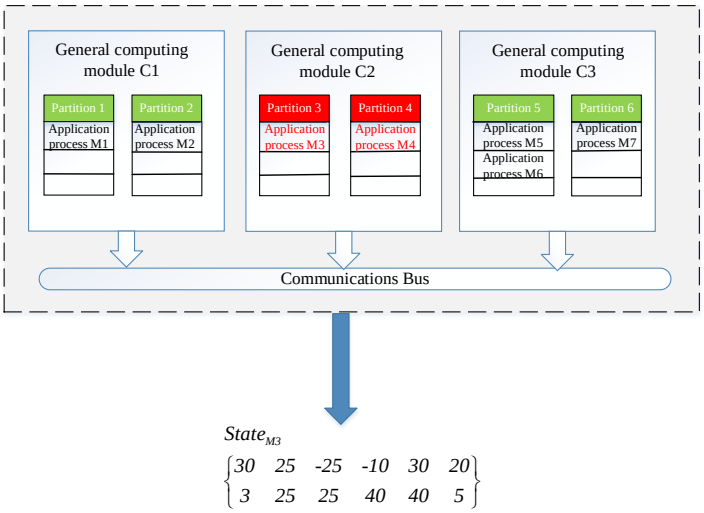


Fig. 2. M3 Reconfiguration State Chart

are combined with their observations of themselves to form comprehensive insights.

$$State_{(i,j)} = \begin{cases} ParInfo_{p_j} & (i = 0, 0 < j < m) \\ App_{index} & (i = 1, j = 0) \\ App_{dur} & (i = 1, j = 1) \\ App_{mem} & (i = 1, j = 2) \\ App_{WD} & (i = 1, j = 3) \\ App_{deadline} & (i = 1, j = 4) \\ App_{rank} & (i = 1, j = 5) \\ 0 & else \end{cases} \tag{14}$$

$$ParInfo_{p_j} = \begin{cases} 0.5 * Duruse_{p_j} + 0.5Memuse_{p_j} & (0 < j < m, P_j trouble - free) \\ -0.5 * Duruse_{p_j} - 0.5Memuse_{p_j} & (0 < j < m, P_j trouble) \end{cases} \tag{15}$$

Where $State_{(i,j)}$ denotes the state matrix. m denotes the number of partitions. $ParInfo_{Pj}$ denotes the information of partition j. App_{index} denotes the software number. App_{dur} denotes the software runtime period. App_{mem} denotes the software runtime memory. App_{WD} denotes the worst runtime of the software. $App_{deadline}$ denotes the software cutoff time. App_{rank} denotes the software importance level. $Duruse_{Pj}$ denotes the remaining time slice resources of partition j. and $Memuse_{Pj}$ denotes the remaining memory resources of partition j.

In this paper, the reconfiguration of agent follows the software priority order, indicating that software with high priority is reconfigured before software with low priority. When the agent with high priority executes an action, it affects the environment and other

prioritized observations. In this example, the agent M3 has a priority of 5, while the agent M4 has a priority of 4. M3 executes the refactoring before M4. After executing the reconfiguration action, the agent M3 migrates the application to partition 2. But partition 2 does not have any available resources at the moment. When the agent M4 executes the action, its state matrix changes, as depicted in Figure 3.

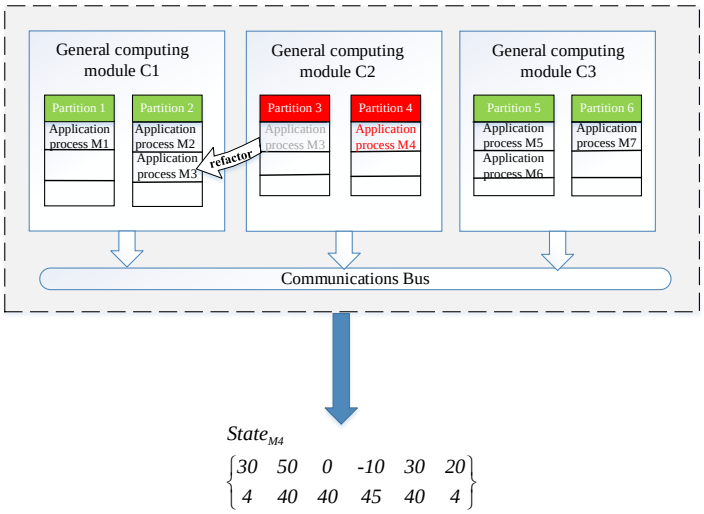


Fig. 3. M4 reconfiguration state diagram

Agent Action Design. In this paper, an individual action of an agent is defined as the process of scheduling the current software to be reconfigured for a specific available partition. The set of all executable discrete scheduling actions constitutes the action space. The specific definition is shown in Equation (16). When there are m available partitions, the size of the current action space is m. It contains m successful scheduling actions. That is, the software that needs to be reconfigured can be scheduled for any of the available partitions.

$$\begin{gathered} Action_i = (Act_0, Act_1 ... Act_{m-2}, Act_{m-1}) \\ Act_j = \begin{cases} 1 \; j = k, k \in [0, m-1] \\ 0 \qquad else \end{cases} \\ ActionSpace \subseteq \{Action_i\} \qquad (i \in [0, m-1]) \end{gathered} \tag{16}$$

m denotes the number of partitions. $Action_i$ denotes the i-th action. Act_j denotes the j-th action component, and $ActionSpace$ denotes the set of action spaces.

Exploratory strategies based on average sequential cumulative rewards. During the process of algorithm training, the agent selects actions based on a strategy of exploration

and exploitation. In the traditional VDN algorithm, the strategy is defined as shown in Equation (17) below.

$$\pi(a|s) = \begin{cases} 1-\varepsilon & a = argmaxQ(s,a) \\ \varepsilon & a = random(a) \end{cases} \tag{17}$$

The algorithm generates a random number specifically when it is executed. When the random number is less than the exploration coefficient, the algorithm executes the exploration policy. This policy randomly selects an action from the action space for execution. When the random number is greater than 0, the algorithm executes the exploitation strategy, which selects the optimal action based on the existing experience. The traditional strategy is to gradually decrease the value over time. It does not consider the feedback information from the agent during the training process, and it lacks informed action selection.

The algorithm in this paper explores a utilization strategy using an adaptive approach based on the cumulative reward of the average sequence [38]. The exploration factor is dynamically adjusted based on the cumulative reward value of the agent during the training process. In reinforcement learning, the sequence of actions taken by an agent from the initial state to the goal state is crucial. The cumulative reward of the sequence is defined by Equation (18).

$$G_t = \sum_{k=0}^{T} \gamma^k R_{t+k+1} \tag{18}$$

G_t denotes the cumulative serial reward at moment t,γdenotes the reward decay coefficient, and R_{t+k+1} denotes the reward value of the k+1st action at moment t. Therefore, the exploration factor$\varepsilon$and the average sequence cumulative reward are defined as shown in Equation (19) below.

$$\begin{aligned} \varepsilon &= \frac{1}{1+\log_2(\bar{G}+1)} \\ \bar{G} &= \frac{1}{L}\sum_{i=e-L}^{e-1} G_i \end{aligned} \tag{19}$$

L denotes the number of sequence entries, $\bar{G}$ denotes the average reward of the first L sequences. With the exploration utilization strategy described above, the exploration factor is dynamically adjusted based on the average cumulative reward obtained sequentially. Aggregated environmental feedback information guides the agent in choosing their actions.

Reward function design. During the reconfiguration of an integrated modular avionics system, a new blueprint is generated after performing an application migration operation. The reward function is used to evaluate the impact of the action on the refactoring process. For a positive impact, the reward function provides positive feedback. Increasing the reward value tends to motivate the agent to perform better actions in the next round of training. The reward function provides negative feedback for unfavorable outcomes. Decrease the reward value to discourage the agent from engaging in negative actions. The algorithm in this paper is designed with two reward functions. These functions are used to calculate the cost of actions and the overall reward of the agent.

The cost-reward function of agent actions is primarily applied during the competitive phase of multi-agent reinforcement learning. During this phase, the agents compete with each other for time slices and memory resources in order to achieve higher local payoffs. For an agent system, successful migration will result in the efficient utilization of the allocated time slice and memory resources for each partition. And the consumed resources are inversely proportional to the cost of that action. That is, the smaller the ratio of consumed partition resources to available partition resources, the more favorable the migration of other agents. Once the migration fails, it indicates that the action is unfavorable and penalizes the agent. The final reward function of a single agent is shown in Equation (20).

$$r(s,a)_t^j = \begin{cases} 1-\mu_1 * Cpuuse^j_{Patition_i} - \mu_2 * Memuse^j_{Patition_i} & migration\ successful \\ -1 & migration\ failed \\ \mu_1 + \mu_2 = 1 & \end{cases} \tag{20}$$

$r(s,a)_t^j$ denotes the local reward return value for agent j at time t. $Cpuuse^j_{Patition_i}$ denotes the time slice occupancy of the partition after agent j migrates to partition i. $Memuse^i_{Patition_j}$ denotes the memory occupancy of the partition after agent j migrates to partition i. μ_1 denotes the time slice weighting coefficients, and μ_2 denotes the memory weighting coefficients.

The overall reward function is primarily used in the cooperative phase of multi-agent reinforcement learning. In this phase, the goal of cooperation among the agents is to achieve a higher overall reward. At this time, some individuals may receive a reduced reward in exchange for an overall increase in rewards. This prevents all agents from falling into local optimal solutions by avoiding the use of greedy strategies. The algorithm's explorability and stability have been enhanced. According to the multi-objective optimization constraint metrics in the resource constraints of the reconfiguration blueprint, this paper evaluates the current agent actions based on load balancing, reconfiguration fault tolerance, and reconfiguration time ratio. The overall reward function is shown in Equation (21).

$$R(s,a)_t^j = \begin{cases} \lambda_1 * LB + \lambda_2 * RTR + \lambda_3 * RID & migration\ successful \\ -1 & migration\ failed \end{cases} \tag{21}$$

$R(s,a)_t^j$ denotes the overall reward return value for agent j at time t.

4. Results and Discussion

4.1. Experimental design

In this paper, we have reconfigured the blueprint for the algorithm that generates intelligence, written in the Python language. The parameters of the agent need to be set when training the agent. The parameter settings are shown in Table 1 below.

The network topology structure employs a traditional DQN MLP network, as illustrated in Figure 4 below.

Based on the system migration model of the integrated modular avionics system and the experimental data, the reconfiguration configuration state migration graph shown in

Table 1. Parameter Settings

parameters	parameter value
Training round	20000
Competitive learning rate	0.1
Cooperative learning rate	0.9
Sample size	16
Empirical sample size	2000
Target network update rate	0.01
Memory utilization weights μ_1	0.5
Time-slice utilization weights μ_2	0.5
Starting rounds of cooperation	16000
Average accumulated experience bars L	10

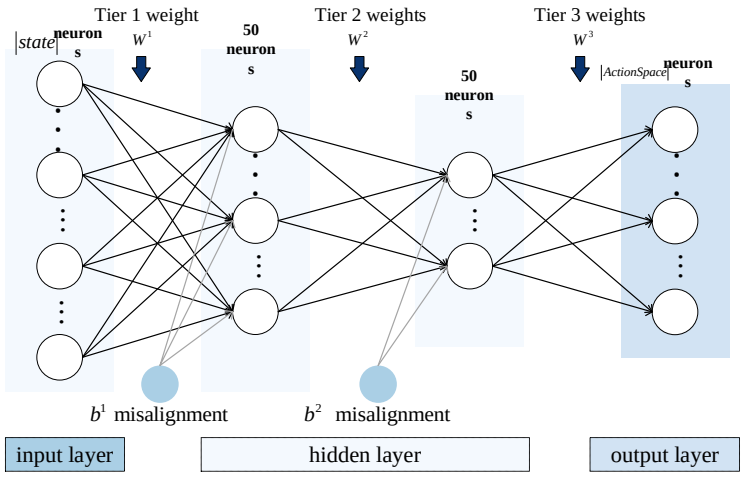


Fig. 4. Schematic of MLP network topology

Figure 5 below has been constructed. Among them, S0 is a blue node representing the initial configuration blueprint. There are eight faults, each of which generates a reconfiguration state. There are eight green nodes corresponding to S1 to S8. S_{end} is a red node representing the final state of the reconfiguration. Two scenarios cause a reconfiguration state to enter the end state. The first reason is that there are no faults that require reconfiguration. The second reason is that the available system resources are insufficient to meet the reconfiguration requirements.

Specifically, the initial state is S0. When the system injects a single fault with fault number "Fault1," System Partition 1 fails. The system invokes the intelligent reconfiguration algorithm to generate a new reconfiguration blueprint. The system restores the normal state of S1 by migrating the application and utilizing the reconfiguration blueprint. When the system injects a continuous fault with a fault number of Fault5, a continuous fault occurs in system partitions 1 and 3. At this point, the system proceeds to invoke the intelligent reconfiguration algorithm to create a new reconfiguration blueprint based on S1.

Subsequently, the system executes the reconfiguration process to restore the system to its normal state, S5. Since the consecutive faults for Fault5 have been reconfigured, there are no longer any faults in the system. Therefore, the state is transferred to the reconfiguration end state S_{end}. The remaining steps of the state transfer process are the same as described above.

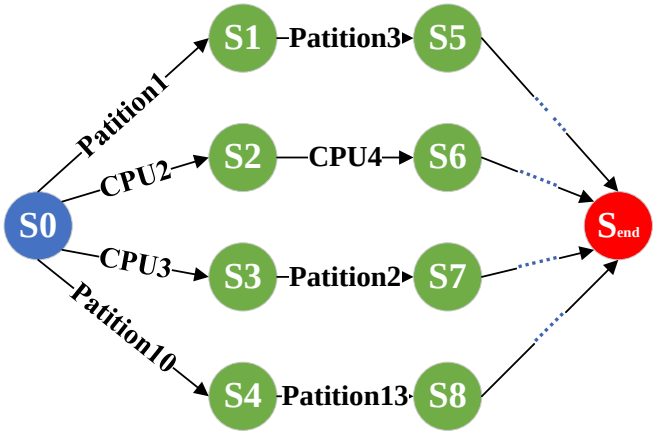


Fig. 5. System Reconfiguration Configuration State Migration Diagram

4.2. Experimental results and analysis

The configuration table is generated for the eight faults mentioned above in this experiment. After the experiment, all the faulty applications were moved to the available partitions. The reconfiguration blueprint meets the requirements for time constraints, space constraints, and uniqueness constraints. The validity of the experimental results is verified.

Parameters affect experimental results. For the parameter impacts of the algorithm, this paper takes the reconfiguration impacts arising from an environment in which fault 1 occurs in an integrated modular avionics system as an example. The impact of the single fault parameter experiment is analyzed. Fault 1 is a typical single fault. That is, 1 partition fails and 1 software needs to be reconfigured. For the aforementioned fault, the experimental results are presented in Table 2 of the Single Fault Parameter Experiment data.

This paper presents an example of the reconfiguration impact in an environment where a Fault6 fault occurs in an integrated modular avionics system. It analyzes the impact of experiments on continuous fault parameters. Fault6 is a typical continuous fault. Three partitions have failed and six software need to be reconfigured. For the above fault, the results of the experiment are shown in Table 3 of the Continuous Fault Parameter Table experiment data below.

Table 2. Single Fault Parameter Experiment Data Sheet

λ_1	λ_2	λ_3	maximum values	convergence value	LB	$RFFT$	MSR
0.45	0.1	0.45	0.941853	0.931086	0.851807	0.940000	1.000000
0.4	0.2	0.4	0.939961	0.931882	0.859705	0.940000	1.000000
0.35	0.35	0.3	0.937844	0.929896	0.859705	0.940000	1.000000
0.25	0.45	0.3	0.942316	0.939148	0.858650	0.940000	1.000000
0.2	0.4	0.4	0.952482	0.946362	0.862412	0.940000	1.000000
0.1	0.45	0.45	0.961241	0.958732	0.887322	0.940000	1.000000

Table 3. Experimental Data Sheet for Continuous Fault Parameters

λ_1	λ_2	λ_3	maximum values	convergence value (math.)	LB	$RFFT$	MSR
0.45	0.1	0.45	0.625812	0.374881	0.682018	0.660000	0.666667
0.4	0.2	0.4	0.674227	0.675892	0.640000	0.560000	1.000000
0.35	0.35	0.3	0.632193	0.599479	0.679393	0.520000	1.000000
0.25	0.45	0.3	0.602516	0.579556	0.684264	0.580000	1.000000
0.2	0.4	0.4	0.631485	0.597244	0.701263	0.580000	1.000000
0.1	0.45	0.45	0.586616	0.357389	0.784097	0.660000	0.888889

According to the above experimental results, when a single failure occurs, all parameter selection cases are able to complete the reconfiguration for that failure. The reconfiguration time share and migration success rate are the same. The algorithm's return value at this point depends on the load balancing situation. From the experiments, we can see that the final choice of the single failure parameter λ_1 Select 0.1.λ_2 Select 0.45.λ_3Choosing 0.45. The impact of load balancing value is more significant than other parameter choices.

When continuous failure occurs, it is not possible to explore a viable reconfiguration option when both 0.45 and 0.1 are selected for λ_1. The metrics for return value are unsatisfactory. When λ_1 non-0.4 is chosen, the return value hardly exceeds 4. According to the experimental results, the return value is more significant than other parameter choices when λ_1 Select 0.4.λ_2 Select 0.2.λ_3 Select 0.4 for continuous failure.

Single Fault Experimental Results. Experiments were primarily conducted on the quality of blueprints and convergence speed.

A.Blueprint quality experiment

The single fault blueprint quality comparison experiment compares four metrics: the return value of the individual reconfiguration algorithm, load balance, reconfiguration time ratio, and migration success rate. The multi-agent algorithm calculates the average value of the agent. To mitigate the instability of reinforcement learning, the data is averaged over 10 rounds of iterations. The single fault has a specific location and less faulty software. The experiment allows for a maximum of 1 fault location in the CPU and a maximum of 3 faulty software. Figure 6 displays the graphs used to compare the quality of reconfiguration blueprints for S1-S4.

According to the figure above, it can be seen that in the case of a single fault, all the various intelligent reconfiguration algorithms have MSR metrics of 1. This means that in the case of fewer software faults, all the algorithms are capable of performing

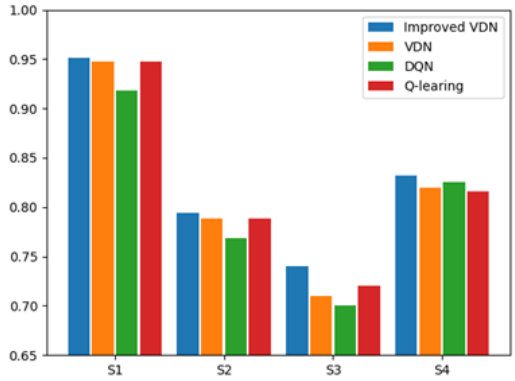


Fig. 6. Comparison of the quality of S1-S4 reconfiguration blueprints

the reconfiguration task. From the data in the table, it can be seen that in the case of the same number of training sessions. The Improved VDN algorithm is capable of generating blueprints with higher load balancing metrics compared to other algorithms. As a result, the reward values also converge to a higher level.

B.Convergence rate experiments

To compare the convergence speed of various reconfiguration blueprint generation algorithms in the presence of a single fault. In this paper, we selected S1 training data generated by an integrated modular avionics system in a typical single fault environment, specifically Fault1, as an example for validation and analysis. The reward values of the four algorithms were compared over 20,000 training rounds. The average reward value per 10 rounds for the total number of training rounds for the Fault1 fault is plotted against the blueprint reward trend, as shown in Figure 7 below. The figure shows the number of converged rounds for the Improved VDN, VDN, DQN, and Q-learning blueprint algorithms based on fault improvement: 665, 1,163, 1,661, and 1,827 rounds, respectively. The Improved VDN algorithm demonstrates a faster convergence speed compared to other algorithms.

Continuous Failure Experiment Results. The following provides a comparison of blueprint quality and convergence speed.

A.Blueprint quality comparison

More faulty software is involved in consecutive faults, with up to 2 CPUs implicated in the fault location and up to 6 instances of faulty software in the experiment. Figure 8 is used to compare the quality of S1-S4 reconfiguration blueprints.

From the figure, it can be seen that there is a continuous failure, such as S6. At this time, the system resources drop dramatically. The corresponding mean squared error (MSR) values of the Improved VDN, VDN, DQN, and Q-learning algorithms are 1, 0.666667, 0.666667, and 0.5, respectively. It shows that in the case of a large number of faulty software, only VDN can generate a usable reconfiguration blueprint within a limited time. The other three algorithms are unable to complete the refactoring and can

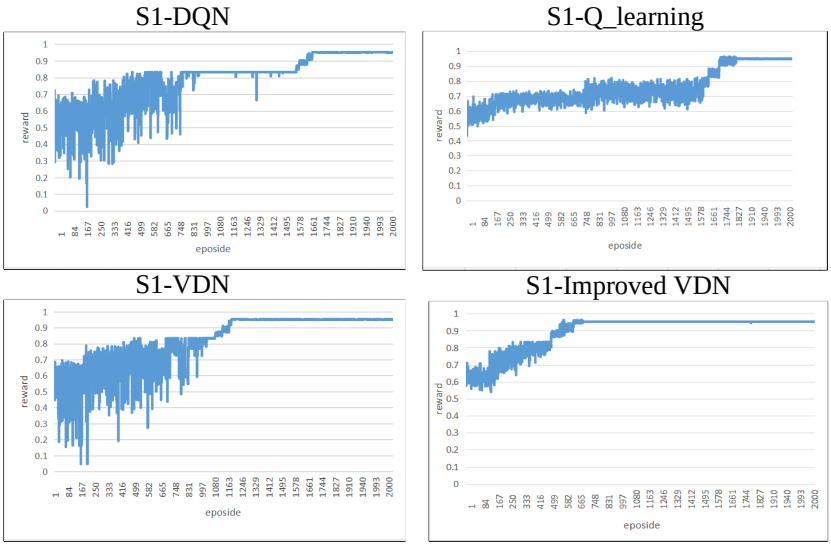


Fig. 7. S1 Blueprint Reward Change Trend Chart

only attempt to migrate the software to be refactored based on the software priority. The software with a lower migration success rate has higher load balancing metrics and refactoring time ratio metrics compared to the Improved VDN algorithm. However, the reward value for final convergence is still lower than that of the Improved VDN algorithm. As can be seen from the data in the table, with the same amount of training, the Improved VDN algorithm is more effective at generating reconfiguration blueprints and converges faster than other algorithms. It also has an advantage over other comparative algorithms in its ability for simultaneous multi-objective optimization.

B.Comparison of convergence speeds

To compare the convergence speed of various reconfiguration blueprint generation algorithms in the presence of continuous faults. In this paper, the training data of the integrated modular avionics system generating S6 under the typical continuous fault environment, Fault6, is selected as an example for validation and analysis. And the algorithms are selected to compare the corresponding reward values of the four algorithms in 20,000 rounds of training. As shown in Table 4, the reward values of the three reconfiguration blueprint generation algorithms changed after 20,000 iterations of training in the presence of Fault1 fault.

For the Fault6 fault, the average reward value of every 10 rounds of training is used to plot the trend of reward change in the blueprint. Under this framework, the exploration factor is adaptively adjusted by utilizing cumulative rewards derived from the average sequence. The number of converged rounds for the improved intelligent reconfiguration blueprint algorithm based on VDN is 1744 rounds. The intelligent reconfiguration blueprint generation algorithms based on VDN, Q-learning, and DQN are all in an oscillating state and cannot converge.

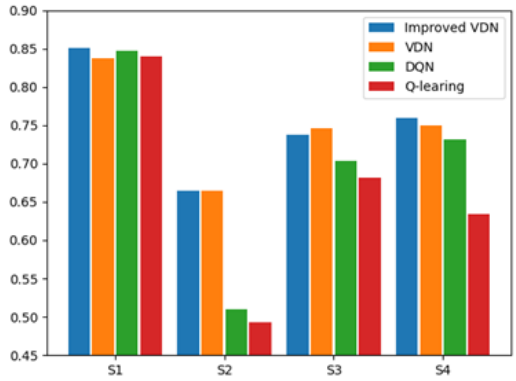


Fig. 8. Comparison of the quality of S5-S8 reconfiguration blueprints

Table 4. S6 State 1-20,000 Training Blueprint Quality Values

Number of iterations	Improved VDN	VDN	DQN	Q-learning
1	0.162586	0.162586	0.137438	0.000125
1000	0.243601	0.243601	0.173120	0.073178
2000	0.321210	0.321210	0.179622	0.180646
3000	0.483040	0.483039	0.192260	0.209591
4000	0.482808	0.482808	0.190402	0.235901
...	...	...	...	...
10000	0.449848	0.449848	0.361703	0.273046
110000	0.432568	0.432568	0.321205	0.291970
120000	0.429175	0.429175	0.396585	0.293272
...	...	...	...	...
18,000	0.670433	0.567463	0.513909	0.493352
19000	0.670433	0.565181	0.515823	0.492240
20000	0.670433	0.569339	0.512242	0.494273

In summary, this paper proposes an intelligent reconfiguration blueprint generation algorithm based on VDN and average sequence cumulative rewards. Compared to traditional linearly decreasing exploration factor strategies, the introduced exploration strategy in this algorithm exhibits higher adaptability and flexibility. It better addresses the learning dynamics of agents during the training process, avoiding blind reduction of the exploration factor that might lead to local optima. Through the design of two reward functions, the system can effectively evaluate the actions of agents, guiding them to achieve a balance between cooperation and competition in multi-agent systems. This enables better completion of the reconfiguration tasks for electronic systems.

A comprehensive analysis of experimental results was conducted, which included parameter selection experiments, single-fault experiments, and continuous-fault experiments. The proposed algorithm demonstrates advantages in both the quality and speed metrics of reconfiguration blueprint generation compared to traditional algorithms based on VDN, DQN, and Q-learning. Furthermore, its advantages become more pronounced,

especially in complex scenarios such as continuous faults. The algorithm presented in this paper demonstrates improved applicability and robustness.

Hence, it can be concluded that this paper incorporates the average sequence cumulative reward mechanism into the VDN algorithm and applies it to the comprehensive electronic system reconfiguration blueprint generation method. This approach not only enables the generation of higher-quality reconfiguration blueprints but also achieves the efficient generation of such blueprints.

5. Conclusion

This paper presents the design and validation of an intelligent generation algorithm for reconfiguration blueprints using multi-agent reinforcement learning. The algorithm utilizes average sequential cumulative rewards instead of the conventional VDN algorithm's $\varepsilon\text{-}greedy$ strategy. The algorithm design includes four main elements: integrated modular avionics system state design, exploration and utilization strategy design, action space design, and reward function design. This paper validates the proposed algorithm by comparing its experimental results with reconfiguration blueprint generation algorithms based on traditional VDN, Q-learning, and DQN. The comparison demonstrates that the proposed algorithm exhibits superior convergence characteristics and optimization effects, making it suitable for addressing similar multi-objective optimization reconfiguration problems. In addition, there is room for optimizing load balancing in complex fault environments, and evaluation metrics such as software and module correlation are not considered. In the design of agent systems, as the complexity of faults increases and the number of software components requiring reconfiguration grows, the number of agent systems also increases. This significantly impacts both the solving speed and the convergence rate. Therefore, the definition of an agent system can be further refined. Similarly, bandwidth, ports, routing, topology, and other factors need to be considered in the future to enhance the reconfiguration of distributed integrated electronic systems.

References

1. Hubbard, P.D.: Fault management via dynamic reconfiguration for integrated modular avionics. Computer Science (2015)
2. Chen, J., Du, C., Han, P.: Scheduling independent partitions in integrated modular avionics systems. PLOS ONE 11(12), e0168064 (2016)
3. Wang, P., Zhao, C., Yan, F.: Research on the reliability analysis of the integrated modular avionics system based on the aadl error model. International Journal of Aerospace Engineering (2018)
4. Burger, S., Hummel, O.: Towards automatic reconfiguration of aviation software systems. In: 2011 IEEE 35th Annual Computer Software and Applications Conference Workshops. pp. 200–205. IEEE, Munich, Germany (2011)
5. Burke, M., Audsley, N.: Distributed fault-tolerant avionic systems - a real-time perspective. arXiv - CS - Distributed, Parallel, and Cluster Computing (2010)
6. Wang, H., Niu, W.: A review on key technologies of the distributed integrated modular avionics system. International Journal of Wireless Information Networks 25(12), 358–369 (2018)
7. He, D., Qiao, Q., Gao, J., Chan, S., Zheng, K., Guizani, N.: Simulation design for security testing of integrated modular avionics systems. IEEE Network 34(1), 159–165 (2020)

8. Gui, S., Luo, L., Tang, S., Meng, Y.: Optimal static partition configuration in arinc653 system. Journal of Electronic Science and Technology 9(4) (2011)
9. Blikstad, M., Karlsson, E., Lööw, T., Rönnberg, E.: An optimisation approach for pre-runtime scheduling of tasks and communication in an integrated modular avionic system. Optimization and Engineering 19, 977–1004 (2018)
10. Cui, Y., Shi, J., Wang, Z.: Backward reconfiguration management for modular avionic reconfigurable systems. IEEE Systems Journal 12(1), 137–148 (2018)
11. da Fontoura, A.A., do Nascimento, F.A.M., Nadjm-Tehrani, S., de Freitas, E.P.: Timing assurance of avionic reconfiguration schemes using formal analysis. IEEE Transactions on Aerospace and Electronic Systems 56(1), 95–106 (2020)
12. Suo, D., Zhu, J., An, J.: A new approach to improve safety of reconfiguration in integrated modular avionics. In: 2011 IEEE/AIAA 30th Digital Avionics Systems Conference. pp. 1C4–1–1C4–12. IEEE, Seattle, WA, USA (2011)
13. Huseyinov, I., Bayrakdar, A.: Novel nsga-ii and spea2 algorithms for bi-objective inventory optimization. Studies in Informatics and Control 31(3), 31–42 (2022)
14. He, Z., Li, J., Wu, F., Shi, H., Hwang, K.S.: Derl: Coupling decomposition in action space for reinforcement learning task. IEEE Transactions on Emerging Topics in Computational Intelligence (2023)
15. Housseyni, W., Mosbahi, O., Khalgui, M., Li, Z., Yin, L., Chetto, M.: Multiagent architecture for distributed adaptive scheduling of reconfigurable real-time tasks with energy harvesting constraints. IEEE Access 6, 2068–2084 (2017)
16. Li, J., Shi, H., Hwang, K.: Using goal-conditioned reinforcement learning with deep imitation to control robot arm in flexible flat cable assembly task. IEEE Transactions on Automation Science and Engineering (2023)
17. Shi, H., Li, J., Mao, J., Hwang, K.: Lateral transfer learning for multiagent reinforcement learning. IEEE Transactions on Cybernetics (2021)
18. Jang, B., Kim, M., Harerimana, G., Kim, J.W.: Q-learning algorithms: A comprehensive classification and applications. IEEE access 7, 133653–133667 (2019)
19. Yang, Y., Juntao, L., Lingling, P.: Multi-robot path planning based on a deep reinforcement learning dqn algorithm. CAAI Transactions on Intelligence Technology 5(3), 177–183 (2020)
20. Sunehag, P., Lever, G., Gruslys, A., Czarnecki, W.M., Zambaldi, V., Jaderberg, M., Lanctot, M., Sonnerat, N., Leibo, J.Z., Tuyls, K., Graepel, T.: Value-decomposition networks for cooperative multi-agent learning. arXiv:1706.05296 (2017)
21. Zhou, Q., Gu, T., Hong, R., Wang, S.: An aadl-based design for dynamic reconfiguration of dima. In: 2013 IEEE/AIAA 32nd Digital Avionics Systems Conference (DASC). pp. 4C1–1–4C1–8. IEEE, East Syracuse, NY (2013)
22. Saadi, A., Oussalah, M., Hammal, Y.: A csp-based approach for managing the dynamic reconfiguration of software architecture. International journal of information technologies and systems approach 14(1), 18 (2021)
23. Chen, L., Wang, L.: Research on functional hazard analysis method for ima reconfiguration. Computer Engineering (2016)
24. Kim, H., Kim, J.: A load balancing scheme for gaming server applying reinforcement learning in iot. Computer Science and Information Systems 17(3), 891–906 (2020)
25. Nguyen, N.T., Luu, L., Vo, P.L., Nguyen, T.T.S., Do, C.T., Nguyen, N.: Reinforcement learning - based adaptation and scheduling methods for multi-source dash. Computer Science and Information Systems 20(1), 157–173 (2023)
26. Zhang, T., Chen, J., Lv, D., Liu, Y., Zhang, W., Ma, C.: Automatic generation of reconfiguration blueprints for ima systems using reinforcement learning. IEEE embedded systems letters 13(4), 182–185 (2021)
27. Liu, B., Zhang, Q., Wang, S.: Automatic generation of reconfiguration blueprints for ima systems using reinforcement learning. In: The 4th Annual IEEE International Conference on Cy-

ber Technology in Automation, Control and Intelligent. pp. 664–668. IEEE, Hong Kong, China (2014)
28. Hollow, P., McDermid, J., Nicholson, M.: Approaches to certification of reconfigurable ima systems. The International Council on Systems pp. 1–8 (2000)
29. Gaska, T., Watkin, C., Chen, Y.: Integrated modular avionics—past, present, and future. IEEE Aerospace and Electronic Systems Magazine 30(9), 12–23 (2015)
30. Balan, N., Ila, V.: A novel biometric key security system with clustering and convolutional neural network for wsn. Tehnički vjesnik 29(5), 1483–1490 (2022)
31. Mani, V., Yarlagadda, S.R., Ravipati, S., Swarnamma, S.C.: Ann optimized hybrid energy management control system for electric vehicles. Studies in Informatics and Control 32(1), 101–110 (2023)
32. Zhang, T., Zhang, W., Dai, L., Chen, J., Wang, L., Wei, Q.: Integrated modular avionics system reconstruction method based on sequential game multi-agent reinforcement learning. ACTA ELECTONICA SINICA 50(4), 954–966 (2022)
33. Wang, H., Zhong, D., Zhao, T.: Avionics system failure analysis and verification based on model checking. Engineering Failure Analysis 105, 373–385 (2019)
34. Li, X., He, D., Gao, Y., Liu, X., Chan, S., Pan, M., Choo, K.: Light: Lightweight authentication for intra embedded integrated electronic systems. IEEE Transactions on Dependable and Secure Computing 20(2), 1088–1103 (2023)
35. Kavitha, S., Uma Maheswari, N., Venkatesh, R.: Intelligent intrusion detection system using enhanced arithmetic optimization algorithm with deep learning model. Tehnički vjesnik 30(4), 1217–1224 (2023)
36. Long, G.L., Shi, L., Xin, G., Gao, S., Zhang, W., Xu, J.: Machine-vision-based online self-optimizing control system for line marking machines. Studies in Informatics and Control 32(2), 93–104 (2023)
37. Liu, Y., Zhang, Y., Jiang, Y., Liu, W., Yang, F.: Uwb-ins fusion positioning based on a two-stage optimization algorithm. Tehnički vjesnik 30(1), 185–190 (2023)
38. dos Santos Mignon, A., de Azevedo da Rocha., R.L.: An adaptive implementation of -greedy in reinforcement learning. Procedia Computer Science 109, 1146–1151 (2017)

Jing Cheng is an associate professor with the School of Computer Science and Engineering, Xi'an Technological University, China. Cheng received a Ph.D. from Northwestern Polytechnical University, China. Her research interests include mobile application testing, crowdsourcing testing, artificial intelligence, and software modeling. Contact her at chengjing@xatu.edu.cn.

Wen Tan is a graduate student at the School of Computer Science and Engineering, Xi'an Technological University. Tan received her bachelor's degree from Xi'an Technological University. Her research interests include reinforcement learning, computer networks. Contact her at tanwen@st.xatu.edu.cn.

Guangzhe Lv is a senior engineer at Xi'an Institute of Aeronautical Technology. He graduated from Xidian University with a master's degree. His research interests include embedded software and distributed systems. Contact him at guangzhe_lv@163.com.

Guodong Li is a researcher at Aviation Industry First Flight Academy and a PhD candidate at Northwestern Polytechnical University. His research interests include aircraft

airborne system design, simulation, testing, and experimentation. Contact him at guodonglee@163.com.

Wentao Zhang is a Ph.D. candidate at the School of Software, Northwestern Polytechnical University. He received his master's degree from Northwestern Polytechnical University in 2023. His research interests include reinforcement learning, game optimizationand planning decision making. Contact him at wentaoz1223@gmail.com.

Zihao Liu is a Design Assistant in the Sixth Research Department of Leihua Electronic Technology Research Institute, Aviation Industry Corporation of China. He received his master's degree from Northwestern Polytechnical University. His research interests include embedded, reinforcement learning. Contact him at 2020264414@mail.nwpu.edu.cn.

Received: November 29, 2023; Accepted: June 28, 2024.

Computer Science and Information Systems 21(4):1359–1387 https://doi.org/10.2298/CSIS240229038B

Biometric Lock with Facial Recognition Implemented with Deep Learning Techniques

José Misael Burruel-Zazueta[1], Héctor Rodríguez-Rangel[1], Gloria Ekaterine Peralta-Peñuñuri[1], Vicenç Puig Cayuela[2], Ignacio Algredo-Badillo[3], and Luis Alberto Morales-Rosales[4,⋆]

[1] Tecnológico Nacional de México/Instituto Tecnológico de Culiacán
Juan de Dios Batiz No. 310 pte, Guadalupe, 80220 Culiacán Rosales, Sin., México
d13170395@culiacan.tecnm.mx
hector.rr@culiacan.tecnm.mx
gloria.pp@culiacan.tecnm.mx
[2] Institut de Robòtica i Informàtica Industrial, CSIC-UPC
C/ Llorens i Artigas 4-6, 08028 Barcelona, España
vicenc.puig@upc.edu
[3] CONACYT-Instituto Nacional de Astrofísica, Óptica y Electrónica
Tonantzintla 72840, Puebla, México
algredobadillo@inaoep.mx
[4] CONACYT-Universidad Michoacana de San Nicolás de Hidalgo
Avenida Francisco J. Múgica S/N Ciudad Universitaria, Morelia, México
lamorales@conacyt.mx

Abstract. The increased criminal activity associated with unauthorized entry into facilities has become a global concern. Traditional mechanical locks suffer from drawbacks such as key loss, theft, duplication risks, and time-consuming operation. Therefore, biometrics has been explored as a key to accessing a restricted area. However, some challenges still need to be solved in developing such systems, including user registration, response speed, maintainability, and the ability to distinguish between real and fake individuals. This paper proposes and develops a biometric lock system (BLS) whose opening is performed by recognizing a person's face. It solves the challenges of re-training, antispoofing, real-time response, and works inside an embedding system. The proposed BLS overcomes these challenges using a pre-trained network called FaceNet for feature extraction and coding into 128-dimensional vectors. We use the characteristic vector provided by FaceNet and a cosine distance to recognize the persons. It also incorporates ResNet18 + remote photoplethysmography (rPPG) to avoid spoofing. The architecture was implemented in a BLS, demonstrating an impressive false acceptance rate of 0% under varying lighting conditions, with an average response time of 1.68 seconds from facial detection to door opening. The BLS has easy maintainable devices, providing enhanced security by accurately identifying individuals and preventing unauthorized access. The system can distinguish between real and fake people without using specialized hardware. Making it a versatile solution suitable for homes, offices, and commercial spaces. The results underscore the potential efficacy of our proposed BLS in mitigating security challenges related to unwarranted access to restricted facilities.

Keywords: FaceNet, Jetson Nano, CNN, Door lock, Embedded system

⋆ Corresponding Author

1. Introduction

Building burglary is a security issue that remains one of the main problems in the Americas according to worldwide incidence rates [40]. Only some countries, such as the United States, Mexico, and Canada, have reduced their rates in the last 15 years. However, cases such as Brazil, Ecuador, Guatemala, and Chile have alarmingly increased crime rates. Some locks are used regularly to protect buildings, houses, and offices, which are activated mechanically, electronically, smartly, and hybrid.

The facility or asset protection is done according to its importance (i.e., the bank deposit lock differs from a building lock). The type of lock is defined by the asset's characteristics, such as the security level, user number, and response time. The scenarios may vary depending on these concepts since the security levels for protecting a bank vault are higher than for a house or office. Using the same example, the number of users allowed in both restricted areas varies.

Mechanical locks boast resilience in power outages and durability owing to their robust materials. Nevertheless, they rely on physical keys susceptible to loss, duplication, and time-consuming operation, presenting challenges during emergencies or high-traffic scenarios. Conversely, electronic locks, including RFID cards, passwords, cryptography, NFC, biometrics, and hybrid systems, offer keyless entry, remote monitoring, and swift response times but are rendered useless without electricity and hinge asset security on the safety of accompanying devices or password management, often incurring higher costs. RFID cards [46] emulate traditional keys with enhanced user-friendliness and speed but remain vulnerable to loss or theft. Common passwords bolster security but heighten the risk of forgetting access credentials. Advanced techniques like one-time passwords (OTP) [43], IoT [43], cryptographic solutions [8], and NFC devices [42] necessitate additional devices for code or signal generation, elevating security levels but increasing user complexity and usability challenges in having to manipulate these devices.

Cost of the approaches related to smart locks are based on biometrics. These locks use the features present in the user's body as an access key [28]. Several user's body features distinguish one from the other, e.g., fingerprints [33, 45, 66], eye iris [44, 57, 58], vein patterns [54, 62], voice [13, 27, 65], face [6, 7, 17, 47], among others. These locks are easy to use, have a high response speed, and do not require additional devices or keys.

A hybrid lock consists of two or more unlocking techniques to increase the security of the systems [18, 23, 31]. They are used mostly in restricted areas, such as safes, bank vaults, and militarized areas. They are characterized by being difficult to use and prolonged access response.

There are some differences with respect to the biometric features implemented in a lock. Facial recognition requires less user cooperation with the system. Unlike fingerprint, vein, or voice recognition, the user has minimal interaction with the system [5]. The fingerprint or vein recognition requires the user to directly contact a scanner for the biometric readings. Alternatively, for an iris reading to be effective, the user must position his/her eye in proximity to an iris scanner. Therefore, showing a face to a camera is a simple task that almost anyone, including children, can do.

Face recognition techniques can be divided into three approaches: local, holistic, and hybrid [32]. Local approaches [2, 50] only use some features present on the face (eyes, mouth, nose, etc.), generally having faster processing speed but poor recognition effectiveness. Holistic approaches [9, 20] use the entire face for feature extraction, with high

effectiveness and speed with adequate hardware. Hybrid approaches [11, 37] combine the benefits of holistic and local approaches. However, they are slow to process but highly effective.

The most widely used technique in the field of face recognition is convolutional neural networks (CNNs) [1]. Due to their high specialization in image processing, they are a holistic approach that can represent facial features in small information vectors. The main disadvantage of CNNs is the high demand for data since they require high training to reach high rates of recognition effectiveness.

This training can take many hours of work depending on the number of users to be registered. This process presents a disadvantage since the administrator must stop the lock operation with each new user registration to train the facial identification system until the process finishes. In addition, the registration images of users to a facial recognition system are limited by the number of possible samples to be captured during the registration. This means that taking a few thousand pictures to train a CNN model is not feasible since this process would be unaffordable for the administrator and the users to register. This constraint is related to *One-shot learning*, which refers to a model that needs to be learned to classify with only one opportunity [38] (or shot).

Facial recognition locks can also be spoofed by users who present people's printed or digital photographs in front of a camera, making another challenge in developing facial recognition biometric locks. However, there are few proposals to deal with this type of attack. Some of these techniques directly involve the user by requesting some movement to verify a real person (e.g., blinking [14]).

Others use thermal cameras [64] with simple convolutional neural networks trained to identify real faces [10]. These techniques have shortcomings in that they can increase hardware costs (thermal cameras), can be spoofed by videos (blinking), or the light conditions and quality of a photograph could allow intruders to gain access.

In summary, biometric locks based on facial recognition are a viable alternative for building access control due to their effectiveness, response time, and ease of use. For its development, it is necessary to overcome some challenges: the amount of data available for user registration (one-shot-learning), detection that users are real (anti-spoofing) without impacting the ease of use and avoid the use of specialized hardware not to raise costs, and finally implement a recognition system in a maintainable embedded system.

Our proposal takes a strategic two-phase approach to development. The first phase involves coding and integrating face recognition techniques, specifically addressing the challenges of one-shot learning and anti-spoofing. In the second phase, we design a modular system of independent devices that seamlessly work together as an electronic door lock, ensuring a comprehensive and robust solution.

In this work, as the solution for *One Shot Learning*, we selected a CNN architecture named FaceNet [52], which extracts the characteristics of the faces and stores them in a vector, thus compares them to identify people based on their similarities. The main advantage of FaceNet is that it is a pre-trained network, so it does not require numerous images for facial identification. Hence, it can extract the features of new faces with a single image and store them in a vector. Another important feature of FaceNet is the effectiveness demonstrated during its development. According to the results obtained by the authors, it reached 99.67% effectiveness with training of more than 500 million images from more than 10,000 different subjects of all races, ethnicities, and ages.

To avoid spoofing through photographs displayed on the camera, a pre-trained CNN for facial skin detection called ResNet18 [21] and the remote photoplethysmography technique (rPPG) [34] were used. ResNet18 extracts small sections of the image corresponding to the user's facial skin. These skin extracts are processed with an rPPG algorithm, which can identify the heart rhythm through the light reflection changes caused by the blood circulation under the skin. Photographs do not show a change in reflected light, while videos depend directly on the device's luminance, which is why the rPPG functions as an anti-spoofing filter. The ResNet18 is a low-density network that decreases the performance impact on the computer and, in conjunction with a lightweight mathematical algorithm such as rPPG, provides a low computational cost reality detector.

We integrated an anti-spoofing system (rPPG), a feature extractor (ResNet18), and a facial recognizer (FaceNet) in an embedded system to develop a biometric lock with a zero percent false positive rate with a real-time response speed. It is important to mention that the ResNet18 + rPPG + FaceNet offer a balance between effectiveness, speed being executed, and reality detection compared to other prototypes highlighted in the state of the art, which, in some cases, demonstrate high performance in one aspect but too low or no performance in another.

Finally, a biometric-lock system (BLS) capable of controlling access through facial recognition was developed. The BLS was divided into three modules: (i) user interfaces and control, which describe the interaction of the hardware elements (Camera, Jetson Nano, display, and Arduino UNO) that are responsible for running the facial recognition system (FRS) and door opening, as well as interacting directly with the user, (ii) FRS, which represents the software responsible for processing the images received for registration or authentication of users of facial recognition by implementing ResNet18 + rPPG to validate that the faces received belong to real people and FaceNet for the face characteristics extraction, and (iii) Power circuit, this module is responsible for managing the electrical energy to lock and unlock the door.

This three-phase modulation allows us to visualize the elements of the prototype individually. Thus, it is possible to perform individual maintenance on each hardware device separately, increasing the maintainability of the BLS. Although there are BLS alternatives on the market, most of these devices are completely hermetic, making the maintenance task difficult, in addition to their high costs.

The scenario proposed in this work is to generate a lock capable of allowing access to a large number of people, which can vary from the size of a family (3-8 members) to a corporation of workers (100-200 members). It is also necessary not to require any key, password, card, or additional device, as well as to reduce the user's cooperation to only present his/her face to the camera. Therefore, it is considered a real-time response speed (0-2 seconds) to avoid a backlog of people trying to gain access to the building. This time was determined by observing a school RFID access control system under actual operating conditions. Also, a false acceptance rate (also known as false match rate, FMR) under 0.1% according to the regulations of NIST [19].

Therefore, the main contributions of this work are enumerated as follows:

- We propose a novel three-module architecture for facial recognition (ResNet18 + rPPG + FaceNet) available in [5]. ResNet18 and rPPG are responsible for calculating

[5] https://github.com/MisaelZazueta/Face-Recognition-Door-Lock

the visible heart rate on the user's face to rule out those who do not have a heartbeat, thus limiting facial recognition to real faces and not intruders seeking to circumvent the system through photographs or videos. All real faces are passed to FaceNet for identification with a single sample of the person's face; we do not carry out a learning stage; instead, we use the characteristic vector provided by FaceNet and cosine similarity to distinguish and recognize the persons listed in our database. This architecture avoids specialized equipment or further interaction with the user. It also provides a lightweight architecture suitable for running on embedded devices such as the Jetson Nano.

- We designed an interface and control interface architecture that consists of four elements (Jetson Nano, Camera, LCD Display, and Arduino UNO) for the software execution. This architecture is based on deep learning techniques (Jetson Nano) capable of making decisions (Boolean) and sending signals (Arduino UNO) to a power circuit to control the energy flow for an actuator. These interfaces are also responsible for generating the necessary images (Camera) to be processed by the software and interacting with the user as visual support (LCD screen).
- We built a BLS based on facial recognition by integrating the facial recognition system, the user and control interface architecture, and a power circuit. This BLS is suitable for the use suggested in the above-proposed scenario and is maintainable due to its software and hardware modules division.

According to the obtained results, the BLS has a high degree of security. This is because the system did not allow access to people not registered in the system, i.e., it obtained a 0% false acceptance rate (0 false positives). The system had only non-identification errors for some users. That is, it obtained an average false rejection rate 9%. However, the rate of people who achieved at least one successful access was 95.12%. In addition, the successful results showed an effectiveness of 91% in all shots. Finally, the response time from face detection to door opening averaged 1.68 seconds. These results, in conjunction with a 100% effectiveness in antispoofing, demonstrate a balance between the parameters sought for BLS development.

The following sections of this paper are organized as follows. Section 2, Related Work, briefly describes some of the work done by other authors. Section 3 describes the Materials and Methods used to develop the work. Section 4 discusses the results obtained in the BLS testing stages. Finally, Section 5 contains the conclusions inferred from the results obtained.

2. Related Work

The development of biometric door locking systems is a topic that computer researchers are addressing to develop security devices with better quality standards. However, some approaches specialize in hardware or software, and the minority of these developments encompass hardware and software. The following is a summary of some of the work related to solutions proposed by other authors. First, we describe work on biometric locks based on facial recognition, each using different hardware devices and facial recognition techniques to perform their tasks. Afterward, we describe outcomes with facial recognition techniques to visualize an overview of what is currently being used by the scientific community in terms of facial recognition.

In [41], the authors use a Raspberry Pi to run the facial recognition software. Because this device has limitations in running deep learning-based applications, a Neural Compute Stick 2 (NCS2) is added as an auxiliary device. Deep learning-based classification tasks are processed on the NCS2. Also, they use an electromagnetic lock driven by a coil, which in turn responds to the signals emitted by a GPIO output of the Raspberry Pi. A third party developed the implemented facial recognition. It consists of a neural network optimized for microcontrollers called MobileNet [24], combined with a Single Shot Detector (SSD) [36] system to detect that it is a real person. The results correspond to an accuracy of 88.75% in facial recognition with a population of 16 people and five repetitions per person. In addition, the system did not fail in any of the five attempts to deceive the design through digital photographs displayed on smartphones.

The authors of [35] developed a face recognition system implemented on a Jetson Nano microcomputer. A Haar algorithm performs face detection and FaceNet performs face recognition [52]. This system does not mention any reality detection stage to verify that the recognition applies only to real people. A reed switch initiated the algorithm once a person opened the door. Their results showed an accuracy of 97.7%

The paper [26] describes a door lock based on a Raspberry Pi microcontroller. Other hardware elements in this paper are a solenoid in charge of activating and deactivating the door lock, a relay to control the voltage flow to the solenoid, a camera module, and a buzzer that works as an audible alarm against intruders. Facial recognition is based on a simple convolutional neural network (CNN) [49]. The database comprised 1100 images, of which 1040 were for training and 60 for testing. The testing process was performed on five registered people and five not registered. Giving 120 samples at 0.5 m, 1.0 m, and 1.5 m distance and four-light conditions: morning, afternoon, evening, and night. The accuracy in 1.0 m and 0.5 m was 100% in all light conditions. The accuracy obtained at a 1.5 m distance was 97.5% in the morning, afternoon, and evening. At night it was 100%. The average accuracy was 97.83%.

A hierarchical system is developed in [60]. In this work, two neural networks operate together to determine the identity of individuals. A ResNet101 [29, 30] was modified to define the faces from images. Subsequently, the FaceNet [52] extracts a feature vector from the face, which is compared inside the database to identify the person. All this is executed inside a Raspberry Pi 4 microcontroller. The door lock is composed only of a servo-motor that transforms the electrical signals into mechanical movements(i.e., it rotates to drive a sliding door once it receives electrical signals directly from the microcontroller). In addition, this system sends e-mail notifications to the administrator every time someone interacts with the facial recognizer. The accuracy obtained was 87.36%. The tests consisted of a total of 230 samples.

In [15], a facial recognition system is described whose main feature is the inclusion of a stage of reality detection of the person by blinking eyes. This has the limitation that intruders can show videos of authorized persons blinking to the camera. The detection of faces in the images is performed by a technique called Histogram of Oriented Gradients (HOG) [16], which generates a vector of features. This vector is classified by a simple Support Vector Machine (SVM) [22]. The software is run on a Raspberry Pi 3 B+ microcontroller. The power module for the proposed door lock consists of a relay, a solenoid, and a power supply. Furthermore, an ultrasonic sensor is included to detect if a person is in front of the Pi-Camera used to obtain the images. The results generated in this work were

92.68% of maximum accuracy in optimal lighting conditions, and the average execution time was 10.196 seconds.

In [51], two different methods are used for face recognition. The first one is based on a local binary pattern histogram (LBPH) algorithm [55]. Each histogram created by this algorithm represents each image in the database. Then, histograms generated by a real-time video are compared with those in the database. The Euclidean distance between histograms is obtained. The second method uses a Histograms of Oriented Gradients (HOG) algorithm [16], which extracts a 128-dimensional feature vector from each face. Like the previous algorithm, the Euclidean distance between the vectors obtained in real-time and those found in the database is calculated. Both algorithms are run on a Raspberry Pi, which controls a MOSFET switch that activates and deactivates the solenoid that locks the gate. A magnetic sensor tells the system if the door has been closed to relock the solenoid. Also, a system to detect the presence of users nearby is used. This uses an Arduino, a motion sensor (PIR), and a lamp. When the motion sensor sends a signal to the Arduino, it turns on the room lamp. The accuracy of the LBPH method was 98%, while HOG was 96%, in a total of 200 tests for each.

Related work describing the development of a biometric lock based on facial recognition focuses primarily on the interaction of hardware elements and facial recognition software. In the works described above, there is an imbalance in the anti-spoofing issue, since, there are few authors who address this issue to prevent malicious attacks by intruders. In this work, a biometric-lock system is developed that approaches all the necessary aspects for implementing a door security system based on facial recognition, with the advantages of embedded devices, but with an individual focus on facial recognition and an anti-spoofing technique using a Resnet18 + rPPG technique that calculates the heart rate from changes in the user's skin tone.

3. Materials and Methods

Hardware and software complement each other to develop a biometric-lock system (BLS) that allows users to access a restricted area through facial recognition. The development methodology to construct the architecture proposed for the BLS is divided into three modules, as shown in Figure 1. The first module refers to the development of the Facial Recognizer System (*FRS*). It is composed of two blocks: (i) *User Registration* and (ii) *User Authentication*, up to the interaction of the information exchanges that exist between the modules. The second module, named *User and Control Interfaces*, consists of four hardware elements and executes the logic tasks. The third is the electronic module, named *Power Circuit*, for locking and unlocking the access door when receiving a signal. We explain each module in the following subsections.

3.1. Facial Recognizer System FRS

FRS is divided into two main blocks. The *User Registration* block has the purpose of registering the people who can unlock the door to gain access, this task is done by an administrator. On the other hand, the *User Authentication* block is in charge of facial

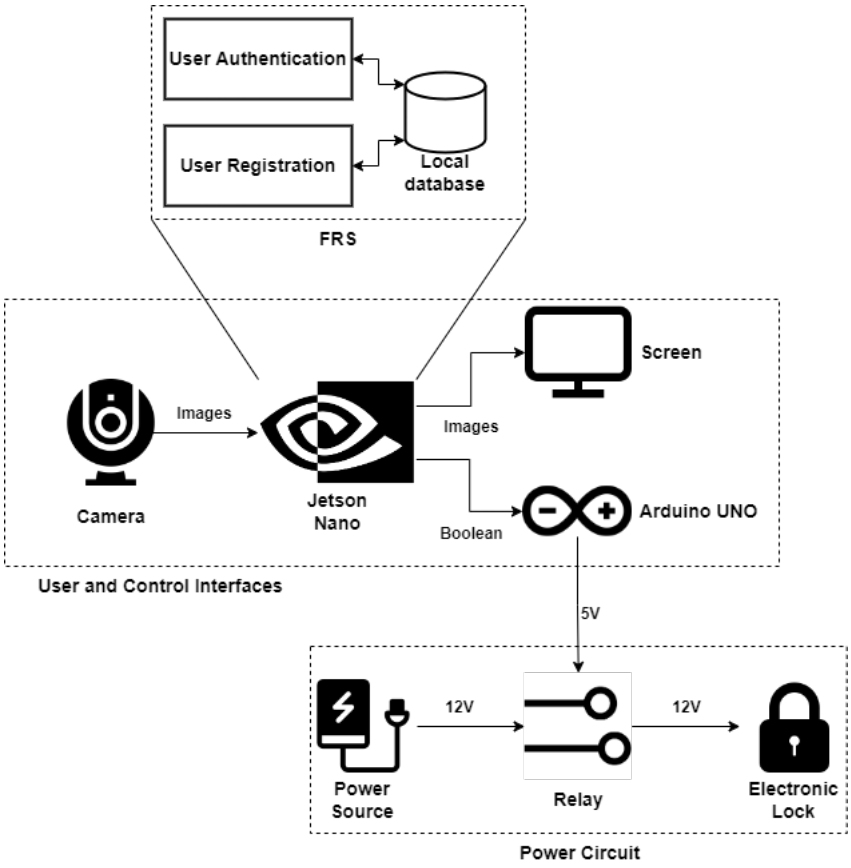


Figure 1. Development methodology to construct the BLS architecture.

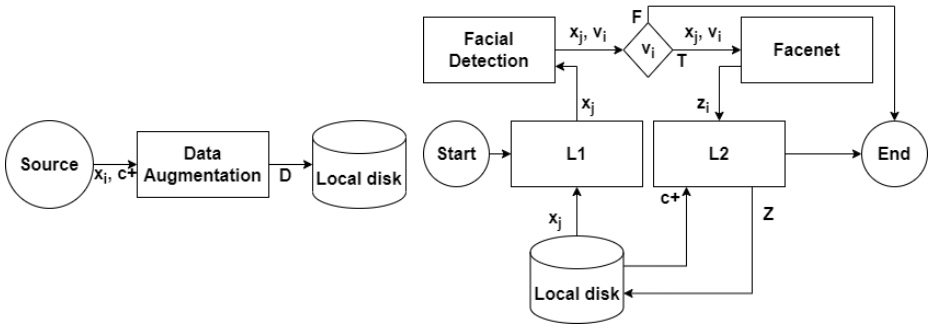
recognition of all persons (users) who show themselves in front of the camera and grant (or not) access to the restricted area.

User registration *User Registration* block registers users that the system will allow access to once they are facially recognized and is divided into two stages that describe the image flow and processing as shown in Figure 2.

As shown in Figure 2-(a), in the first stage, a photograph of the user's face (x_i) is provided frontally by *Source*. Also, enter the user's name ($c+$) by which he/she will be identified.

Before storing the user data (photo and name), two copies of the images are generated in the *Data Augmentation* module. These copies correspond to images derived from the original user's photograph, one with a treatment that brings the user's face closer and the other with a slight change in the angle of the face. In this way, it is sought to cover certain angles or distances of the user in front of the *Camera*. These three images are stored in a directory (D) per user called the user's name. This D is stored in the *Local disk*.

Then, in the second stage, as shown in Figure 2-(b), the *L1* module sends the images extracted from the *Local disk* (x_j) to the *Face Detection* module. *Face Detection* detects



(a) User photo and name capture stage. (b) Feature vector generation and storage.

Figure 2. Extraction of the architecture for the creation of the database.

and extracts faces from an image. This is achieved through models or molds in an XML format called Haar Cascades, which contain features of a frontal face in a numerical data matrix. If one or more regions of the image correspond successfully, these regions are extracted as coordinates within the original image. If an image contains faces, and they are detected, the image and its coordinates (x_j, v_i) are sent to *Facenet* module. This module uses a neural network called FaceNet to extract the characteristics of the detected faces.

CNNs are composed of the feature extraction layer and the learning layer. The feature extraction layer is made up of convolution and reduction (known as pooling) layers, and the learning layer is typically a fully connected neural network. CNNs are designed to process data in multiple arrays, for example, a color image composed of three-dimensional arrays containing pixel intensities in RGB [63].

FaceNet is one of the Deep Learning-based facial recognition applications. It is a one-shot learning algorithm that calculates the similarity distance for each face in Euclidean space [61]. FaceNet [52] was created by Google in 2015, and it is a pre-trained CNN that extracts the main features of faces and transforms them into a 128-dimensional vector known as *embedding*, that represents the principal features extracted from the faces.

Conventional CNNs fail to solve the problem called *one-shot-learning* [12], which is attributed to the inability to do optimal training with only one image per user in the database. In contrast, FaceNet, being a pre-trained CNN with more than 500 million images, according to the authors, achieves a 99.63% effectiveness [52], which is more than the effectiveness reached by the human eye.

FaceNet architecture, as shown in Figure 3, starts from a batch of face thumbnails that works as a Deep Architecture (CNN) input. The Deep architecture transforms the thumbnail image into a 128-D vector. This vector is normalized in L2 process, which results in an Embedding of the face. The *embedding* is followed by Triplet Loss during training.

Facenet module returns to *L2* module a 128-dimensional vector (z_i) corresponding to the features of the face in the image.*L2* also extracts the user's name $(c+)$ and assign its to the z_i. The final result of the images in database treatment is a Pickle file with the extension ".pkl" (Z), which are all the identities of Euclidean positions in a 1x128



Figure 3. FaceNet structure.

dimensions space stored in the *Local disk* module. Pickle is a Python module used to serialize a file on a disk and then deserialize it back into the program at runtime.

In Z, all the feature vectors of the faces available in D are stored. Each time one or more users are stored, all the feature vectors of the users stored in D must be re-calculated.

User authentication This process requires the interaction of five components as shown in Figure 4. It starts with an image (x_i, 160x160 pixels) sent to the *Face Detection* module. The extracted image and coordinates (x_i, v_i) are dispatched to the *Reality Detection* module.

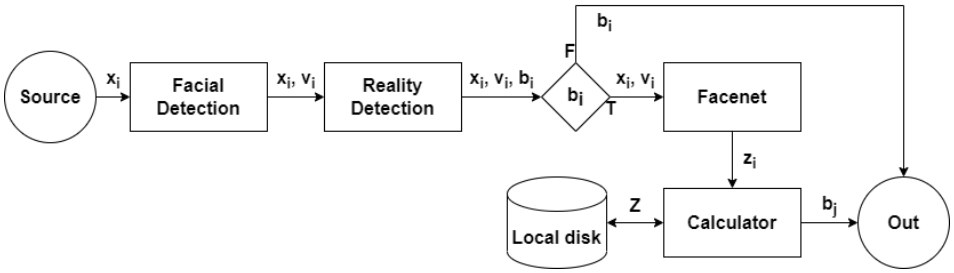


Figure 4. Architecture extraction for facial identification.

Real faces must correspond to real people's faces shown on camera in real-time, while fake faces correspond to attempts to spoof the system using printed or digital photos. In other words, in this module, a classification treatment is carried out with a CNN to the regions obtained from the *Face Detection* module, becoming an anti-spoofing filter for images of non-real people, such as photographs placed in front of the camera.

The *Reality Detection* module uses a typical CNN called ResNet18 that consists of 17 convolution layers and one fully connected network layer, and the activation function used is ReLU [21]. Its purpose is to extract small image regions corresponding to the user's skin parts. Then, remote photoplethysmography (rPPG) treatment detects heart rates and classifies the extracted faces into real or false faces. The rPPG is a heart rate measurement technique that calculates light reflected from the skin, this change is calculated by comparing the reflected light from one frame to the next. The amount of blood in a specific skin area causes capillary dilation and constriction, which causes the light reflected from the skin to change according to the volume of blood circulating [59]. This heart rate measurement method reached a 98.97% test accuracy [56], using a 0.307 M effective pixel camera (ours is 8.08 M effective pixels).

When an image is filtered, and the result is a fake face *Reality Detection* sends a false Boolean (b_i), and the image (x_i) with the face squared in red is displayed on the screen as shown in Figure 7-(c). Otherwise, if the filtered face gives a true result, the *Reality Detection* module sends the original image and its coordinates (x_i, v_i). *Facenet* module is responsible for extracting the features from the faces present in the image and sending a new 128D vector (z_i) to the *Calculator* module. Then, the *Calculator* module calculates the angles from this new vector with all the vectors (Z) extracted from the *Local disk*.

The angles between vectors are calculated by cosine similarity (CS) using the Equation 1 where a_i is the resulting angle, w_i is the feature vector obtained from the new image generated in real-time, and z_i is the representation of the feature vectors stored in *Local disk*. The resulting values range is from 0 to 1 ($a_i \in \mathbb{R}(0, 1)$), where 1 is the biggest angle, and 0 is the smallest. Figure 5 shows an example of a comparison between two vectors, the separation angle and the defined threshold.

$$a_i(w_i, z_i) = 1 - \frac{w_i \cdot z_i}{||w_i||_2 \cdot ||z_i||_2} \tag{1}$$

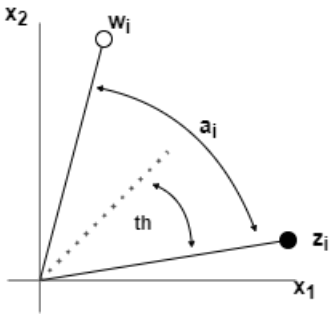


Figure 5. Cosine similarity for comparison of two data vectors.

At this point, if the smallest angle is under a threshold (th), the *Calculator* module extracts the name ($c+$) of the most similar vector, which is a person's name. Otherwise, if the smallest angle is not below the th, the name "Unknown" is assigned. Similarly, if two or more vectors obtain the same CS and are below the th, "Unknown" is assigned.

We implemented a threshold of 0.30 based on the need to ensure greater similarity between the feature vectors of the detected faces, which helps to strengthen the system's security and reduce the risk of confusion between users. We pointed out that decreasing the similarity threshold raises the level of features required for a face to be recognized as valid, resulting in a decrease in False Positive cases.

In contrast, in [53], a th of 0.40 was calculated using the C4.5 [48] algorithm, obtaining a Recall of 96.42, which indicates the presence of False Positive cases. By adjusting the threshold to 0.30 in our work, we seek to mitigate this problem and reduce the probability of misidentifications, prioritizing the security of the system.

It is essential to note that by lowering the threshold, it is possible to increase the rate of False Negatives, i.e., cases in which a legitimate user may be erroneously rejected.

However, in the context of security, it is preferable to go inaccurate in denying access to a legitimate user than to allow access to an intruder. Therefore, by sacrificing some False Negatives in favor of reducing False Positives, we are strengthening the integrity and reliability of the facial recognition system.

Figure 6 shows a simulation of the CS obtained of a user's face taken in real-time (w_i) compared to each of the previously recorded users (z_n). Because z_2 has the smallest angle and is below the th, the output is assigned the z_2 identity.The CS calculation process takes one millisecond (0.001 seconds) on average per registered user, i.e., 1-300 registered users have a minimal impact on the system response time.

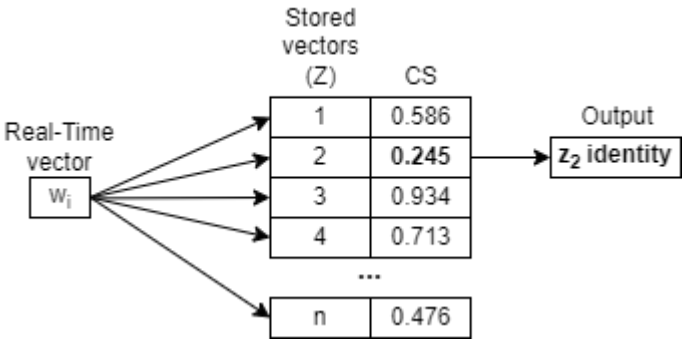


Figure 6. Simulation of angle calculation between vectors for identity assignment.

Finally, the output generated by the *Calculator* module generates a new image created with the same input image (x_i). Still, the detected person's face is framed with a green square with its name on it and displayed on the *Screen* as shown en Figure 7-(a). Also, a Boolean (b_i) is sent to an *Arduino UNO* to unlock the door in case the facial recognition is successful. Otherwise, if the result is unsuccessful, the image of the unrecognized person's face will be displayed on the screen in a red square with the word "Unknown" as shown in Figure 7-(b).



(a) Successful facial recognition in Screen. (b) Unsuccessful facial recognition in Screen. (c) Spoof try rejected in Screen.

Figure 7. Possible results showed in Screen.

3.2. User and Control Interfaces

This module is responsible for performing the image acquisition and processing tasks as well as the delivery of results that can be positive or negative for the unlocking or blocking of the door, respectively. The *Jetson Nano* microcomputer, the *Camera*, and the *Screen* are three key system interfaces due to the importance of their tasks. A cabinet was designed and machined to protect and fix these interfaces to contain them in a strategic position for their functions.

The *Camera* acquires images for two elementary purposes in the system. First, it takes photographs to add users to the database used in the *User Registration* process, and second, it captures the users' images that will be facially identified by the *User Authentication* process.

A microcomputer *Jetson Nano* runs the FRS. It receives the images from the *Camera* and sends Boolean signals to an *Arduino UNO* as an output. This device executes the looking and unlocking task sending signals to the *Power Circuit* module. This task is delegated to the *Arduino UNO* to reduce the computational overhead generated on the *Jetson Nano*. The *Screen* receives the output images from the *Jetson Nano*, and it works as visual support so the users can see their faces.

The devices descriptions are:

- *Camera*: IMX219 is a 4.60mm diagonal CMOS (Type 1/ .0) active pixel type image sensor with a square pixel matrix and 8.08M effective pixels. This chip is powered by three power supplies, 2.8V analog, 1.2V digital, and 1.8V IF, and has low power consumption [3].
- *Jetson Nano*: 472 GFLOPS of computing performance with a quad-core 64-bit ARM CPU and an integrated 128-core NVIDIA GPU. It also packs 4GB LPDDR4 memory in an efficient low-power package with 5W / 10W power modes and 5V DC input [39].
- *Screen*: 7 inches touch IPS Screen connected via HDMI, 1024x600 hardware resolution.USB protocol translator, converting the touch signal into standard multi-point touch protocol to achieve smooth multi-point touch control. Powered with 5V via micro-USB.
- *Arduino UNO*: is a development board based on ATmega328P, with 14 terminals for data input or output. It works with a 16 MHz resonator, USB/Power jack connections [4].

3.3. Power Circuit

Since the *Arduino UNO* device is capable of sending electrical signals in the range of 0V to 5V and the *Electronic Lock* is powered with 12V, a *Power Circuit* module is needed to control the current flow required by the *Electronic Lock* with the *Arduino UNO* signals. This *Power Circuit* is composed of three electronic devices: a *Relay*, a *Power Source*, and the aforementioned *Electronic Lock*.

The facial recognition system will send a true or false value depending on the result of the user's identification. When the *Arduino UNO* receives a true value, it sends a 5V

signal and activates the *Relay*. This device acts as a "switch", i.e., when the *Relay* receives the signal, it activates and allows the voltage flow (12V) from the *Power Source* to the *Electronic Lock*. The *Electronic Lock* is a counter-plate that unlocks when receiving a 12V signal. This counter plate is located on the door frame. The door lock prevents the door from opening until the counter plate is energized and allows manipulation.

- *Electronic Lock*: is a fixed device that, upon receiving an electrical signal, its opening mechanism unlocks to allow a door to open.
- *Relay*: is an electronic device that distributes an electrical signal in two directions (A or B). When the relay is in a normal state, it sends the signal in direction A, but when it is activated, it sends the signal in direction B.
- *Power Source*: is an electronic device that transforms electrical energy from alternating current into direct current. It is commonly used to transform ordinary household electrical energy (120 volts AC) into electrical energy for charging mobile devices (5/12/24 volts direct current).

4. Testing and results

The Facial Recognition System (FRS) was developed, compiled, and executed in Python language. It uses Tensorflow-GPU and Keras libraries to implement CNNs for reality detection and facial recognition (ResNet18 and FaceNet). In the same way, the "switch" process executed by the Arduino UNO was developed, compiled, and executed in C++.

Three stages of FRS testing were performed: (i) face recognition testing with artificial lighting, (ii) face recognition testing with natural lighting, and finally, (iii) reality detection testing using printed and digital photographs and videos.

The individuals considered for testing the system represent a mixed sample of Mexican people aged 14 to 40. Approximately 90% of the sample is in the 18 to 23 age range. The skin tones of the individuals are diverse, ranging from tan to lighter skin tones.

The first testing stage was performed with artificial lighting on 85 users, 72 registered users, and 13 non-registered users. The tests were conducted with illumination on the top of the users' heads to avoid shadows that would hinder facial recognition. The population of test users registered 425 trials (5 for each user) to the FRS. From 360 registered-users trials, the system successfully recognized 321 times (TP, True Positive) and 39 times failed to identify the person (FN, False Negative), giving the result of "Unknown". No identity assignment errors were recorded, i.e., the system did not confuse any user (registered or not) with another (FP, False Positive). Within the controlled environment testing stage, 13 users who were not registered in the system were tested. The total number of attempts was 65. 100% of the cases were identified as "U", i.e., of all the cases predicted as "Unknown", the real result the system gave was the same (TN, True Negative). Table 1 shows the confusion matrix resulting from the tests in a controlled environment.

Tables 2 to 6 show the relationship matrices related to face recognition performed in a controlled environment on 72 users (S) registered in the system. User number 23 was recognized only once by the FRS, and 4 obtained a "U" (Unknown) result. During the execution of the tests, different variables were observed that influenced the facial identification of some test subjects, e.g., face pose, glasses, or fringe. If any of these variables

Table 1. Resulting confusion matrix for the controlled environment testing stage.

		Predicted	
		Positive	Negative
Real	Positive	321	0
	Negative	39	65

changed from the initial sampling, facial recognition showed delays and failed identifications. Therefore, as users attempted facial recognition and failed to make a successful identification, they were asked to change these variables to cover the issues surrounding the face presentation to the system.

Table 2. Relationship matrix of face recognition results with artificial lighting to registered users (1/5)

		Predicted															
		S1	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**	**S8**	**S9**	**S10**	**S11**	**S12**	**S13**	**S14**	**S15**	**S16**
Real	**S1**	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	S2	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	S3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	S4	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0
	S5	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0
	S6	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0
	S7	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0
	S8	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0
	S9	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0
	S10	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0
	S11	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0
	S12	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0
	S13	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0
	S14	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0
	S15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0
	S16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
	“U"	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	3

Table 3. Relationship matrix of face recognition results with artificial lighting to registered users (2/5)

		Predicted															
		S17	**S18**	**S19**	**S20**	**S21**	**S22**	**S23**	**S24**	**S25**	**S26**	**S27**	**S28**	**S29**	**S30**	**S31**	**S32**
Real	**S17**	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	S18	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	S19	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0
	S20	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0
	S21	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0
	S22	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0
	S23	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
	S24	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0
	S25	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0
	S26	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0
	S27	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0
	S28	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0
	S29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	S30	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0
	S31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0
	S32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5
	“U"	0	0	0	0	0	0	4	0	0	0	0	0	5	0	0	0

Table 4. Relationship matrix of face recognition results with artificial lighting to registered users (3/5)

		Predicted															
		S33	**S34**	**S35**	**S36**	**S37**	**S38**	**S39**	**S40**	**S41**	**S42**	**S43**	**S44**	**S45**	**S46**	**S47**	**S48**
Real	**S33**	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	S34	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	S35	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0
	S36	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0
	S37	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0
	S38	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0
	S39	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0
	S40	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0
	S41	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
	S42	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
	S43	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0
	S44	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0
	S45	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0
	S46	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0
	S47	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0
	S48	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5
	“U"	0	0	0	0	1	0	0	0	4	4	0	0	0	0	0	0

Table 5. Relationship matrix of face recognition results with artificial lighting to registered users (4/5)

		Predicted															
		S49	**S50**	**S51**	**S52**	**S53**	**S54**	**S55**	**S56**	**S57**	**S58**	**S59**	**S60**	**S61**	**S62**	**S63**	**S64**
Real	**S49**	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	S50	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	S51	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0
	S52	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0
	S53	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0
	S54	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0
	S55	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0
	S56	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0
	S57	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0
	S58	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0
	S59	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0
	S60	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0
	S61	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0
	S62	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0
	S63	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0
	S64	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5
	“U"	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0

Table 6. Relationship matrix of face recognition results with artificial lighting to registered users (5/5)

		Predicted							
		S65	**S66**	**S67**	**S68**	**S69**	**S70**	**S71**	**S72**
Real	**S65**	0	0	0	0	0	0	0	0
	S66	0	5	0	0	0	0	0	0
	S67	0	0	5	0	0	0	0	0
	S68	0	0	0	4	0	0	0	0
	S69	0	0	0	0	5	0	0	0
	S70	0	0	0	0	0	5	0	0
	S71	0	0	0	0	0	0	0	0
	S72	0	0	0	0	0	0	0	5
	“U"	5	0	0	1	0	0	5	0

The faces of the users with the highest number of failed attempts showed a trend in the images captured during the registration stage. For example, the presence of severe occlusions in the eye area is notorious because the users had very thick eyebrows, wore glasses and/or wore very pronounced bangs. Another factor detected in the registration images is the darkening of the mouth area due to pronounced commissures. These factors produce occlusions and shadows in critical areas of the user's face during registration, thus losing essential features of the user's face and making it difficult to authenticate users during testing is shown in Tables 2 to 6. Similarly, make-up influences the recognition of

some users, i.e., if the user presents different make-up conditions during registration and authentication, the BLS will tend to fail due to changes in facial features.

The second testing stage was performed outdoors under natural light conditions during sunset. Table 7 shows the results obtained from testing 10 test users registered in the system between 19:30 and 19:45 GMT-6. It is a time when the sun begins to set, and the amount of light decreases considerably but without reaching total darkness. Testing outdoors is to add a variable related to the amount of light available for face recognition and reality detection.

At the end of this stage, the system failed to recognize 6 out of 50 attempts. Also, within this stage, 25 tests were performed on five subjects not registered in the system. As in the previous stage, the system did not recognize unregistered subjects. Table 7 shows the 44 facial recognition hits (TP), 25 successes in non-face recognition (TN), six unsuccessful recognitions (FN), and 0 identity assignment errors (FP) obtained.

Table 7. Resulting confusion matrix for the uncontrolled environment testing stage

		Predicted	
		Positive	Negative
Real	Positive	44	0
	Negative	6	25

Table 8 shows the relationship matrix of the tests performed on users with natural lighting.

Table 8. Relationship matrix of face recognition results with natural lightning to registered subjects

		Predicted									
		S73	**S74**	**S75**	**S76**	**S77**	**S78**	**S79**	**S80**	**S81**	**S82**
Real	**S73**	5	0	0	0	0	0	0	0	0	0
	S74	0	5	0	0	0	0	0	0	0	0
	S75	0	0	4	0	0	0	0	0	0	0
	S76	0	0	0	5	0	0	0	0	0	0
	S77	0	0	0	0	5	0	0	0	0	0
	S78	0	0	0	0	0	2	0	0	0	0
	S79	0	0	0	0	0	0	5	0	0	0
	S80	0	0	0	0	0	0	0	3	0	0
	S81	0	0	0	0	0	0	0	0	5	0
	S82	0	0	0	0	0	0	0	0	0	5
	“U"	0	0	1	0	0	3	0	2	0	0

As in the previous testing stage, the users who presented more errors in the testing stage, as shown in Table 8, demonstrated more shadows and occlusions on the face in the capture of log images.

As the final results of the facial recognition tests in the first two stages, Table 9, 365 hits in face recognition (TP), 45 system failures in confusing registered subjects with un-

known subjects (FN), 90 hits in non-recognition of non-registered subjects (TN), and 0 mistaken identity assignments to registered or non-registered subjects (FP) can be observed.

Table 9. Resulting confusion matrix from the two face recognition stages of testing

		Predicted	
		Positive	Negative
Real	Positive	365	0
	Negative	45	90

Once the two stages of face recognition testing were completed, results were obtained for the evaluation criteria: false rejection rate (FRR), false acceptance rate (FAR), accuracy, and F1. FRR (Equation 2) is a binary classification measure that provides an evaluation of how well the system avoids false negatives (FN). FAR (Equation 3) evaluates how well false positives are avoided. Both measures generate values between 0 and 1; the closer the result is to 0, the better the system evaluation. Accuracy is the most widely used measurement in classification systems. It is calculated by dividing the total number of hits by the total number of samples, i.e., accuracy is the percentage of hits in the system (Equation 4).

$$FRR \text{ (False rejection rate)} = 1 - TP/(TP + FN) \tag{2}$$

$$FAR \text{ (False acceptation rate)} = 1 - TN/(TN + FP) \tag{3}$$

$$Accuracy = (TP + TN)/(TP + FP + FN + TN) \tag{4}$$

The evaluation criteria described above were applied to the results corresponding to the face recognition tests in the two stages shown in Tables 1, 7, and 9. The evaluations obtained are summarized in Table 10.

Table 10. FRS evaluation criteria results of tests in the face recognition stages

		Evaluation criteria		
		FRR	FAR	Accuracy
Conditions	Artificial lightning	0.1083	0.0	0.9082
	Natural lightning	0.1200	0.0	0.9200
	Average	0.1098	0.0	0.9100

To verify that the system is robust to different types of lighting (natural or artificial), a paired T-Student [25] distribution parametric analysis was performed. We proposed the next hypotheses:

- H_0*: No significant difference exists in tests performed with natural and artificial illumination.*
- H_1*: There is a significant difference in tests performed with natural and artificial illumination.*

A random sampling of 10 users from the artificial lighting test stage was generated for comparison. Using the individual effectiveness obtained for each user (20% per successful recognition) showed in Tables 2 to 6 and Table 8, the following data were obtained: mean difference $\bar{x} = -6$, sample size $n = 10$, standard deviation $S_d = 32.73$. The degrees of freedom ($d_f = 9$) and significance level ($\alpha = 0.05$) are considered for the parametric analysis. The corresponding T-Student statistic value is $t_\alpha : 0.005$, $df : 9 = 2.2622$. Equation 5 was used and $t = -0.5797$ was obtained.

$$t = \frac{\bar{x}}{\frac{S_d}{\sqrt{n}}} = \frac{-6}{\frac{32.73}{\sqrt{9}}} = -0.5797 \tag{5}$$

Figure 8 shows that the t-value (statistic) obtained is within the confidence interval. H_0 is accepted, i.e., no significant difference exists between the tests performed with artificial and natural lighting. Because of this, the system is robust to changes in illumination from natural to artificial or vice versa.

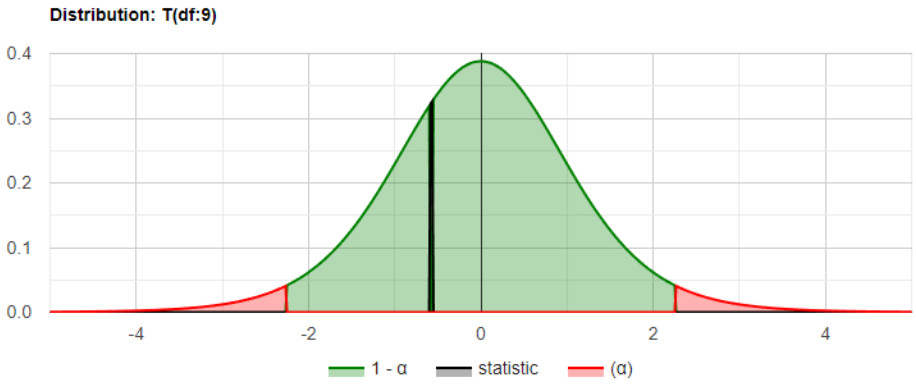


Figure 8. Paired T-Student distribution parametric analysis graph

In the last testing stage, 90 photographs and ten videos were submitted to the FRS. None of the attempts were successful in spoofing the system (TN). In addition, the system was recognized as real people in all 500 attempts in the previous two stages of testing (TP). Thus, as shown in Table 11, the *Reality Detection* module obtained 500 TP and 100 TN.

Table 11. Resulting confusion matrix for the reality detection testing stage

		Predicted	
		Positive	Negative
Real	Positive	500	0
	Negative	0	100

4.1. Discussion

In Table 12, we compared different works considering hardware, techniques involved for facial recognition, and the performance results obtained.

For the hardware comparison, we show that the control interfaces of [15,26,41,51,60] are based on different Raspberry Pi models. Some advantages of this device are its low cost, its small size, which makes it easy to hide inside the room and its ease of use. On the other hand, [35] and our proposal are based on Jetson Nano, which, unlike any Raspberry Pi model, has an integrated GPU that accelerates the execution of algorithms. However, it is larger in size and higher in cost. The devices used as power circuitry vary in all cases but demonstrate a similarity of operation in power management. Hardware costs vary in two price ranges, from $45 to $60, and from $225 to $250 USD. This is due to the incorporation of a Neural Compute Stick 2 ([41]) and the integrated GPU in the Jetson Nano ([35], ours), which provide additional support for image data processing and improved response speed.

In [35,41,60] and our proposal, pre-trained One-Shot-Learning MobileNet and FaceNet networks were used for face recognition. The main advantage of implementing this type of network is saving training time that is usually required to properly input a neural network. In addition, FaceNet was trained with more than 500 million images, while MobileNet with 3 million. In the remaining cases, they trained their own models for facial identification, and being less dense, they require less computational power and generally less execution time. While in the proposals [15,26,51], it is necessary to retrain the facial identification models each time a new user is required to register. This is because the models need to learn to classify each of the classes created.

However, [26, 35, 60] exceed 96.00% in terms of successful recognitions. It is necessary to remark that these proposals do not have an anti-spoofing stage, which is totally inconvenient for face recognition tasks in security despite having a good response time ([35]). [15] shows an effectiveness rate of 92.68%, but due to its anti-spoofing method, it requires more than 10 seconds to open a door and an additional action (blink) by the user. Our proposal is based on a CNN ResNet18 that makes image extractions corresponding to parts of the skin and an rPPG algorithm that calculates the person's heart rate by the change of illumination on the skin caused by blood flow. This means that a printed or digital image will not cause such illumination changes in the skin.

Most of the work tested in artificially lit areas, only [26] demonstrated an average effectiveness of 97.5% in daylight conditions at four different times. [35] demonstrated an effectiveness of 97.7% and a response time of 0.47 s, making it competitive for face recognition tasks that require high response speed and do not require an anti-spoofing system. The FAR demonstrated in [41] was 35.71%, which is not suitable for face recognition systems. It means the system allowed access to 35 unauthorized people out of 100.

Our proposal demonstrated an average effectiveness of 91% and a response speed of 1.68s, which competes with the related work described above. The resulting average FRR was 1.098%, corresponding to the 45 false-negative results, which denied access to users registered in the system. In addition, a FAR of 0% represents non-existent false positives. This shows that the system errors consist solely of false rejections. Of these, 95.12% of the subjects succeeded in gaining access on at least one attempt. The system recognized most of the users who were initially not recognized in their first attempt, after a small adjustment of the face position with respect to the camera to reduce shadows, center the

Table 12. Comparison of hardware, software, and performance of related work and the proposal

		References							
		[41]	[35]	[26]	[60]	[15]	[51]		**Proposed**
Hardware	Procesor	Raspberry + NCS2 $245	Jetson Nano $225	Raspberry Pi $45	Raspberry Pi 4 $61	Raspberry Pi 3 B+ $50	Raspberry Pi $45		Jetson Nano $225
	Power circuit	Coil + Electro-magnetic Lock	Reed Switch	Relay + Solenoid	Servo-motor	Relay + Solenoid	MOSFET + Solneoid		Relay + Electronic Lock + Power Source
Software	Recognition	MobileNet	FaceNet	CNN	ResNet + FaceNet	HOG + SVM	LBPH	HOG	FaceNet
	Liveness	SSD	-	-	-	Blink detection	-	-	ResNet18 + rPPG
Performance	Recognition tests	80	1260	120	230	-	200	200	500
	Reality tests	5	-	-	-	-	-	-	600
	Recognition accuracy	88.75%	97.70%	Morning = 96.67% Afternoon = 96.67% Evening = 96.67% Night = 100% AVG = 97.5%	87.36%	92.68%	98.00%	96.00%	CE = 90.82% UE = 92.00% AVG = 91.00%
	FRR	0%	-	0%	-	-	4%	2%	CE = 1.083% UE = 1.2% AVG = 1.098%
	FAR	35.71%	-	Morning = 6.25% Afternoon = 6.25% Evening = 6.25% Night = 0% AVG = 4.76%	-	-	0%	6%	0%
	Reality accuracy	100%	-	-	-	-	-	-	100%
	Execution	-	0.47 s	-	-	10.196 s	-	-	1.68 s

face in the image, among other aspects that could affect the recognition. But, we remark that it does not grant access to unregistered users and does not confuse one user with another. In this work, numerous tests on both aspects of user authentication (recognition and reality) demonstrate that the system shows a balance of effectiveness, FAR, FRR, and speed of response.

Table 13. Top selled Amazon facial recognition based door locks

Model	Price (USD)
Necchuizo	$365
We Technology	$249
NGP	$399

The Table 13 shows the top three best-selling commercial door locking devices with facial recognition, according to Amazon.com in the US. They consist of electronic devices that completely replace the door lock in question. In none of the cases are technical specifications of effectiveness, FAR, or FRR given. To operate, they require batteries every so often. Being all-in-one devices, they are challenging to maintain in case any of the com-

ponents fail. Likewise, they require mobile applications to control their user registration and deletion operations, including WiFi connectivity.

Modulating the hardware provides high maintainability to the BLS because each device works individually and can be easily repaired or replaced. Also, this modulation facilitates installation or change of location. On the other hand, there are similar commercial locks on the market that achieve better aesthetics and smaller size, but because of this, they increase the difficulty of maintenance and location changes. In addition, these locks operate with disposable or rechargeable batteries, which is a disadvantage if replacement is omitted in a timely manner. Finally, most commercial locks have a registration limit of 100 users, while FaceNet was trained with 10000 identities.

5. Conclusions

This paper presents a proposal for a biometric-lock system (BLS) that works as a facial recognition lock. This system limits access to restricted areas by unlocking doors to users previously registered as authorized persons.

We divide the development of this BLS into three main blocks. The first one refers to the user and control interfaces, which are the devices used to interact with the user and perform facial recognition tasks. The second block is composed of three devices, a relay that controls the voltage flow, a power supply that generates the required voltage, and an electronic lock that locks or unlocks the door opening. Finally, the third block describes the development of the software for the registration and authentication of the users that will interact with the system.

The biggest challenge during the development of the BLS was to produce facial recognition software that is robust to photo and/or video attacks without the use of specialized hardware and that is capable of running on embedded devices. Identifying real people from non-real people is an extremely important feature as it represents a significant security aspect of facial recognition systems. Also, it is important to mention that there are some works in the state of the art that achieve these objectives using different techniques (blink detection, human detection). However, these techniques require actions by the user, increasing their difficulty of use, and this causes a considerable increase in response time. Because our proposal measures the heart rate at the same image of the user's face, it decreases user cooperation and response time.

Tests performed on the BLS show an effectiveness of 90.82% in artificial light and 92.00% in natural light. A T-Student test was performed to verify if there was a significant difference between the two lighting conditions. The results indicate the system's resilience to changes in illumination, showcasing no significant disparities. On average, it achieved an effectiveness rate of 91.00% across all facial recognition trials, with the remaining 9% representing instances where user identification faltered. Notably, the system exhibited a flawless False Acceptance Rate (FAR) of 0%, ensuring zero false positives or user confusion. Furthermore, in a notable statistic, 78 out of 82 users (95.12%) successfully gained access after at least one of five attempts, demonstrating the system's ability to recognize users even if initial recognition fails by adjusting the facial positioning relative to the camera. Also, a 100% effectiveness rate was obtained in the anti-attack tests on the system regarding the test carried out in the experiment. Finally, the average response time obtained was 1.68 seconds. Nevertheless, an average FRR of 1.098% was

obtained since some users presented occlusions in important face areas, such as the eyes and mouth. These factors represent a future challenge to reduce the FRR and grant access to registered users.

The results showed the competitiveness of the proposed BLS compared to the works presented in the state of the art. However, it does not cover the security scenarios required by maximum security buildings; banks, vaults, or prisons. In other words, the BLS has the necessary effectiveness and speed to be implemented in medium security restricted access locations such as offices, classrooms, houses, among others. The commercial devices shown in Table 13 are generally put to the same use. Although these devices currently on the market have significantly decreased in price, the modularization of our proposal could considerably improve the cost-benefit of operating two or more locks with the same Jetson Nano.

In future work, we will focus on improving the effectiveness of facial recognition by implementing different image resolutions, modifying the thresholds for facial identification, or using other CNNs for image processing, such as MobileNet, VGGFace, and DeepFace. Also, different actuators for door opening will be implemented to cover more applications; magnetic locks, servo-motors, and pneumatic or hydraulic pistons, among others. Finally, increase the number of doors operated by the same device to measure operability.

References

1. Ali, W., Tian, W., Din, S.U., Iradukunda, D., Khan, A.A.: Classical and modern face recognition approaches: a complete review. Multimedia tools and applications 80, 4825–4880 (2021)
2. Arashloo, S.R., Kittler, J.: Efficient processing of mrfs for unconstrained-pose face recognition. In: 2013 IEEE sixth international conference on biometrics: theory, applications and systems (BTAS). pp. 1–8. IEEE (2013)
3. ArduCam: Camera sony imx219, https://www.arducam.com/product/arducam-raspberry-pi-camera-v2-8mp-ixm219-b0103/, web accessed in: 07-06-2022
4. Arduino: Arduino uno r3, https://docs.arduino.cc/hardware/uno-rev3, web accessed in: 07-06-2022
5. Augusto, J.C., McCullagh, P.: Ambient intelligence: Concepts and applications. Computer Science and Information Systems 4(1), 1–27 (2007)
6. Bhatt, H.S., Bharadwaj, S., Singh, R., Vatsa, M.: Recognizing surgically altered face images using multiobjective evolutionary algorithm. IEEE Transactions on Information Forensics and Security 8(1), 89–100 (2012)
7. Bud, A.: Facing the future: The impact of apple faceid. Biometric technology today 2018(1), 5–7 (2018)
8. Chen, S., Shen, J., You, X., Chen, J., Yu, C.: A dynamic cryptography door lock system based on visible light communication. In: 2018 23rd Opto-Electronics and Communications Conference (OECC). pp. 1–2 (2018)
9. Cui, Z., Li, W., Xu, D., Shan, S., Chen, X.: Fusing robust face region descriptors via multiple metric learning for face recognition in the wild. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3554–3561 (2013)
10. Das, P.K., Hu, B., Liu, C., Cui, K., Ranjan, P., Xiong, G.: A new approach for face anti-spoofing using handcrafted and deep network features. In: 2019 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI). pp. 33–38. IEEE (2019)

11. Ding, C., Tao, D.: Robust face recognition via multimodal deep face representation. IEEE transactions on Multimedia 17(11), 2049–2058 (2015)
12. Eloff, R., Engelbrecht, H.A., Kamper, H.: Multimodal one-shot learning of speech and images. In: ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 8623–8627 (2019)
13. Gałka, J., Masior, M., Salasa, M.: Voice authentication embedded solution for secured access control. IEEE Transactions on Consumer Electronics 60(4), 653–661 (2014)
14. Ganjoo, R., Purohit, A.: Anti-spoofing door lock using face recognition and blink detection. In: 2021 6th International Conference on Inventive Computation Technologies (ICICT). pp. 1090–1096. IEEE (2021)
15. Ganjoo, R., Purohit, A.: Anti-spoofing door lock using face recognition and blink detection. In: 2021 6th International Conference on Inventive Computation Technologies (ICICT). pp. 1090–1096 (2021)
16. Ghorbani, M., Targhi, A.T., Dehshibi, M.M.: Hog and lbp: Towards a robust face recognition system. In: 2015 Tenth International Conference on Digital Information Management (ICDIM). pp. 138–141 (2015)
17. González-Jiménez, D., Alba-Castro, J.L.: Toward pose-invariant 2-d face recognition through point distribution models and facial symmetry. IEEE Transactions on Information Forensics and Security 2(3), 413–429 (2007)
18. Goud, K.N., Sindhuri, K.: Enhanced Security for Smart Door Using Biometrics and OTP, pp. 517–526. Springer International Publishing, Cham (2022), https://doi.org/10.1007/978-3-030-96634-8_47
19. Grassi, P.A., Garcia, M.E., Fenton, J.F.: NIST special publication 800-63b: Digital identity guidelines. Tech. Rep. 800-63B, National Institute of Standards and Technology (June 2017), https://doi.org/10.6028/NIST.SP.800-63B
20. Hafez, S.F., Selim, M.M., Zayed, H.H.: 2d face recognition system based on selected gabor filters and linear discriminant analysis lda. arXiv preprint arXiv:1503.03741 (2015)
21. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
22. Hearst, M., Dumais, S., Osuna, E., Platt, J., Scholkopf, B.: Support vector machines. IEEE Intelligent Systems and their Applications 13(4), 18–28 (1998)
23. Hemalatha, A., Gandhimathi, G.: Rfid, password and otp based door lock system using 8051 microcontroller. International Journal of Engineering Research and Technology 7(11), 1–6 (2019)
24. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications (2017), https://arxiv.org/abs/1704.04861
25. Hsu, H., Lachenbruch, P.A.: Paired t test. Wiley StatsRef: statistics reference online (2014)
26. Irjanto, N.S., Surantha, N.: Home security system with face recognition based on convolutional neural network. (IJACSA) International Journal of Advanced Computer Science and Applications 11(11) (2020)
27. Jayamaha, R.G.M.M., Senadheera, M.R.R., Gamage, T.N.C., Weerasekara, K.D.P.B., Dissanayaka, G.A., Kodagoda, G.N.: Voizlock - human voice authentication system using hidden markov model. In: 2008 4th International Conference on Information and Automation for Sustainability. pp. 330–335 (2008)
28. Jovanović, B., Milenković, I., Bogićević Sretenović, M., Simić, D.: Extending identity management system with multimodal biometric authentication. Computer Science and Information Systems/ComSIS 13(2), 313–334 (2016)
29. Khowaja, S.A., Lee, S.L.: Hybrid and hierarchical fusion networks: a deep cross-modal learning architecture for action recognition. Neural Computing and Applications 32(14), 10423–10434 (2020)

30. Khowaja, S.A., Lee, S.L.: Semantic image networks for human action recognition. International Journal of Computer Vision 128(2), 393–419 (2020)
31. Komol, M.M.R., Podder, A.K., Ali, M.N., Ansary, S.M.: Rfid and finger print based dual security system: A robust secured control to access through door lock operation. American Journal of Embedded Systems and Applications 6(1), 15–22 (2018)
32. Kortli, Y., Jridi, M., Al Falou, A., Atri, M.: Face recognition systems: A survey. Sensors 20(2) (2020), https://www.mdpi.com/1424-8220/20/2/342
33. Kumar, A., Ravikanth, C.: Personal authentication using finger knuckle surface. IEEE Transactions on Information Forensics and Security 4(1), 98–110 (2009)
34. Lewandowska, M., Nowak, J.: Measuring pulse rate with a webcam. Journal of Medical Imaging and Health Informatics 2(1), 87–92 (2012)
35. Lindner, T., Wyrwal, D., Bialek, M., Nowak, P.: Face recognition system based on a single-board computer. In: 2020 International Conference Mechatronic Systems and Materials (MSM). pp. 1–6 (2020)
36. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) Computer Vision – ECCV 2016. pp. 21–37. Springer International Publishing, Cham (2016)
37. Mian, A., Bennamoun, M., Owens, R.: An efficient multimodal 2d-3d hybrid approach to automatic face recognition. IEEE transactions on pattern analysis and machine intelligence 29(11), 1927–1943 (2007)
38. Nortje, L.: Direct and indirect multimodal few-shot learning of speech and images. Ph.D. thesis, Stellenbosch: Stellenbosch University (2020)
39. NVIDIA: Jetson nano developer kit, https://developer.nvidia.com/embedded/jetson-nano-developer-kit, web; accessed in 01-29-2021
40. Organization, U.N.: Unodc burglary 2018. https://dataunodc.un.org/data/crime/burglary, accessed: 2022-03-23
41. Orna, G., Benitez, D.S., Perez, N.: A low-cost embedded facial recognition system for door access control using deep learning. In: 2020 IEEE ANDESCON. pp. 1–6 (2020)
42. Pacheco, J., Miranda, K.: Design of a low-cost nfc door lock for a smart home system. In: 2020 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS). pp. 1–5 (2020)
43. Patil, B., Mahajan, V., Suryawanshi, S., Pawar, M.: Automatic door lock system using pin on android phone. International Research Journal of Engineering and Technology (IRJET) 5(11), 1007–1011 (2018)
44. Pillai, J.K., Patel, V.M., Chellappa, R., Ratha, N.K.: Secure and robust iris recognition using random projections and sparse representations. IEEE transactions on pattern analysis and machine intelligence 33(9), 1877–1893 (2011)
45. Prasad, S., Govindan, V., Sathidevi, P.: Palmprint authentication using fusion of wavelet and contourlet features. Security and Communication Networks 4(5), 577–590 (2011)
46. Prity, S.A., Afrose, J., Hasan, M.: Rfid based smart door lock security system. American Journal of Sciences and Engineering Research E-ISSN-2348-703X 4(3) (2021)
47. Queirolo, C.C., Silva, L., Bellon, O.R., Segundo, M.P.: 3d face recognition using simulated annealing and the surface interpenetration measure. IEEE transactions on pattern analysis and machine intelligence 32(2), 206–219 (2009)
48. Quinlan, J.R.: Improved use of continuous attributes in c4. 5. Journal of artificial intelligence research 4, 77–90 (1996)
49. Ranjan, R., Sankaranarayanan, S., Castillo, C.D., Chellappa, R.: An all-in-one convolutional neural network for face analysis. In: 2017 12th IEEE International Conference on Automatic Face Gesture Recognition (FG 2017). pp. 17–24 (2017)
50. Ren, J., Jiang, X., Yuan, J.: Relaxed local ternary pattern for face recognition. In: 2013 IEEE international conference on image processing. pp. 3680–3684. IEEE (2013)

51. Saputra, R., Surantha, N.: Smart and real-time door lock system for an elderly user based on face recognition. Bulletin of Electrical Engineering and Informatics 10(3), 1345–1355 (2021), https://beei.org/index.php/EEI/article/view/2955
52. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: A unified embedding for face recognition and clustering. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE (2015), http://dx.doi.org/10.1109/CVPR.2015.7298682
53. Serengil, S.I., Ozpinar, A.: Lightface: A hybrid deep face recognition framework. In: 2020 Innovations in Intelligent Systems and Applications Conference (ASYU). pp. 1–5 (2020)
54. Sidiropoulos, G.K., Kiratsa, P., Chatzipetrou, P., Papakostas, G.A.: Feature extraction for finger-vein-based identity recognition. Journal of Imaging 7(5), 89 (2021)
55. Song, K.C., Yan, Y.H., Chen, W.H., Zhang, X.: Research and perspective on local binary pattern. Acta Automatica Sinica 39(6), 730–744 (2013), https://www.sciencedirect.com/science/article/pii/S1874102913600518
56. Tang, C., Lu, J., Liu, J.: Non-contact heart rate monitoring by combining convolutional neural network skin detection and remote photoplethysmography via a low-cost camera. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops (June 2018)
57. Thavalengal, S., Andorko, I., Drimbarean, A., Bigioi, P., Corcoran, P.: Proof-of-concept and evaluation of a dual function visible/nir camera for iris authentication in smartphones. IEEE Transactions on Consumer Electronics 61(2), 137–143 (2015)
58. Thavalengal, S., Bigioi, P., Corcoran, P.: Iris authentication in handheld devices-considerations for constraint-free acquisition. IEEE Transactions on Consumer Electronics 61(2), 245–253 (2015)
59. Wang, W., Den Brinker, A.C., Stuijk, S., De Haan, G.: Algorithmic principles of remote ppg. IEEE Transactions on Biomedical Engineering 64(7), 1479–1491 (2016)
60. Waseem, M., Khowaja, S.A., Ayyasamy, R.K., Bashir, F.: Face recognition for smart door lock system using hierarchical network. In: 2020 International Conference on Computational Intelligence (ICCI). pp. 51–56 (2020)
61. William, I., Ignatius Moses Setiadi, D.R., Rachmawanto, E.H., Santoso, H.A., Sari, C.A.: Face recognition using facenet (survey, performance test, and comparison). In: 2019 Fourth International Conference on Informatics and Computing (ICIC). pp. 1–6 (2019)
62. Wimmer, G., Prommegger, B., Uhl, A.: Finger vein recognition and intra-subject similarity evaluation of finger veins using the cnn triplet loss. In: 2020 25th International Conference on Pattern Recognition (ICPR). pp. 400–406 (2021)
63. Wolf, L., Hassner, T., Maoz, I.: Face recognition in unconstrained videos with matched background similarity. In: CVPR 2011. pp. 529–534. IEEE (2011)
64. Xi, S., Yang, L., Zhao, Y., et al.: A practical design for face recognition with anti-spoofing based on non-visible light cameras. Academic Journal of Computing & Information Science 3(2) (2020)
65. Yan, Z., Zhao, S.: A usable authentication system based on personal voice challenge. In: 2016 International Conference on Advanced Cloud and Big Data (CBD). pp. 194–199. IEEE (2016)
66. Yang, W., Hu, J., Wang, S.: A delaunay quadrangle-based fingerprint authentication system with template protection using topology code for local registration and security enhancement. IEEE transactions on Information Forensics and Security 9(7), 1179–1192 (2014)

José Misael Burruel Zazueta received a degree in mechatronics engineering in 2018 and a master's degree in computer science in 2021, both from the Tecnológico Nacional de México campus Culiacan. During his master's degree he developed research works in face recognition based on deep learning, generating research articles accepted in prestigious

conferences in Mexico. Also, he has collaborated in research related to wind time series forecasting with a paper published in IEEE. He has supervised four engineering thesis. He is currently pursuing a PhD in engineering sciences, focusing on biometric recognition based on artificial intelligence techniques.

Héctor Rodríguez Rangel is a graduate of the Technological Institute of Morelia, obtaining a Computer Systems bachelor's degree (2009). The master's and doctorate degrees were carried out in the Postgraduate Department of the Faculty of Electrical Engineering at UMSNH (2009, 2014). He is currently a Full-time Professor at the Technological Institute of Culiacán a and a member of the Mexican National System of Researchers, distinguished as Level 1. During his doctoral degree, he made an academic stay at the University of Oregon. In his doctorate, he worked on a qualitative bifurcation diagram project. His lines of research focus on Optimization, Intelligent Computing, and Pattern Recognition.

Gloria Ekaterine Peralta-Peñuñuri is a full-time teacher in the Technological Institute of Culiacán. In recent years, she has participated in financed research projects together with researchers from the Technological Institute of Culiacán, where several postgraduate and bachelor's degree students have participated. Gloria is currently in the second semester of her Ph.D. program. Her lines of research focus on Education and Applied Intelligent Computing.

Vicenç Puig received the telecommunications engineering degree in 1993 and the Ph.D. degree in Automatic Control, Vision, and Robotics in 1999, both from Universitat Politècnica de Catalunya (UPC). He is Full Professor of the Automatic Control Department and a Researcher at the Institut de Robòtica i Informàtica Industrial, both from the UPC. He is currently the Head of the Research Group in Advanced Control Systems and the Responsible of the PhD Programme in Automatic Control and Robotics. at UPC. Formerly, he was the Director of the Automatic Control Department. He has developed important scientific contributions in the areas of fault diagnosis and fault tolerant control using interval and linear-parameter-varying models using set-based approaches. He has participated/leaded more than 20 European and national research projects in the last decade. He has also led many private contracts with several companies and has published more than 120 journal articles and more than 450 in international conference/ workshop proceedings. He has supervised over 20 Ph.D. dissertations and over 40 master's theses/final projects. He is currently the chair of the IFAC Safeprocess TC Committee 6.4 (2020-until now) and was the vice chair (2014–2017). He has been the general chair of the Third IEEE Conference on Control and Fault-Tolerant Systems (Systol 2016 and 2021) and the IPC chair of the IFAC Safeprocess 2018.

Ignacio Algredo-Badillo received the B.Eng in Electronic Engineering from Technologic Institute of Puebla (ITP) in 2002 and the M.Sc. and Ph.D degrees in Computer Science from National Institute for Astrophysics, Optics and Electronics (INAOE) in 2004 and 2008, respectively. Since 2017, he is researcher of the Mexican National Council for Science and Technology (CONACYT). Currently, he has been involved in the design and development of cryptographic systems, reconfigurable architectures, software radio platforms, FPGA implementations, and application specific hardware acceleration.

Luis Alberto Morales-Rosales graduated in 2009 from the National Institute of Optical and Electronic Astrophysics as a Ph.D. in Computer Sciences. He is part of the Mexican National System of Researchers, distinguished as Level 1. Currently, he works at Michoacán University of San Nicolás de Hidalgo (UMSNH) as Professor Conacyt. His research includes Computational Models for Transportation Engineering, Intelligent Computing, Security and Distributed Systems.

Received: February 29, 2024; Accepted: July 20, 2024.

Computer Science and Information Systems 21(4):1389–1410 https://doi.org/10.2298/CSIS240401041S

BLSAE-SNIDS: A Bi-LSTM Sparse Autoencoder Framework for Satellite Network Intrusion Detection

Shi Shuxin[2], Han Bing[1], Wu Zhongdai[1,3], Han Dezhi[2], Wu Huafeng[4,⋆], and Mei Xiaojun[4]

[1] State Key Laboratory of Maritime Technology and Safety
200135 Shanghai, China
han.bing@coscoshipping.com
[2] College of information Engineering, Shanghai Maritime University
201306 Shanghai, China
dezhihan88@sina.com.cn
shishuxin@stu.shmtu.edu.cn
[3] Shanghai Ship and Shipping Research Institute Co.,Ltd.
200135 Shanghai, China
wu.zhongdai@coscoshipping.com
[4] Merchant marine college, Shanghai Maritime University
201306 Shanghai, China
xjmei94@163.com, hfwu@shmtu.edu.cn

Abstract. Due to disparities in tolerance, resource availability, and acquisition of labeled training data between satellite-terrestrial integrated networks (STINs) and terrestrial networks, the application of traditional terrestrial network intrusion detection techniques to satellite networks poses significant challenges. This paper presents a satellite network intrusion detection system named Bi-LSTM sparse self-encoder (BLSAE-SNIDS) to address this issue. Through the development of an innovative unsupervised training Bi-LSTM stacked self-encoder, BLSAE-SNIDS facilitates feature extraction from satellite network traffic, diminishes dimensionality, considerably reduces training and testing durations, and enhances the attack prediction accuracy of the classifier. To assess the efficacy of the proposed model, we conduct comprehensive experiments utilizing STIN and UNSW-NB15 datasets. The results obtained from the STIN dataset demonstrate that BLSAE-SNIDS achieves 99.99% accuracy with reduced computational and transmission overheads alongside enhanced flexibility. Furthermore, results from the UNSW-NB15 dataset exhibit BLSAE-SNIDS' proficiency in detecting various network intrusion attacks efficiently. These findings indicate that BLSAE-SNIDS suits general satellite security networks and offers a novel approach to designing security systems for polar satellite networks, thus exhibiting practical utility.

Keywords: Satellite-terrestrial integrated networks, LSTM, Automatic encoder, Unsupervised learning, Network security, Deep learning.

1. Introduction

As a significant augmentation to terrestrial networks, satellites extend the coverage of such networks, facilitating convenient access for remote users. Consequently, the prevail-

⋆ Corresponding Author

ing concept in sixth-generation mobile communications (6G) is to amalgamate satellite networks with terrestrial networks to construct an STIN system to achieve seamless global coverage. Nevertheless, STIN presents inherent security vulnerabilities, primarily due to satellite node exposure and the utilization of long-distance communication links spanning distances between 150 and 35,800 km [1]. These factors make the system susceptible to cyber-attacks, engendering substantial security risks [2, 3]. As 6G technology continues to advance, the security of STIN has emerged as a paramount concern in recent years.

Due to their extensive coverage, high capacity, and long-distance transmission capabilities, satellite communication networks find widespread applications in navigation, spaceflight, radio and television broadcasting, disaster rescue, and relief. Particularly in remote areas characterized by limited services and dispersed populations, satellite communication emerges as the optimal choice due to its independence from substantial ground infrastructure investments.

The Arctic shipping lanes, pivotal for maritime navigation, pose challenges in relying on ship-to-shore communications for navigation and hazard warnings. These challenges arise from inadequate data on the navigational environment in the Arctic, a lag in communications infrastructure, and a dearth of navigational aids such as beacons, lighthouses, and coastal radio services. Consequently, due to limited experience in Arctic navigation, ships operating in the region are exposed to elevated safety risks. To foster endeavors such as environmental conservation, resource utilization, and commercial transportation in the Arctic, it is imperative to conduct thorough research on communication and navigation support systems tailored for Arctic shipping lanes, aiming to enhance safety during polar navigation. Introducing the message service of a satellite navigation system presents a novel approach to information transmission in the Arctic region.

However, as the network scale expands, the vulnerability of satellite communication transmission links to intrusion grows. Regrettably, a substantial disparity exists between satellite and terrestrial networks concerning computing power, bandwidth, and other resources. Furthermore, upgrading satellite hardware post-launch poses significant challenges. Hence, it is necessary to devise effective Star-Terrestrial Network Intrusion Detection Systems (SAT-IDS) to afford robust protection for STIN.

The primary objective of a Network Intrusion Detection System (NIDS) is the detection of harmful intrusions, typically employed to monitor network traffic, distinguishing between normal and malicious activity to effectively mitigate the latter. Designing NIDSs for satellite networks necessitates consideration of the limited computational power of each satellite node, alongside stringent security and privacy requirements during transmission, thereby emphasizing the importance of resource utilization efficiency [4–6].

Numerous design approaches for NIDSs have been proposed, often leveraging machine learning algorithms to discern traffic patterns [7,8]. Evaluating NIDS efficacy hinges on its ability to accurately detect attacks, demanding comprehensive datasets encompassing both normal and abnormal behaviors. However, applying such datasets from terrestrial networks directly to satellite communications presents challenges owing to the unique characteristics of satellite networks.

1. Satellite and terrestrial networks exhibit differing susceptibilities to various attack types. While terrestrial networks may grapple with challenges such as Backdoor and Botnets, exacerbated by the computational prowess and openness of terrestrial network nodes [9–11] , satellite nodes with constrained resources face concerns over

malicious distributed denial of service (DDoS) attacks, such as Synchronized Sequence Number (Syn) flooding. Accordingly, formulating secure datasets for distinct satellite network domains warrants meticulous consideration.
2. Furthermore, the limited resources and computational capabilities inherent in satellite networks pose additional challenges, as satellite nodes, when targeted, can swiftly succumb to attacks, making recovery arduous.
3. Extracting pertinent feature selections from satellite network traffic is inherently challenging. The intricacies of satellite networks, coupled with the costliness of communication links, render the acquisition of large training datasets prohibitive.

Addressing the challenges inherent in existing satellite network intrusion detection methods, this paper presents a novel approach termed Bi-LSTM sparse autoencoder-based Satellite Network Intrusion Detection System (BLSAE-SNIDS). The system leverages autoencoders for feature learning on unlabeled data, thereby generating new feature representations in an unsupervised manner. These features are inputted into a classifier following a pre-training phase to enhance intrusion detection capability and classification accuracy. The contributions of this study are outlined as follows:

1. Given the intricacies of satellite networks, obtaining ample labeled data for learning purposes proves exceedingly difficult and costly. Consequently, we propose BLSAE-SNIDS, which employs an overarching architecture based on Bi-LSTM sparse autoencoders capable of extracting implicit knowledge from unlabeled data and learning unsupervised representations. This approach enables superior classification outcomes with a limited number of labels.
2. Utilizing the Bi-LSTM sparse autoencoder effectively reduces the dimensionality of input vectors in the classification task. This operation can be performed within the network nodes of the satellite network, substantially curbing the volume of traffic transmitted to the satellite network's control center. Moreover, the computational nodes need not be equipped with high-performance hardware to swiftly detect known and unknown intrusion traffic. This outcome holds significant value, considering the precious resources of satellite network links.
3. The robustness of the Intrusion Detection System (IDS) is evaluated using a real STIN dataset. Extensive experimentation on the STIN dataset demonstrates that the proposed model adeptly detects various satellite network attacks.

2. Related Work

The expansion of satellite networks has brought about considerable security challenges, prompting the development of security technologies, notably Network Intrusion Detection Systems (NIDS). Numerous studies have employed diverse machine learning techniques to construct effective NIDS for detecting attacks in satellite and terrestrial networks. Di et al. [12] proposed a defense architecture against Distributed Denial of Service (DDoS) attacks in satellite networks, combining distributed multipoint detection, proximity source defense, collaborative management, and protection integrity. Additionally, Li et al. [9] devised security datasets for STIN satellite and terrestrial networks, presenting a distributed NIDS employing federated learning (FL) in STIN. This system allocates resources effectively across domains to analyze and block malicious traffic, particularly DDoS attacks.

Ashraf et al. [13] introduced a novel intrusion detection approach, merging Random Forest (RF) and Multi-Layer Perceptron (MLP) using data from satellite and terrestrial networks. This method enhances intrusion detection accuracy compared to other models, validated through experiments on NSL-KDD, KDD-CUP 99, and STIN datasets. For DDoS attacks in Flying Self-Organizing Networks (FANET), Zhang et al. [14] investigated the operational features of satellite nodes, establishing security domains based on login and logout mechanisms. They proposed a hierarchical distributed satellite network intrusion detection model and a collaborative mechanism for intrusion detection agents inside and outside the satellite. Guo et al. [15] introduced a blockchain-based Distributed Collaborative Entrance Defense (DCED) framework to shield the satellite network from DDoS attacks. This system records and aggregates network traffic characteristics at the satellite network's entrance, bolstering its defense capabilities. Azar et al. [16] proposed a hybrid intrusion detection system for Satellite Terrestrial communication systems (STIN) and terrestrial networks (UNSW-NB15). They employed a Random Forest (RF) based Sequential Forward Feature Selection (SFS) method to select critical features from the dataset. Experimental results demonstrate a notable enhancement in detection accuracy and computational efficiency. The SFS-RF model, employing 10 selected features, achieved 90.5% accuracy on the STIN dataset and 78.52% on the UNSW-NB15 dataset. The RF-SFS-GRU model excelled in deep learning, attaining 87% and 79% accuracy on the respective datasets.

The Satellite Intrusion Detection System (SAT-IDS) uses machine learning to differentiate between normal and abnormal network traffic. Various machine learning approaches have been explored in SAT-IDS development, aiming to conduct feature selection tasks to extract pertinent features from satellite network traffic datasets, thereby enhancing classification outcomes. Nonetheless, acquiring extensive labeled training sets poses a significant challenge due to the intricate nature of satellite networks and the costly communication links involved. Consequently, self-supervised learning and unsupervised feature learning have emerged as viable alternatives. In these methodologies, the algorithm initially learns a robust feature representation from vast amounts of unlabeled data. Later, when addressing a specific classification task, this learned feature representation can be applied to labeled data to solve the classification problem through supervised learning. Unlike supervised learning, unsupervised learning does not necessitate assuming that the unlabeled data shares the same distribution as the labeled data, rendering the model more adaptable. Zhu et al. [17] proposes a flexible and innovative framework for satellite network intrusion detection systems, leveraging deep learning techniques. Here, a sparse autoencoder serves as a network feature extractor. However, it is noteworthy that the research is confined to system design and has not been executed practically.

Therefore, this paper proposes a Satellite Network Intrusion Detection System (BLSAE-SNIDS) based on a Bi-LSTM sparse self-encoder. The approach leverages unlabeled data to train the Bi-LSTM sparse self-encoder for feature extraction from satellite network traffic while reducing dimensionality. By reusing the self- encoder's encoder part and integrating it with a classifier for data classification, the model's parameters and training time can be significantly reduced while enhancing its generalization capacity. The method acquires implicit knowledge from unlabeled data through unsupervised learning, facilitating superior classification outcomes even with limited labeled data. This effec-

tively differentiates between normal and abnormal traffic for satellite network intrusion detection.

3. Methods

3.1. Model Overview

This paper introduces a satellite network intrusion detection method based on a Bi-LSTM sparse autoencoder, as depicted in Fig. 1. The proposed method employs unlabeled data to train the Bi-LSTM sparse autoencoder, leveraging the temporal characteristics of satellite network traffic data. Feature extraction and dimensionality reduction are achieved through the encoder section of the Bi-LSTM sparse autoencoder, followed by reconstruction via the decoder section. Subsequently, the output of the trained encoder undergoes classification, mapping to distinct classes through a classifier. This approach effectively learns essential features from input data, reducing model parameters and training time by reusing the autoencoder's encoder segment. Consequently, the method demonstrates proficiency in distinguishing between normal and abnormal traffic for satellite network intrusion detection.

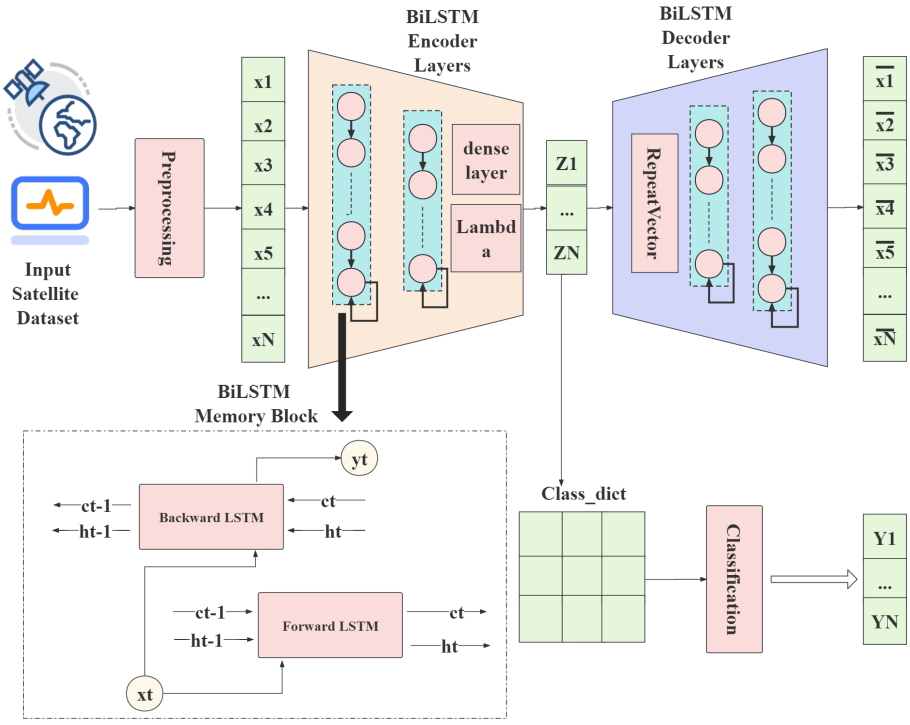


Fig. 1. BLSAE-SNIDS structure diagram

The Bi-LSTM encoder comprises two bidirectional LSTM layers with 64 and 32 hidden units, respectively. It takes a series of high-dimensional input data, efficiently preserves dependencies between multiple data points in the time series, and produces a fixed-size vector. This process reduces the high-dimensional input vector representation to a lower dimension. In our model, the Bi-LSTM encoder consists of two layers: the first with 64 LSTM units and the second with 32 LSTM units. This is succeeded by a Lambda layer with 'n' units and a dense layer with 'n' units, where 'n' represents the number of categories. The Lambda layer maps the encoder output to one of the 'n' categories through a predefined mapping.

The Bi-LSTM decoder generates a fixed-size input sequence using a simplified representation of the input data in the hidden space. During reconstruction, the decoder minimizes the difference, i.e., the reconstruction error, between the original and reconstructed data. The input to the decoder is the dimensionality-reduced output of the encoder. The decoder unfolds the encoding by stacking it in the reverse order of the encoder. It consists of multiple repetition vector layers replicating the encoder output 16 times, followed by two bidirectional LSTM layers with 32 and 64 hidden units, respectively, and a time-distributed dense layer with 1 hidden unit. The final decoder output is a reconstruction of the encoder input.

The class dictionary (Class_dict) matrix records the activation count of each neuron in the encoder for each class label. By selecting the neuron with the highest count for each class in the class dictionary, we determine the relevant class for each encoded sample.

3.2. Bi-LSTM stacked autoencoder

Bi-LSTM LSTM, a temporal recurrent neural network, is adept at processing and predicting significant events with long intervals and time series data delays. Bi-LSTM, linking two hidden layers operating in both input-to-output directions, is proposed for classifying satellite network traffic. Its structure enables efficient utilization of future and past features within specific time ranges during training, thereby enhancing prediction accuracy. The Bi-LSTM layer effectively handles time series problems and improves prediction accuracy. Additionally, the auto-encoder (AE) layer in our proposed structure learns features from data and reduces data dimensionality, complemented by the Bi-LSTM layer to further enhance prediction accuracy [18, 19]. LSTM networks excel in characterizing time series data. Unidirectional LSTM methods may yield errors by relying solely on forward memory processing, potentially failing to capture contextual information. The LSTM layer consists of a series of LSTM units that only process input data in the forward direction. In contrast, Bi-LSTM includes additional LSTM layers processing data in reverse, as illustrated in Fig. 2. Training a Bi-LSTM network is akin to concurrently training two independent unidirectional LSTM networks. One network is trained on the original input sequence, while the other is trained on a reversed copy. This approach furnishes the network with more contextual information, fostering faster and more comprehensive problem learning.

The BiLSTM method extends traditional LSTM by incorporating sequence information in forward (past to future) and backward (future to past) directions, enhancing sequence classification performance. BiLSTM networks have emerged as superior solutions for capturing interactions between contexts. Training BiLSTM networks involves simultaneously training two unidirectional LSTM networks on the input sequence. One LSTM

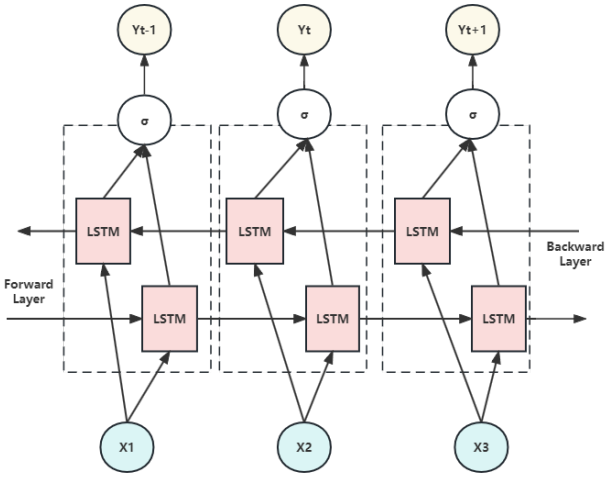


Fig. 2. Expanded view of Bi-LSTM

network processes the input sequence, while the other operates on a reverse copy of the input sequence. This approach provides the network with additional background information, facilitating faster and more comprehensive problem learning.

As depicted in Fig. 2, the LSTM unit comprises forgetting gates, input gates, and output gates to facilitate effective short-term and long-term memorization. Specifically, BiLSTM memory blocks are trained in two directions: one LSTM proceeds from past to future. At the same time, the other progresses from future to past, ensuring that each time step incorporates both past and future data. Each layer computes the respective function for each element in the input sequence. The variables associated with LSTM units are represented accordingly.

$$\begin{aligned} i_t &= \sigma\left(W_{ii}x_t + b_{ii} + W_{hi}h(t-1) + b_{hi}\right) \\ f_t &= \sigma\left(W_{if}x_t + b_{if} + W_{hf}h(t-1) + b_{hf}\right) \\ g_t &= \tanh\left(W_{ig}x_t + b_{ig} + W_{hg}h(t-1) + b_{hg}\right) \\ o_t &= \sigma\left(W_{io}x_t + b_{io}W_{hp}h(t-1) + b_{ho}\right) \\ c_t &= f_t * c(t-1) + i_t * g_t \\ h_t &= o_t{}^* \tanh\left(c_t\right) \end{aligned} \tag{1}$$

where$(W_{ii}, W_{if}, W_{ig}, W_{io})$ is the input weights, $(W_{hi}, W_{hf}, W_{hg}, W_{ho})$ is the cyclic weights, $(b_{hi}, b_{hf}, b_{hg}, b_{ho})$ is the cyclic bias, h_t denotes the hidden state at moment t, c_t denotes the cell state at moment t, x_t denotes the input at moment t, $h\,(t-1)$ denotes the hidden state of the layer at moment (t-1) or the initial hidden state at moment 0, and i_t , f_t , g_t , o_t denote the input gates, forgetting gates, cell gates and outputs, respectively. gates. is the sigmoid function and '$*$' is the Hadamard product.

Automatic encoder The autoencoder is a neural network architecture employed in unsupervised learning, data compression, downscaling, and data generation. It compresses

input data to obtain a concise representation (encoding) and then reconstructs the original data with minimal information loss (decoding). Typically, a standard autoencoder comprises an input layer, an output layer, and multiple hidden layers. Fig. 3 illustrates a basic autoencoder model, which can be elucidated by assessing the reconstruction error (loss) between the encoder, decoder, input, and output data [20–22].

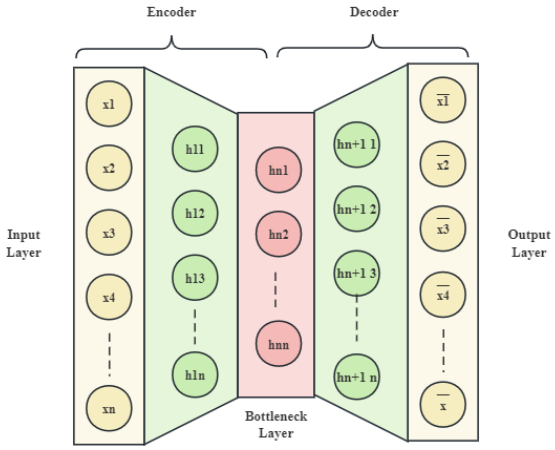


Fig. 3. A simple autoencoder structure

Encoding: The encoding operation maps the input data x, i.e., the high-dimensional vector $x \in R^m$, to a low-dimensional bottleneck layer representation (h). This is given by the following equation, where w_i is the weight matrix, b_i is the bias, and f_1 is the activation function.

$$h = f_1 \left(w_i x + b_i \right) \tag{2}$$

Decoding: Eq. (3) illustrates how the decoding operation uses the bottleneck layer representation (h) to produce the output of the attempted reconstruction x. Where f is the activation function of the decoder, w is the weight matrix, b denotes the bias, and x denotes the reconstructed input sample.

$$\widehat{x} = f_2 \left(w_j h + b_j \right) \tag{3}$$

Reconstruction Loss: as shown in Eq. (4), a reconstruction loss (L) is computed in a typical self-encoder model to minimize the difference between the output and the input. Here, x denotes the input data, $\widehat{x}$ denotes the output/predicted data, and n is the number of samples in the training dataset [22].

$$\min_{w,b} L\left(x-\widehat{x}\right); where\ L\left(x-\widehat{x}\right)=\frac{1}{n}\sum_{n=1}^{n}\left|\widehat{x_t}-x_t\right| \tag{4}$$

Sparse Autoencoder A sparse autoencoder is a variant of the traditional autoencoder that includes a sparsity constraint on the hidden units, ensuring that only a small number of neurons are active at any given time. This constraint is typically enforced by adding a penalty term to the loss function. The sparsity constraint helps the autoencoder to learn more robust and useful features by focusing on the most significant aspects of the data. This characteristic is particularly advantageous for satellite network traffic, where capturing essential features and reducing dimensionality are crucial for efficient and accurate intrusion detection. The sparse autoencoder's structure consists of an encoder, which compresses the input data into a lower-dimensional representation, and a decoder, which reconstructs the original data from this representation. By doing so, the autoencoder can effectively capture the most critical features of the data, improving the performance and generalizability of the intrusion detection system.

Bi-LSTM Stack autoencoder In satellite network intrusion detection, we formulated a model that integrates Bi-LSTM and autoencoder components, as depicted in Fig. 4 The training effect of satellite network traffic using ordinary autoencoders is general. This method treats each traffic instance as an independent input, neglecting the temporal characteristics inherent in satellite network traffic and resulting in elevated training losses.

3.3. Proposed SAT-IDS

Drawing inspiration from machine learning and deep learning methodologies, we introduce two hybrid intrusion detection system (IDS) techniques. The initial approach integrates a Bi-LSTM sparse encoder with a class dictionary termed BLSAE-SNIDS-CD. The second technique combines a Bi-LSTM sparse encoder with a deep neural network (DNN), denoted as BLSAE-SNIDS-DNN.

BLSAE-SNIDS-CD We utilize a trained Bi-LSTM Decoder to encode the network traffic data. Initially, a Class_dict matrix of size $Num_class \times Num_class$, where Num_class represents the number of classes, is initialized. This matrix records the activation frequency of each neuron in the encoder for every class label. Iterating over the training labels, we update the Class_dict matrix based on the following guidelines: for each image, we identify the index of the largest value in its feature mapping (representing the most active neuron in the encoder) and increment the value in the corresponding row and column of the Class_dict matrix by 1. This results in an output of the Class_dict matrix, which illustrates the activation frequency of each encoder neuron in distinguishing between different classes. Ideally, each neuron should activate exclusively for one class and not others.

To initialize a vector named Neuron_class, we map each neuron in the encoder to a class label. We iterate through the rows of the Class_dict matrix and assign each neuron to the class with the highest value in that row. Subsequently, we output the Neuron_class

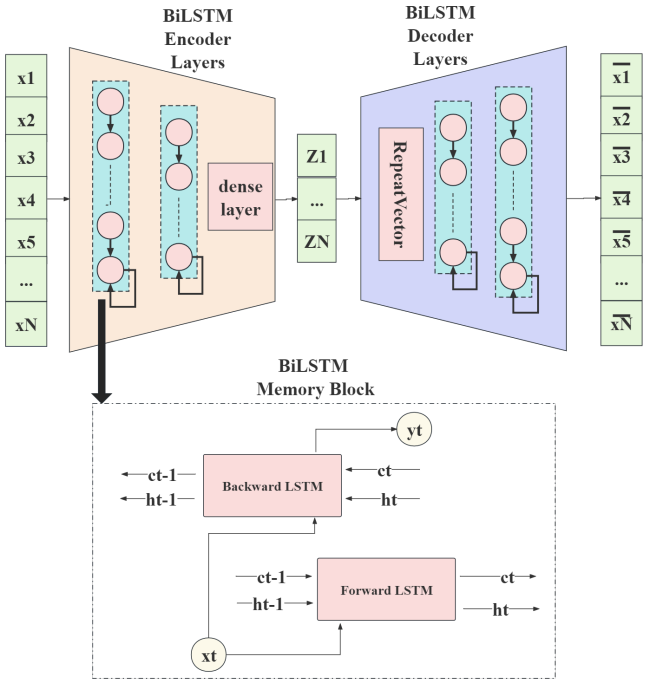


Fig. 4. Structure of Bi-LSTM autoencoder

vector, which indicates the class each neuron represents in the encoder. Then, we iterate over the predicted vector and replace each index with the corresponding class label from the neuron_class vector. Next, we compare the predicted vectors to the true labels of the test images. This comparison yields a Boolean vector indicating the correctness of each prediction.

The model can significantly influence satellite network traffic analysis, particularly in scenarios featuring a substantial volume of unlabeled data and a limited amount of labeled data. Notably, the model does not presuppose that the unlabeled data must align with the distribution of the labeled data.

BLSAE-SNIDS-DNN The structure of BLSAE-SNIDS-DNN, illustrated in Fig. 5, employs the same unlabeled data as BLSAE-SNIDS-CD for training the autoencoder. This enables the model to learn the intrinsic features of the input data and minimize the Mean Square Error (MSE) between the input and output. During training, BLSAE-SNIDS-DNN utilizes the encoder section for feature extraction on labeled data and then constructs a fully connected layer atop it as a classifier. Training the classifier with labeled data facilitates the prediction of the input data class based on the feature vector and minimizes the cross-entropy loss (CE). Subsequently, the encoder portion is employed for feature extraction on test data, and the classifier is utilized to predict its category. Leveraging the encoder's pre-trained weights from unsupervised training significantly enhances the model's accuracy.

3.4. Data Collection and Preprocessing

STIN data set This study employs the dataset provided by [9] to represent the satellite dataset for evaluating our model. The STIN security dataset encompasses various attack types observed in both satellite and terrestrial networks, comprising two satellite attack types and nine terrestrial attack types. The construction of the STIN dataset incorporates flow-based features. The distribution of the STIN satellite dataset is illustrated in Fig. 6.

This section outlines the following data preprocessing steps:

1) Feature selection: Feature selection is imperative to accommodate the limited resources of satellite nodes and minimize training and testing costs, optimizing resource utilization and reducing expenses. Initially, irrelevant features pertaining to the dataset's environmental and temporal aspects, such as quintuple, are discarded. Subsequently, features with missing values or that are entirely composed of zeros are eliminated. Further refinement is achieved through Pearson coefficient analysis and variance selection. Pearson's coefficient assesses the correlation between feature columns, leading to the removal of strongly correlated columns and subsequent reduction of the feature space. Variance selection filters each feature column's variance value, eliminating dimensions with low variance. Table 1 demonstrates the retention of the top 16 flow-level features crucial for intrusion detection, encompassing various benign and malicious traffic types simulated in the prototype.

2) Normalization: Normalization ensures uniformity in the range of values across each feature. Utilizing "MinMax" normalization, the range of feature values is scaled to fall between 0 and 1. The difference between the minimum value of the vector and the scale

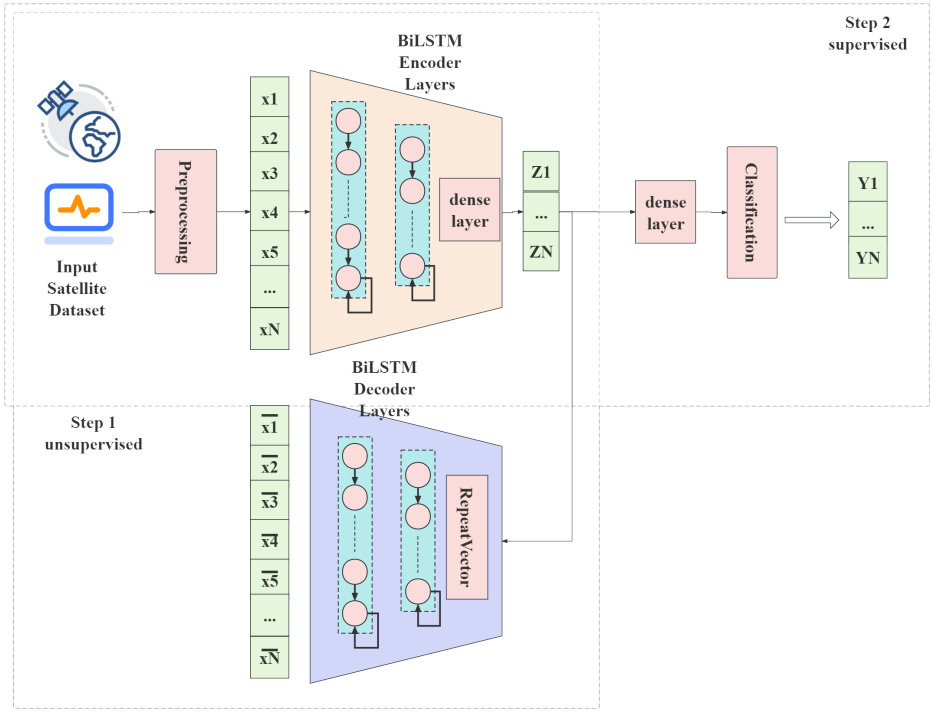


Fig. 5. Structure of supervised polar satellite intrusion detection system

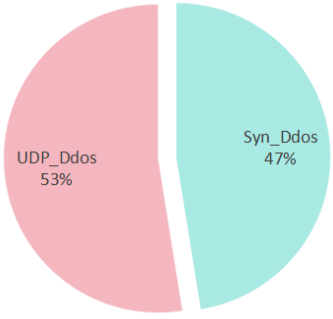


Fig. 6. Distribution of STIN satellite data sets

Table 1. Selected 16 features

Feature	Description
fl_dur	Flow duration
fw_pk	Total packets in the forward direction
l_fw_pkt	Total length of forward packets
l_bw_pkt	Total length of backward packets
pkt_len_min	Minimum length of a fow
pkt_len_max	Maximum length of a flow
pkt_len_std	Standard deviation length of a flow
fl_byt_s	Packet bytes transmitted per second
fl_iat_min	Minimum time between two backward packets,
bw_iat_tot	Total time between of two backward packets
bw_iat_min	Minimum time between of two backward packets
fw_hdr_len	Number of bytes used in forward packet header
bw_pkt_s	Number of backward packets per second
urg_cnt	Number of packets with URG
down_up_ratio	Ratio of forward and backward packet counts
bw_win_byt	Number of backward bytes in the initial window

size determines the new value. The normalization equation is as follows:

$$\bar{x}_i = \frac{x_i - \min(x_i)}{\max(x_i) - \min(x_i)} \tag{5}$$

Here, $\overline{x_i}$ represents the normalized value, x_i denotes the i th feature vector, $\min(x_i)$ returns the minimum value of the vector, and $\max(x_i)$ returns the maximum value of the vector.

UNSW-NB15 dataset This section discusses the following data preprocessing steps.

The UNSW-NB15 dataset specializes in terrestrial traffic representation. This dataset is pivotal in safeguarding against unknown attacks, evaluating performance, and ensuring generalizability for machine learning-based intrusion detection systems. However, the NSL-KDD and KDDCup99 datasets have faced criticism due to their outdated attack types, imbalanced training and test set distributions, and lack of support for certain common network protocols, rendering them incompatible with contemporary cybersecurity requirements [23, 24].

To address the limitations of KDDCup/NSL-KDD, Moustaf and Slay [25] developed a more sophisticated intrusion detection dataset called UNSW-NB15, designed to reflect modern attacks and protocols. UNSW-NB15 is an example of an intrusion detection dataset crafted by researchers at the Australian Center for Cybersecurity (ACCS) using the IXIA tool, extracted from 100 GB of normal and modern datasets from attack traffic. The full UNSW-NB15 dataset comprises 2.5 million data records, encompassing one normal category and nine attack categories - Analytsis, Backdoor, DoS, Exploit, Fuzzers, Generic, Reconnaissance, Shellcode, and Worms - with the raw data featuring 49 features.

The dataset creator also provided a partitioned dataset comprising 10% of the records, including a training set (175,341 records) and a test set (82,332 records), as depicted in

Fig. 7. The statistical distributions of the training and test set samples have been validated to be highly correlated, indicating the reliability of the machine learning model partitioning. Additionally, a few categories, such as Analytics, Backdoor, Shellcode, and Worms, comprise less than 2% of the dataset. In this 10% subset, some redundant features were eliminated, reducing the number of features to 42. In our study, we utilized this 10% dataset subset for categorization.

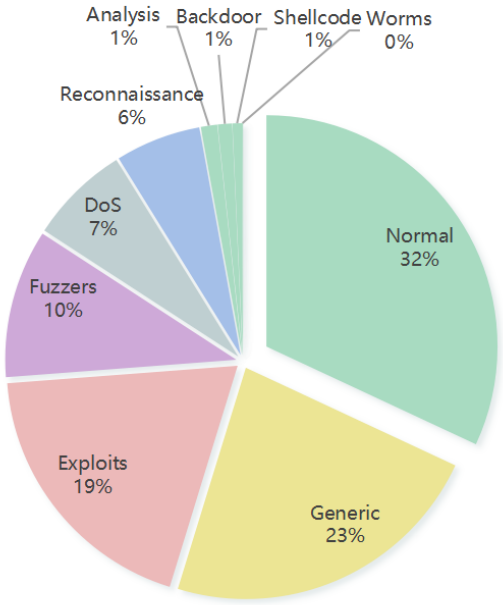


Fig. 7. Distribution of UNSW-NB15 data sets

This section outlines the following preprocessing steps:

1) Data cleaning: The provided UNSW-NB15 training and test sets initially contain 44 features. However, only 42 features are deemed meaningful, while the remaining two features serve as class labels for attacks. ”attack_cat” represents a multi-class label, while ”label” serves as a binary class label. Therefore, irrelevant labels are eliminated based on the model’s task. Additionally, 67,601 rows in the dataset containing null and duplicate values are cleaned.

2) One-hot encoding: The dataset includes three categorical features: ”service,” ”proto,” and ”state.” These features transform using one-hot encoding, a technique that converts each feature value into a binary feature.

3) Normalization: Normalization ensures uniformity in the range of values for each feature. ”MinMax” normalization is employed to scale the feature values between 0 and 1, as demonstrated in equation (5). The difference between the minimum value of the equation and the scale size determines the new value.

4) Feature selection: A comprehensive feature selection approach combining information gain and random forest importance is implemented to identify significant features. Ultimately, 42 optimal flow-level features are retained, which is crucial for intrusion detection and delineating diverse types of benign and malicious traffic simulated in the prototype.

4. Results and Discussion

The proposed BLSAE-SNIDS-CD and BLSAE-SNIDS-DNN are tested on STIN and UNSW-NB15 datasets, with STIN as the satellite dataset and UNSW-NB15 as the ground dataset, to provide high security for the satellite-ground network. The experimental environment is shown in Table 2.

Table 2. Experimental environment

Equipment	
CPU	Intel(R) Xeon(R) Silver 4116 CPU@2.10GHz
GPU	Quadro P4000
RAM	64GB
OS	Ubuntu 16.04
Python	3.8.3
Database	STINUNSW-NB15

4.1. Evaluation Metrics

We utilize standard performance metrics to assess the efficacy of our proposed model on the STIN dataset, including accuracy, precision, recall, and F1 score. They are calculated using the following equations:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{6}$$

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

$$\text{Re}call = \frac{TP}{TP + FN} \tag{8}$$

$$F1\ Score = 2 \times \frac{Precision \times \text{Re}call}{Precision + \text{Re}call} \tag{9}$$

The performance metrics are defined as follows: True Positive (TP) represents the count of accurately recognized abnormalities. True Negative (TN) represents the count of correctly recognized normal events. False Positive (FP) represents the count of normal events incorrectly diagnosed as abnormal. False Negative (FN) represents the count of abnormal events incorrectly recognized as normal.

4.2. Result Discussion

To illustrate the effectiveness of the proposed SAT-IDS, we conducted tests on two datasets: UNSW-NB15 and STIN. The confusion matrix of BLSAE-SNIDS-CD on the STIN test set is depicted in Fig. 8.

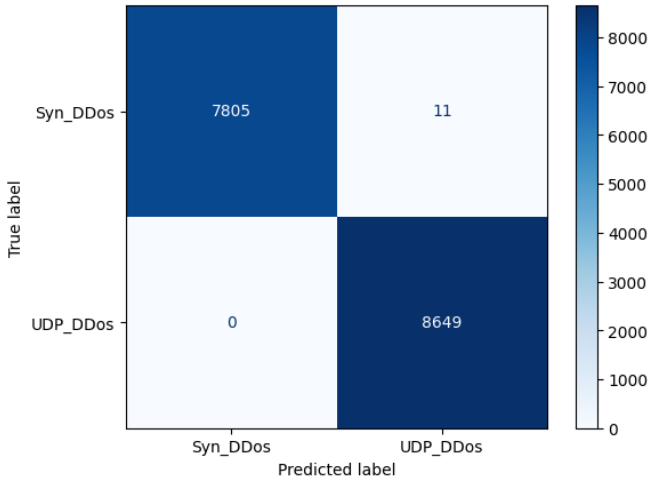


Fig. 8. Confusion matrix of BLSAE-SNIDS-CD on STIN dataset

The confusion matrix of BLSAE-SNIDS-DNN on the test set is displayed in Fig. 9. Notably, the model misclassifies only 1 test case per category, indicating a classification accuracy close to 100%.

The model undergoes training with 50 epochs and a batch size of 256. Subsequently, the learning curve depicting the loss of the training dataset is presented in Fig. 10.

We compare BLSAE-SNIDS-CD and BLSAE-SNIDS-DNN with representative SAT-IDS models. Fig. 11 illustrates the accuracy comparison among all models. It is evident that both BLSAE-SNIDS-CD and BLSAE-SNIDS-DNN exhibit higher accuracy than most of the models, demonstrating strong performance across the board.

The performance comparison of all models using the STIN dataset in terms of accuracy, precision, recall and F1 score is shown in Table 3. The results show that BLSAE-SNIDS-DNN using bi-directional LSTM stacked autoencoder coded features achieved the best performance in all evaluation metrics with 99.99% in each metric.

Table 4 presents the performance of all models utilizing the STIN dataset. Notably, BLSAE-SNIDS-DNN demonstrates superior performance compared to other models, achieving a remarkable accuracy of 99.99% for both attack types. Its performance surpasses that of CNN, SVM, and RF-SFS-GRU by a significant margin.

The STIN security dataset contains different types of attacks from terrestrial and satellite networks. In the literature, researchers have proposed different techniques that are usually applied to terrestrial networks but not applicable to satellite networks due to different reasons such as limited resources, different tolerance to attacks, limited computational

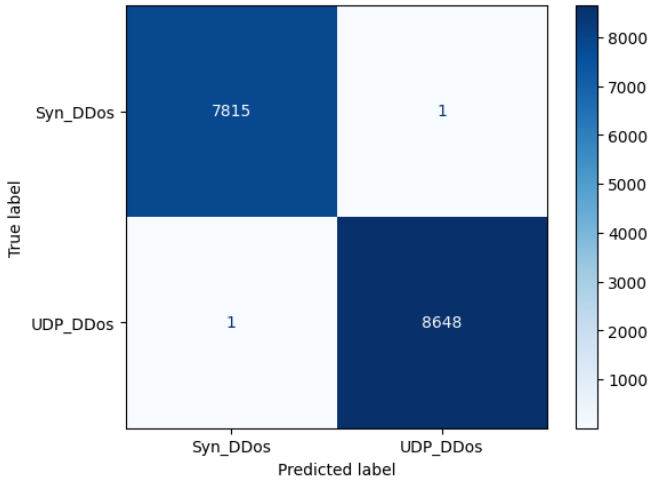


Fig. 9. Confusion matrix of BLSAE-SNIDS-DNN on STIN dataset

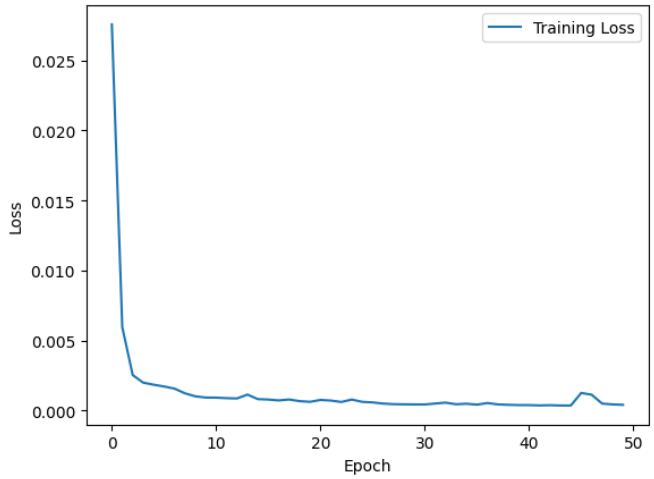


Fig. 10. Learning curves for loss of BLSAE-SNIDS-CD

power, and scarcity of satellite network datasets. The model performance comparison on the UNSW-NB15 dataset is shown in Table 5 and Fig. 12. The results show that the proposed BLSAE-SNIDS-DNN still performs better and outperforms the other models despite the fact that the performance of the other models decreases when used with the STIN dataset. The accuracy score of BLSAE-SNIDS-DNN on the UNSW-NB15 dataset is 90.83%. The model's best performance on these intrusion detection datasets shows the superiority, reliability, and ability to generalize well in the proposed model.

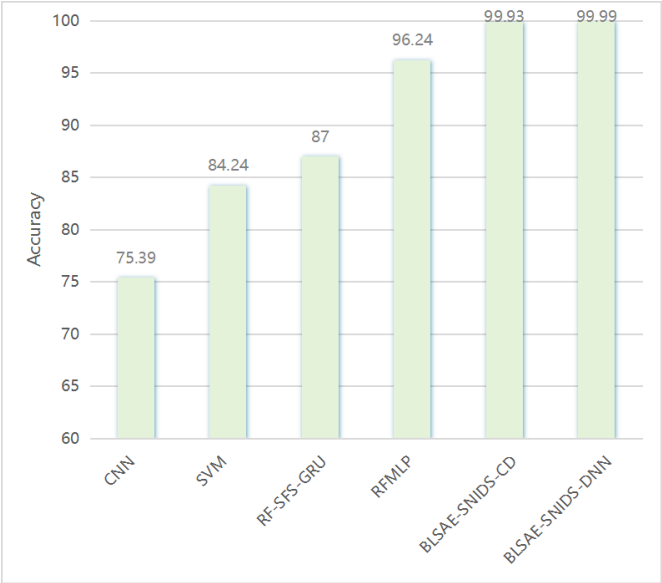


Fig. 11. Accuracy comparison diagram

Table 3. Model performance comparison on STIN satellite dataset

Ref.	Model	Accuracy(%)	Precision(%)	Recall(%)	F1 Score(%)
	CNN	75.39	82.98	75.4	74.28
	SVM	84.24	83.35	85.62	84.48
[16]	RF-SFS-GRU	87	87	86	86
[13]	RFMLP	96.24	94.28	98.67	96.47
Proposed	BLSAE-SNIDS-CD	99.93	99.99	99.99	99.9
Proposed	BLSAE-SNIDS-DNN	99.99	100	100	100

Table 4. Accuracy of classifiers on STIN satellite dataset

Attack Type	CNN	SVM	RF-SFS-GRU	RFMLP	BLSAE-SNIDS-CD	BLSAE-SNIDS-DNN
Syn_DDos	66.3	86.18	93.36	100	99.85	99.99
UDP_DDos	98.4	83.37	94.49	93.18	100	99.99

Table 5. Model performance comparison on UNSW-NB15 dataset

model	Testing acc. (%)	Precision (%)	Recall (%)	F1-Score (%)
LSTM	63.61	69.16	61.61	65.17
ANN	63.61	69.16	61.61	65.17
RF-SFS-GRU	87	87	86	86
SFS-RF	90.5	91.19	90.41	90
BLSAE-SNIDS-DNN	90.83	91.21	90.85	90.4

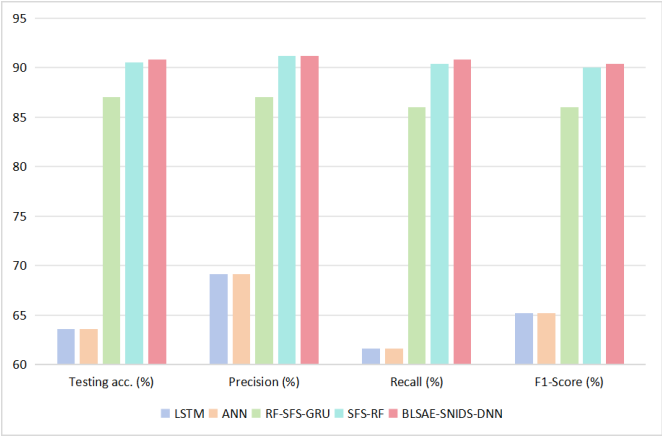


Fig. 12. Performance comparison of all models on UNSW-NB15 dataset

5. Conclusion

This study proposes a satellite network intrusion detection system (BLSAE-SNIDS) based on a Bi-LSTM sparse autoencoder, applied to both the STIN satellite dataset and the modern IDS dataset UNSW-NB15. BLSAE-SNIDS leverages unlabeled dat a to train the Bi-LSTM sparse autoencoder, enabling unsupervised representation learning that reduces dimensionality, parameters, and training time of classifiers. This enhances the model's generalization ability, making it suitable for satellite networks. Comparative analysis with state-of-the-art models demonstrates its superiority. Experimental results indicate that the proposed system achieves higher accuracy than other models on both datasets. However, this study is limited by dataset availability and is confined to the STIN dataset. Future research will aim to construct new datasets encompassing diverse satellite attack types and normal traffic to assess the efficacy of the proposed model.

Acknowledgments. This research is supported by the State Key Laboratory of Maritime Technology and Safety and the Natural Science Foundation of Shanghai under Grant 21ZR1426500.

References

1. Gaofeng Cui, Xiaoyao Li, Lexi Xu, and Weidong Wang. Latency and Energy Optimization for MEC Enhanced SAT-IoT Networks. *IEEE Access*, 8:55915–55926, 2020.
2. Charlotte Van Camp and Walter Peeters. A World without Satellite Data as a Result of a Global Cyber-Attack. *Space Policy*, 59:101458, February 2022.
3. Shangyuan Zhuang, Jiyan Sun, Hangsheng Zhang, Xiaohui Kuang, Ling Pang, Haitao Liu, and Yinlong Liu. Stinattack: A lightweight and effective adversarial attack simulation to ensemble idss for satellite-terrestrial integrated network. In *2022 IEEE Symposium on Computers and Communications (ISCC)*, pages 1–8, 2022.
4. Dezhi Han, Nannan Pan, and Kuan-Ching Li. A traceable and revocable ciphertext-policy attribute-based encryption scheme based on privacy protection. *IEEE Transactions on Dependable and Secure Computing*, 19(1):316–327, 2020.

5. Na Gao, Dezhi Han, Tien-Hsiung Weng, Benhui Xia, Dun Li, Arcangelo Castiglione, and Kuan-Ching Li. Modeling and analysis of port supply chain system based on fabric blockchain. *Computers & Industrial Engineering*, 172:108527, 2022.
6. Jiatao Li, Dezhi Han, Zhongdai Wu, Junxiang Wang, Kuan-Ching Li, and Arcangelo Castiglione. A novel system for medical equipment supply chain traceability based on alliance chain and attribute and role access control. *Future Generation Computer Systems*, 142:195–211, 2023.
7. Shaokang Cai, Dezhi Han, and Dun Li. A feedback semi-supervised learning with meta-gradient for intrusion detection. *IEEE Systems Journal*, 17(1):1158–1169, 2022.
8. Yan Wang, Dezhi Han, and Mingming Cui. Intrusion detection model of internet of things based on deep learning. *Computer Science and Information Systems*, (00):58–58, 2023.
9. Kun Li, Huachun Zhou, Zhe Tu, Weilin Wang, and Hongke Zhang. Distributed Network Intrusion Detection System in Satellite-Terrestrial Integrated Networks Using Federated Learning. *IEEE Access*, 8:214852–214865, 2020.
10. Dezhi Han, Yujie Zhu, Dun Li, Wei Liang, Alireza Souri, and Kuan-Ching Li. A blockchain-based auditable access control system for private data in service-centric iot environments. *IEEE Transactions on Industrial Informatics*, 18(5):3530–3540, 2021.
11. Jiatao Li, Dezhi Han, Dun Li, and Hongzhi Li. Blockchain and OR Based Data Sharing Solution for Internet of Things. In Jiachi Chen, Bin Wen, and Ting Chen, editors, *Blockchain and Trustworthy Systems*, pages 116–127, Singapore, 2024. Springer Nature.
12. Ao Di, Shi Ruisheng, Lina Lan, and Lu Yueming. On the Large-Scale Traffic DDoS Threat of Space Backbone Network. In *2019 IEEE 5th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS)*, pages 192–194, Washington, DC, USA, May 2019. IEEE.
13. Imran Ashraf, Manideep Narra, Muhammad Umer, Rizwan Majeed, Saima Sadiq, Fawad Javaid, and Nouman Rasool. A Deep Learning-Based Smart Framework for Cyber-Physical and Satellite System Security Threats Detection. *Electronics*, 11(4):667, February 2022.
14. Zhang Wen-bo, Sun Peigen, Liu Zhi-guo, and Xu Haifeng. An intrusion detection model for satellite network. In *2010 2nd IEEE International Conference on Information Management and Engineering*, pages 167–170, Chengdu, China, 2010. IEEE.
15. Wei Guo, Jin Xu, Yukui Pei, Liuguo Yin, Chunxiao Jiang, and Ning Ge. A Distributed Collaborative Entrance Defense Framework Against DDoS Attacks on Satellite Internet. *IEEE Internet of Things Journal*, 9(17):15497–15510, September 2022.
16. Ahmad Taher Azar, Esraa Shehab, Ahmed M. Mattar, Ibrahim A. Hameed, and Shaimaa Ahmed Elsaid. Deep Learning Based Hybrid Intrusion Detection Systems to Protect Satellite Networks. *Journal of Network and Systems Management*, 31(4):82, October 2023.
17. Jianlong Zhu and ChunFeng Wang. Satellite Networking Intrusion Detection System Design Based on Deep Learning Method. In Qilian Liang, Jiasong Mu, Min Jia, Wei Wang, Xuhong Feng, and Baoju Zhang, editors, *Communications, Signal Processing, and Systems*, Lecture Notes in Electrical Engineering, pages 2295–2304, Singapore, 2019. Springer.
18. Bo Zhang, Hanwen Zhang, Gengming Zhao, and Jie Lian. Constructing a PM2.5 concentration prediction model by combining auto-encoder with Bi-LSTM neural networks. *Environmental Modelling & Software*, 124:104600, February 2020.
19. Hongzhi Li, Dezhi Han, and Mingdong Tang. A privacy-preserving storage scheme for logistics data with assistance of blockchain. *IEEE Internet of Things Journal*, 9(6):4704–4720, 2021.
20. Adel Binbusayyis and Thavavel Vaiyapuri. Unsupervised deep learning approach for network intrusion detection combining convolutional autoencoder and one-class svm. *Applied Intelligence*, 51(10):7094–7108, 2021.
21. Chongqing Chen, Dezhi Han, and Chin-Chen Chang. Caan: Context-aware attention network for visual question answering. *Pattern Recognition*, 132:108980, 2022.

22. Chongqing Chen, Dezhi Han, and Chin-Chen Chang. Mpcct: Multimodal vision-language learning paradigm with context-based compact transformer. *Pattern Recognition*, 147:110084, 2024.
23. Yuhua Yin, Julian Jang-Jaccard, Wen Xu, Amardeep Singh, Jinting Zhu, Fariza Sabrina, and Jin Kwak. Igrf-rfe: a hybrid feature selection method for mlp-based network intrusion detection on unsw-nb15 dataset. *Journal of Big data*, 10(1):15, 2023.
24. Lijun Xiao, Dezhi Han, Ce Yang, Jiahong Cai, Wei Liang, and Kuan-Ching Li. Ts-dp: An efficient data processing algorithm for distribution digital twin grid for industry 5.0. *IEEE Transactions on Consumer Electronics*, 2023.
25. Nour Moustafa and Jill Slay. Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set). In *2015 military communications and information systems conference (MilCIS)*, pages 1–6, 2015.

Shuxin Shi is currently pursuing the PhD degree with Shanghai Maritime University. His research interests include network security and machine learning.

Bing Han received the B.S. and Ph.D. degrees from the Department of Electrical Engineering, Dalian University of Technology, Dalian, China, in 2003 and 2009, respectively. He is currently a Researcher with the State Key Laboratory of Navigation and Safety Technology, Shanghai Ship and Shipping Research Institute, Shanghai, China. He has authored or coauthored over 40 peer-reviewed journal and conference papers. He was also sponsored by the Shanghai Talent Development Fund and Shanghai Rising-Star Program in 2014 and 2015, respectively. His current research interests include intelligent ship and intelligent systems and their applications, the design of deep sea dynamic positioning control systems.

Zhongdai Wu received a bachelor's degree in management information systems from Dalian Maritime University, China, M.B.A. degree from Tongji University, China. and Ph.D. degree in business management from Tongji University. Currently, he is working at COSCO SHIPPING TECHNOLOGY CO.,LTD., as Vice General Manager & CIO and also at Shanghai Institute of Navigation as vice president. His main research specialized in cloud computing, information system security, big data, Marine satellite communications and navigation, machine learning and blockchain.

Dezhi Han received the bachelor's degree from Hefei University of Technology, Hefei, China, in 1990, and the master's and Ph.D. degrees from Huazhong University of Science and Technology, Wuhan, China, in 2001 and 2005, respectively. He is currently a Professor of Computer Science and Engineering with Shanghai Maritime University, Shanghai, China. His research interests include cloud computing, mobile networking, wireless communication, and cloud security.

Huafeng Wu received the Ph.D. degree in computer science from Fudan University, Shanghai, China, in 2009, the master's degree in traffic information engineering and control from Dalian Maritime University, Dalian, China, in 2004. He was a Postdoctoral Research Fellow with Carnegie Mellon University, Pittsburgh, PA, USA, from 2008 to 2009, and also a Visiting Scholar with Shanghai Jiao Tong University, Shanghai, from 2012 to

2013. He is currently a Professor and a Ph.D. Supervisor with the Merchant Marine College, Shanghai Maritime University, Shanghai. His research interests include the Internet of Shipping, Internet of Things, and wireless sensor networks. Prof. Wu also serves as an Editorial Board Member for the Computer Communications.

Xiaojun Mei received the B.E. degree in navigation technology (excellent engineer education) and the Ph.D. degree in traffic information engineering and control from Shanghai Maritime University, Shanghai, China, in 2016 and 2021, respectively. He was a Postdoctoral Fellow with Shanghai Maritime University from November 2021 to October 2023. From September 2019 to September 2020, he was a visiting Ph.D. student with the Institute for Systems and Robotics, Instituto Superior Te´cnico, University of Lisbon, Lisbon, Portugal. He is currently an Associate Professor with the Merchant Marine College, Shanghai Maritime University and also with the State Key Laboratory of Maritime Technology and Safety, Shanghai Ship and Shipping Research Institute, Shanghai. His current interests include localization in marine/ocean/underwater wireless sensor networks and path planning for AUVs/ASVs.

Received: April 01, 2024; Accepted: July 22, 2024.

Computer Science and Information Systems 21(4):1411–1433 https://doi.org/10.2298/CSIS240328040K

Sentiment Polarity Analysis of Love Letters: Evaluation of TextBlob, Vader, Flair, and Hugging Face Transformer

Gaganpreet Kaur[1,⋆], Amandeep Kaur[1], Meenu Khurana[2], and Robertas Damaševičius[3]

[1] Chitkara University Institute of Engineering and Technology
Chitkara University, Rajpura, Punjab
gaganpreet.kaur@chitkara.edu.in
amandeep@chitkara.edu.in
[2] Chitkara University School of Engineering and Technology,
Chitkara University,Baddi, Himachal Pradesh
meenu.khurana@chitkara.edu.in
[3] Vytautas Magnus University
robertas.damasevicius@vdu.lt

Abstract. Sentiment analysis is the task of computationally identifying and quantifying the emotions and opinions expressed in text. However, existing sentiment analysis tools, while increasingly sophisticated, face challenges when applied to complex and personal domains such as love letters. This study investigates the performance and accuracy of four popular Python libraries for sentiment analysis (TextBlob, Vader, Flair, and Hugging Face Transformer) in determining the polarity and intensity of sentiments in love letters. A corpus of 300 love letters was collected and randomly sampled to provide 500 sentences for analysis. Due to the lack of labelled data, human experts participated in evaluating the quality and accuracy of the sentiment annotations.Inter-rater agreements were computed among four judges across randomly sampled sentence lots in two distinct blind rounds. The results reveal varying degrees of effectiveness and agreement among sentiment analysis tools (TextBlob, Vader, Flair, and Hugging Face) and human judges, with Cohen's Kappa values showing low to moderate agreement (ranging from 0.09 to 0.77), and each tool demonstrating unique strengths—Vader excelling in sentiment intensity and Flair with Hugging Face better at contextual nuances—in handling the emotional complexity of the texts. The study also highlights limitations and proposes some custom metrics for evaluating sentiment analysis tools in the context of love letters, such as tenderness index, passion quotient, nostalgia score, and others. The findings contribute to the emerging field of sentiment analysis and provide insights for developing natural language models better suited for personal and emotionally charged domains.
Keywords: Sentiment Analysis, Polarity Analysis, Natural Language Processing.

1. Introduction

Opinion mining, or sentiment analysis, is the process of computationally analysing text to identify the dominant feelings that are conveyed. Sentiment analysis in writing is ex-

⋆ Corresponding Author

tremely important for a variety of applications, from social media interactions and consumer reviews to literary analysis and interpersonal communication [9, 16, 22, 23, 29, 30]. In 2002, the first publication on sentiment analysis using computer-based methods was published. It was based on product reviews, which it used an unsupervised Pointwise Mutual Information - Information Retrieval algorithm to categorize as thumb up or down for positive and negative sentiment. [26].In the process of automating sentiment analysis, researchers and practitioners have created and utilized a number of algorithms, each with specific advantages and disadvantages [18] [17]. Social networking sites have become essential resources for exchanging feelings with people all over the world as a result of the Internet's explosive growth. Many people use text, photos, music, and video to express their ideas or perspectives. However, text communication through network media on the web might be a little overwhelming. Every second, social networking websites produce a significant amount of unstructured data on the Internet. To comprehend human behavior, data analysis must begin as soon as it is generated. Sentiment analysis can assist with this by identifying polarity in messages [20] [10].

The study begins with a thorough overview of related literature, outlining the present state of sentiment analysis algorithms as we proceed on our comparison journey. The technique is then covered in detail, including the dataset that was used, the evaluation measures that were used, and the particular setups for each algorithm. The next sections provide a thorough examination of each algorithm, contrasting its advantages and disadvantages in relation to sentiment analysis. Lastly, we present conclusions based on our research, providing information about the algorithmic decisions that best satisfy the requirements of sentiment analysis in highly emotive material. By conducting this research, we hope to add significant knowledge to the emerging field of sentiment analysis and assist practitioners in choosing algorithms that are appropriate for their particular use cases [28] [19].

Existing sentiment analysis tools, while increasingly sophisticated, face challenges when applied to complex and personal domains such as love letters. This study investigates the responses and accuracy of four popular tools (TextBlob, Vader, Flair, and Hugging Face Transformer) in determining the sentiment of love letters, specifically focusing on sentences.

- Accuracy of Sentiment Categorisation: Can these tools effectively categorise love letter sentences into neutral, positive, negative, and compound sentiment categories with high accuracy?
- Human-based Evaluation: In the absence of readily available labeled data, how does the sentiment analysis of these tools compare to the judgments of human experts on love letter language?
- Tool Strengths and Limitations: What are the specific strengths and limitations of each tool when analyzing the emotional complexity of love letters?
- Applicability and Model Development: Can the insights gained from this study be used to guide the development of natural language models better suited for sentiment analysis in personal texts like love letters?

By addressing these questions, this study aims to establish the suitability of existing sentiment analysis tools for personal and emotionally charged domains, paving the way for more accurate natural language models in the future [31, 32].

The study makes contributions to the field of sentiment analysis by rigorously evaluating the performance of four prominent sentiment analysis tools—TextBlob, Vader, Flair, and Hugging Face Transformer—specifically within the context of love letters, a complex and emotionally charged type of text. It introduces novel metrics such as the tenderness index, passion quotient, and nostalgia score, which are designed to measure subtle emotional nuances more effectively. The findings highlight each tool's strengths and limitations, revealing a varying degree of accuracy and the ability to handle the intricate emotional content of personal correspondence. These insights are crucial for advancing the application of sentiment analysis technologies in personal and nuanced domains, providing a foundation for future enhancements and tool customizations tailored to specific textual analyses. ‘

The novelty of this study lies in its focused examination of sentiment analysis tools specifically within the domain of love letters—a unique and emotionally nuanced form of personal communication. This is the first study, to the best of the authors' knowledge, to introduce and apply custom metrics such as the tenderness index, passion quotient, and nostalgia score, which are specifically designed to assess and quantify the subtle emotional nuances found in such texts. These innovations address the gap in existing sentiment analysis research by tailoring evaluation techniques to the specific challenges of analyzing deeply personal and emotional content, thereby expanding the applicability and precision of sentiment analysis tools in handling complex, non-standard text datasets.

The paper has VI Sections. Sections I and II covers the Introduction and Literature review. Section III and IV shows the Methodology and Results Discussions on this research. Section V and VI describe the limitations and detailed conclusion of the analysis.

2. Literature Review

The paper compares sentiment analysis technologies using rule-based and machine learning methodologies, revealing that tool performance and strategy don't always impact accuracy, but low accuracy in small datasets necessitates further analysis. [2] [13]. In reviewing recent advancements in sentiment analysis across different domains and modalities, we see a variety of innovative approaches that leverage modern computational techniques to enhance performance and adaptability. Firstly, Li et al. [22] and Karayiğit et al. [16] both implement BERT-based models, but in distinctly different contexts—Li et al. focus on extracting complex aspect-category-opinion-sentiment relationships for implicit sentiment analysis, demonstrating a marked improvement in F1 and recall scores, whereas Karayiğit et al. apply BERT to analyze COVID-19 related sentiments on social media, achieving superior classification accuracy over traditional and other deep learning models. Yang et al. [33] introduce a tri-modal model that incorporates contrastive learning and a transformer architecture to optimize multimodal sentiment analysis. This approach addresses alignment and semantic discrepancies across text, audio, and visual data, showing notable improvements on benchmark datasets.

In the realm of recommendation systems, Gu et al. [8] and Rosewelt et al. [29] blend sentiment analysis with graph neural networks and optimization algorithms, respectively, to refine e-commerce recommendations based on user sentiment extracted from reviews, which both show substantial enhancements in recommendation accuracy. Gunasekar and

Thilagamani [9] focus on cross-domain sentiment analysis by developing a network that integrates aspect and domain-invariant features, resulting in a model that outperforms traditional approaches in adapting to new domains without loss of sentiment analysis accuracy. Shi et al. [30] and Liang et al. [23] both extend the capabilities of sentiment analysis models through innovative learning strategies. Shi et al. introduce an adaptive prompt-based method that utilizes contrastive learning to effectively handle few-shot learning scenarios, while Liang et al. develop a multi-channel model that integrates pre-training mechanisms to better capture nuanced textual information from user-generated content.

The field of sentiment analysis has seen significant advancements through the development and application of rule-based, machine learning, and hybrid methodologies.

Rule-based sentiment analysis systems rely on manually crafted rules to identify and classify sentiments based on the presence of certain keywords, phrases, or patterns that have predefined sentimental values. Liu and Haig [24] proposed the use of fuzzy rule-based systems for interpretable sentiment analysis, highlighting the approach's alignment with the inherent uncertainty of language and its contribution to interpretability. Hutto and Gilbert [11] introduced VADER, a parsimonious rule-based model designed specifically for social media text, showcasing its effectiveness by outperforming individual human raters and various benchmarks in sentiment analysis tasks. Rule-based approaches offer interpretability and straightforward implementation but may lack flexibility and comprehensiveness.

The advent of machine learning, especially deep learning, has significantly advanced sentiment analysis by automating the extraction of complex features from text. Do et al. [6] provided a comprehensive review of deep learning techniques for aspect-based sentiment analysis, underscoring their ability to capture both syntactic and semantic features without the need for manual feature engineering. Similarly, Zhang et al. [34] surveyed deep learning applications in sentiment analysis, illustrating the technique's success across various sentiment analysis tasks due to its sophisticated data representation learning capabilities. Machine learning methodologies, particularly deep learning, provide powerful tools for capturing the complexities of language and sentiment but can suffer from a lack of interpretability and require substantial computational resources. Traditional and deep learning approaches are directly compared in the works of Kapočiūtė-Dzikienė et al. [15] and Pratibha et al. [28], with the former testing these methods on Lithuanian text sentiment analysis and finding traditional techniques slightly outperforming deep learning methods, while the latter curates a multimodal dataset for analyzing sentiments in tweets related to war and peace, demonstrating the growing importance of comprehensive datasets for deep pragmatic analysis.

Hybrid systems aim to combine the interpretability and simplicity of rule-based approaches with the powerful feature extraction and prediction capabilities of machine learning models. Chikersal et al. [4] developed SeNTU, a system that integrates a rule-based classifier with supervised learning (specifically, Support Vector Machines) for Twitter sentiment analysis, demonstrating how rules can refine machine learning predictions. Prabowo and Thelwall [27] combined rule-based classification, supervised learning, and machine learning into a novel method that improves classification effectiveness, showcasing the potential of hybrid approaches in enhancing sentiment analysis performance. Hybrid systems emerge as a promising middle ground, leveraging the strengths of both

approaches to achieve both high performance and interpretability in sentiment analysis tasks.

The discussed articles collectively illustrate the diverse applications of sentiment analysis in improving systems' ability to interpret human emotions accurately across various digital platforms and modalities.

3. Methodology

Based on the identified problems and aims of this study, In this section we explain the steps taken for obtaining them (Figure 3). The section begins explanation of how the dataset for this research was created.The next step is to preprocess the data so that it can be subjected to sentiment analysis using these four tools.

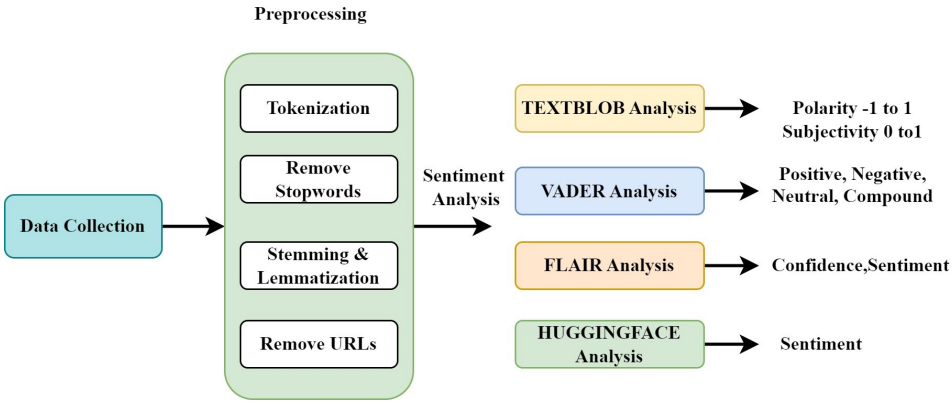


Fig. 1. Sentiment Analysis Flow

- **Dataset Collection**
  In this paper, we provide a carefully selected dataset of 300 love letters gathered from various sources, including well-known websites like Google and Quora. The wide and diverse spectrum of expressions of love captured in the dataset reflects the complex and multifaceted nature of this complex emotion. The dataset including these carefully prepared letters is available to the general public on Mendeley [DOI: 10.17632/rd5bjbnm35.3]. The dataset contains actual sentiments in each love letter, making it a vital resource for sentiment analysis research. We primarily assess how well four different Natural Language Processing (NLP) algorithms do in determining the sentiment conveyed in these love letters. This in-depth investigation seeks to further the field of sentiment analysis in the context of amorous emotions by shedding insight on how well these algorithms capture the complex intricacies of emotional language.
- **Preprocessing**
  For this research number of preprocessing steps were taken. It entails actions changing text to lowercase, tokenizing it, deleting punctuation, removing common words

(stop words), and stemming/lemmatization words to their simplest form. Managing numerals, special characters, contractions, HTML elements, and URLs are additional steps.The objective is to prepare text data for machine learning and sentiment analysis by cleaning and organizing it. [1].

– **Sentiment Analysis Tools** [5]
Sentiment analysis tools known as software applications are used to examine text exchanges to determine the tone, intent, and emotion of each message. Businesses can gain from these technologies in a number of ways, such as more efficient feedback management, enhanced issue solving, higher-quality goods and services, well-informed business decisions, and real-time feedback analysis.

1. **TextBlob** TextBlob is a rule-based sentiment analysis tool that adheres to natural language processing principles. Unlike certain other sentiment analysis tools, TextBlob does not rely on pre-trained models but rather employs a rule-based methodology to determine sentiment in textual data. Even anyone without programming skills can use it because of its simple and intuitive design. Its primary benefit is its ease of integration with Python, despite its limited support for multiple languages, including English. TextBlob's sentiment analysis provides basic emotion identification and categorizes text as positive, negative, or neutral depending on its polarity. Its handling of denial and complex contextual nuances is fairly limited, and it might not be able to portray the subtle emotional nuances in more complex texts. [12].TextBlob is often used in academic contexts and small-scale projects where a quick and easy sentiment analysis is sufficient, even though it is lightweight and simple to use. Additionally, real-time applications are a good fit for it. The tool's robust documentation and active community further contribute to its success in certain applications.
2. **VADER** For use with social media text and short content segments, the rule-based VADER (Valence Aware Dic-tionary and Sentiment Reasoner) Sentiment Analyzer is designed. VADER is particularly good at processing sentiments expressed through colloquial language because of its rule-based architecture and pre-trained lexicon, which enable it to discern sentiment intensity with remarkable accuracy. Because it is specifically made to account for context and negations, this tool is perfect for sentiment analysis in short, passionate texts like tweets and online comments. VADER offers a compound score that accounts for the overall strength of the sentence in addition to polarity scores. [14].On a scale that goes from extremely negative to highly positive, emotions are grouped. VADER is good at capturing sentiment in real-time and doesn't require fine-tuning, but because it is rule-based, it might not be as suitable for longer or more formal writings. However, because of its ease of use and effectiveness in gathering sentiment subtleties, VADER is a popular choice for sentiment analysis tasks in social media and short text contexts.
3. **Flair** This machine learning-based sentiment analysis tool stands out for its advanced features and contextual embeddings. Rather than using rule-based techniques, Flair use pre-trained contextual embeddings to capture the semantic meaning of words and phrases in context. This helps to increase the understanding of sentiments by enabling Flair to recognize the intricate relationships that exist between words and their surroundings. Flair's multilingual features enable it to offer extensive language support for a large range of documents. Additionally, it provides the fine-tuning option, which lets users adjust the model to suit certain domains or activities.Because of its strength

in managing negation, context, and emotion recognition, it is helpful for sentiment analysis jobs in a range of scenarios, including those involving love letters with complex emotional expressions. However, it needs a lot of processing power, especially for real-time applications, and depending on the specific embeddings used, the model size may vary. Flair is well-liked in academia and business because to its sophisticated features and adaptability for sentiment analysis tasks. [25].

4. **Hugging Face** It is a platform that houses a variety of cutting-edge natural language processing (NLP) models, including transformer-based models like BERT, GPT, and many others, even if it isn't a dedicated sentiment analysis tool.Because of their remarkable results on a range of NLP benchmarks, these models—which are available via the Hugging Face Transformers library—have emerged as the preferred options for sentiment analysis assignments. Hugging Face transformer-based models are characterized by their capacity to represent intricate contextual relationships in text, which makes them ideal for sentiment analysis of a wide range of complicated content, including love letters. By using deep learning techniques, these models frequently perform better than conventional rule-based or lexicon-based approaches [15]. Hugging Face platform offers a library of pre-trained models that span numerous languages and domains. Additionally, it facilitates fine-tuning, enabling customers to customize the models to meet their unique requirements. Because of the platform's comprehensive documentation, user-friendly interface, and a large community of contributors to its ecosystem, it is widely used in both academics and industry. Hugging Face is useful for practitioners looking for state-of-the-art solutions for sentiment analysis tasks, especially those involving love letters with rich emotional content because it uses transformer-based models to handle complex linguistic nuances. [3].
   Table[1] shows some important features of the above four tools.

Table 1. Features of Tools

Tools	Algo. Logic	Supportive Lang.	Pre trained	Fine tuning	Emo Recog.	Industry Acceptance
TextBlob	Rule-based	Limited	No	NA	Simple	Academic, Small-scale
Vader	Rule-based	English	Yes	NA	Sentiment intensity	Brief texts & social media
Flair	Machine Learning	multiple lang.	Yes	Yes	Advanced	Business and academia
Hugging face	Transformer-based	multiple lang.	Yes	Yes	Refined	Research and Business

– Sentiment Polarity Analysis of Love Letters

Python functions were developed to compute the polarity of the sentiments as shown in Table 2, which is the short view of the output. It contains the sentiment scores of each sentence (Segment_Content). The descriptions are as follows :

This dataset is used for the analyses of text sentiment through multiple tools. Each "Segmented_Content" segment receives a polarity score (-1 to +1) from TextBlob (measuring negativity/positivity) and a subjectivity score (0 to 1) from the same tool (measuring opinionatedness). VADER further categorises text as Positive, Negative, Neutral,

or with a ”Compound” score (-1 to +1) combining these proportions. Flair and Hugging Face models also provide sentiment analysis and confidence levels [2].

Table 2. Tool Polarity and Judge Score(Partial view)

Segment_Content	TB_Pol	TB_Sub	VD_Pos	VD_Neu	VD_Neg	VD_Comp	F_Sent	F_Confd	HF	J1_T1	J1_T2	J1_T3	J1_T4
Dear My Girl, Yes, I do comprehend that you are experiencing quite a bit of discomfort in your first workplace, ...	0.17	0.38	0.10	0.76	0.12	-0.22	N	0.99	N	1	1	0	1
People will not respect your time and will try to take your attention, most of the time without any malice in their hearts,....	0.12	0.42	0.09	0.87	0.02	0.67	N	0.99	N	1	1	0	1
It's possible that you'll have feelings of guilt because you didn't follow through with what you had planned to do.	0	1	0	0.9	0.1	-0.27	N	0.99	N	1	0	1	1
However, you are not to blame because you are a human being.Everyone gets their attention diverted.	0	0.1	0.12	0.88	0	0.25	P	0.82	N	1	0	0	0
In many respects, the fundamental split is between those individuals who are able to get back on track	0.33	0.39	0.09	0.85	0.05	0.20	P	0.90	N	1	1	0	1
You are going to get interrupted, but you have the option to keep what you say short.	0	0	0.16	0.78	0.05	0.61	P	0.99	P	0	1	1	1

In summary, the following methodology (Figure 2) has been implemented to obtain the goals of this study.

1. **Tool Selection and Parameter Configuration:** Selected tools for evaluation include: TextBlob, Vader, Flair, and Hugging Face Transformer and configuration of the relevant parameters for each tool was done to ensure consistency and fairness in the analysis.
2. **Dataset Preparation:**
   - Collect a corpus of 300 love letters to serve as the dataset for analysis.
   - Randomly select 500 sentences from the love letters to provide a representative sample.
3. **Sentiment Analysis of Love Letters:** The TextBlob, Vader, Flair, and Hugging Face Transformer were used to analyze and categorize selected sentences into neutral, positive, negative, and compound sentiments.
4. **Human-based Evaluation:**
   - Engagement of human experts for two blind rounds to independently evaluate the sentiment of the same set of sentences.
   - Collection of annotations from human judges regarding the sentiment categories for comparison with tool-generated results.
   - Calculate Inter-Rater Agreements between the sentiment analysis tools and human judges using metrics Cohen's Kappa and analysis of the level of agreement/disagreement between tools and human experts.
5. **Strengths and Limitations Assessment:** Examination of the specific strengths and limitations of each sentiment analysis tool.

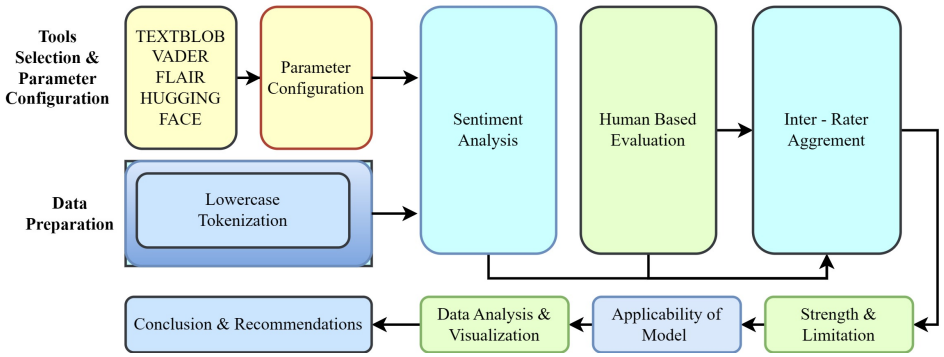


Fig. 2. Research Flow

6. **Applicability :** Based on the findings, derivation of the insights into the applicability of existing sentiment analysis tools for personal and emotionally charged domains like personal love letters, etc.
7. **Data Analysis and Visualization:**
   - Analyses of the collected data using statistical methods to derive quantitative insights.
   - Draw confusion matrices and histograms, to present the results in a comprehensible manner.
8. **Conclusion and Recommendations** for summarization of the study's findings, highlighting the suitability and limitations of sentiment analysis tools for love letters. In the next section, we discuss the outcome of the methodology applied.

4. Results & Discussion

Evaluating sentiment analysis tools in real-time NLP text data without a benchmark or a labeled dataset is hard due to the dependency of traditional supervised learning metrics on ground truth labels. However, other strategies can be used to approximate accuracy or evaluate performance, including Human expert evaluation. This involves using human annotators to assess the accuracy of sentiment predictions by comparing human annotations to the predictions made by each sentiment analysis tool. Hence, user feedback as a tool of evaluation was used in this research work. User feedback serves as an indirect measure of performance if sentiment analysis tools are integrated into an application or platform. Therefore, In this section, we discuss the outcome of the annotations that these tools assign to the segmented text of the love letters.

4.1. Objective Agreement Between the Tools :

This research work aims to evaluate sentiment analysis tools objectively and subjectively. We compare their agreement on polarity and relevance to human judgements in love letters. High agreement validates tool reliability, while disagreements expose limitations and biases, guiding model improvement. Sentiment can be complex, so disagreements

may reflect natural language complexity and automated emotion capture challenges. By understanding these discrepancies, we can refine models, improve context understanding, and consider factors like irony or cultural references that influence sentiment. This section focuses on tool agreement levels on the same content.

Agreement of Random Samples (100) In this 100 sentences were randomly selected from a corpus of love letters to ensure an unbiased and representative analysis. This randomness is crucial as it prevents any sampling bias, guaranteeing that each sentence in the corpus has an equal chance of being chosen. Sentiment and polarity analyses were then conducted on these sentences using various Natural Language Processing (NLP) tools, including TextBlob, VADER, Flair, and Hugging Face. Each tool applies its methodology to evaluate the emotional tone of the text, yielding scores that indicate the sentiment's positivity or negativity. These scores were meticulously recorded in an Excel sheet for detailed analysis and comparison. The core of the study involved analysing the agreement and disagreement among these tools outcomes. This phase is essential to gauge the reliability and consistency of sentiment analysis methodologies. Statistical measures like Cohen's Kappa [21] have been used to quantify the agreement between the tool's sentiment assessments. The results of this analysis are critical; high agreement levels between tools could indicate their reliability, while significant disagreements might highlight the subjective nature of sentiment analysis or the diverse approaches of different tools. Ultimately, the study's structured approach, starting from the random selection of sentences to detailed sentiment analysis and inter-tool agreement assessment, offers a comprehensive understanding of the sentiments in the love letters.

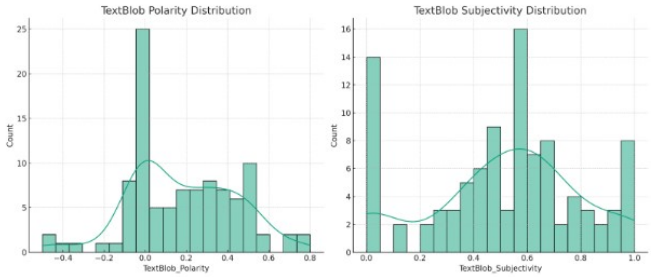


Fig. 3. Textblob Polarity and Subjectivity Distribution

- TextBlob Polarity: The polarity scores in Fig. 4 indicates that the majority of sentiments are positive, with a mean polarity of around 0.18. However, there is a wide distribution, with some sentiments being negative as well.
- TextBlob Subjectivity: The subjectivity scores in Fig 3. show a mean of about 0.51, suggesting a balance between objective and subjective sentiments in the dataset. The distribution is fairly uniform.
- VADER Scores: The VADER in Fig. 4 shows a higher distribution of neutral scores, with positive sentiments also being fairly common. Negative scores are less frequent,

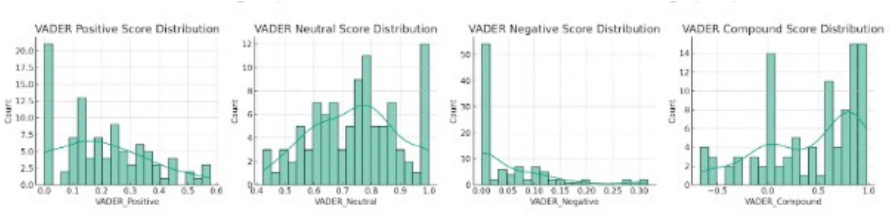


Fig. 4. Vader Score Distribution

and the compound score histogram reflects a skew towards positive compound sentiment.

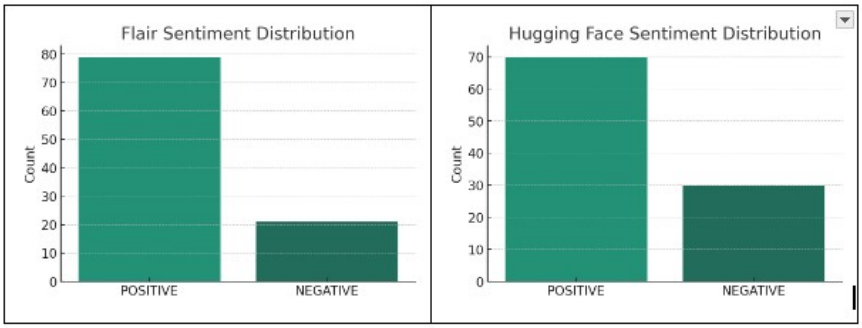


Fig. 5. Flair and Huggingface Distribution

- Flair Sentiment: The bar plot for Flair sentiment in Fig5. indicates that there is a predominance of one sentiment class over the other in the dataset.
- Hugging Face Sentiment: Similar to the Flair sentiment, the Hugging Face sentiment classifier in Fig. 4 shows a distribution skewed towards one sentiment class.

Using Cohen's Kappa score, the agreement with sentiment score from the four tools has been calculated.

The Cohen's Kappa Score Matrix (κ) is calculated to assess the agreement between two raters (or tools) who classify items into mutually exclusive categories. The formula for Cohen's Kappa is given by:

$$\kappa = \frac{p_o - p_e}{1 - p_e} \tag{1}$$

where: p_o is the relative observed agreement among raters, computed as the proportion of items upon which the raters agree. p_e is the hypothetical probability of chance agreement, calculated based on the probabilities of each rater randomly assigning items to categories. To calculate these probabilities, construct a $k \times k$ confusion matrix for k categories, where

the element at the i^{th} row and j^{th} column (n_{ij}) represents the number of items Rater 1 classified into category i and Rater 2 into category j. Then, p_o is calculated as:

$$p_o = \frac{1}{N} \sum_{i=1}^{k} n_{ii} \tag{2}$$

where n_{ii} are the diagonal elements of the confusion matrix (where raters agree), and N is the total number of items. The probability of chance agreement, p_e, is computed as:

$$p_e = \sum_{i=1}^{k} \left(\frac{\sum_{j=1}^{k} n_{ij}}{N} \right) \left(\frac{\sum_{j=1}^{k} n_{ji}}{N} \right) \tag{3}$$

where the first term inside the summation is the proportion of items Rater 1 assigns to category i, and the second term is the proportion of items Rater 2 assigns to the same category. Cohen's Kappa score ranges from -1 (perfect disagreement) to +1 (perfect agreement), with 0 indicating that the observed agreement is exactly what would be expected by chance.

The outcomes are displayed in Table 4.1 and Fig. 6

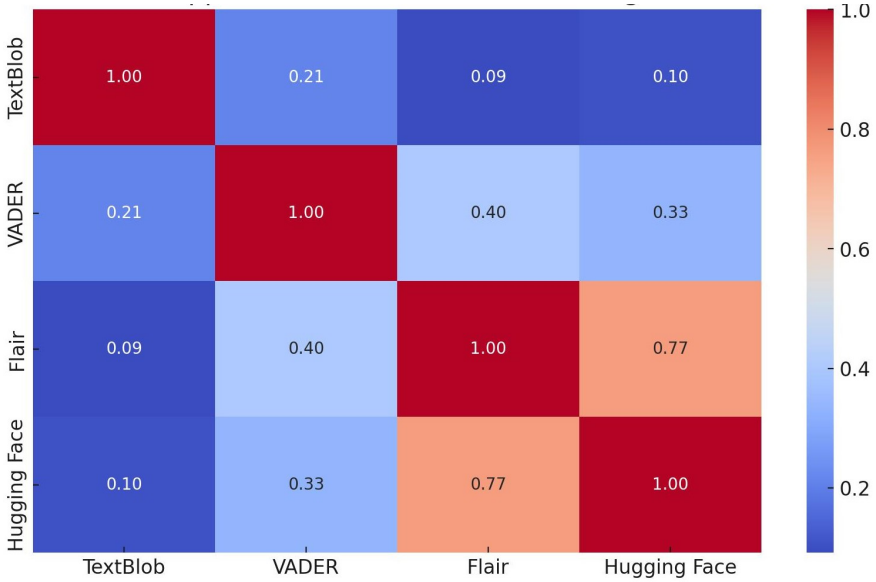


Fig. 6. Cohen's Kappa score Matrix for Sentiment Agreement

Agreement of Random Samples (200) As done before in the previous case of 100, 200 sentences were randomly selected from a corpus of love letters to ensure an unbiased and representative analysis for the second round of evaluation. In this case the following observations can be made.

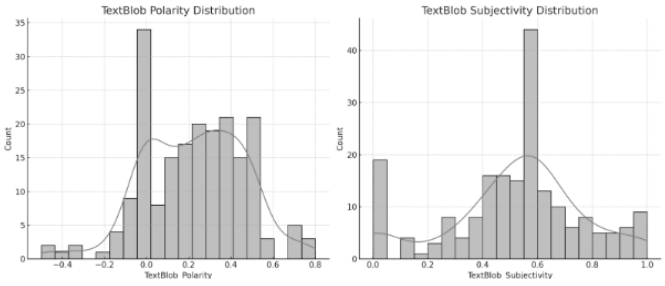


Fig. 7. Textblob Polarity and Subjectivity Distribution

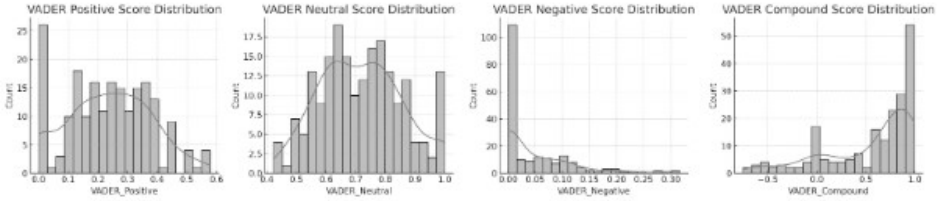


Fig. 8. Vader Score Distribution

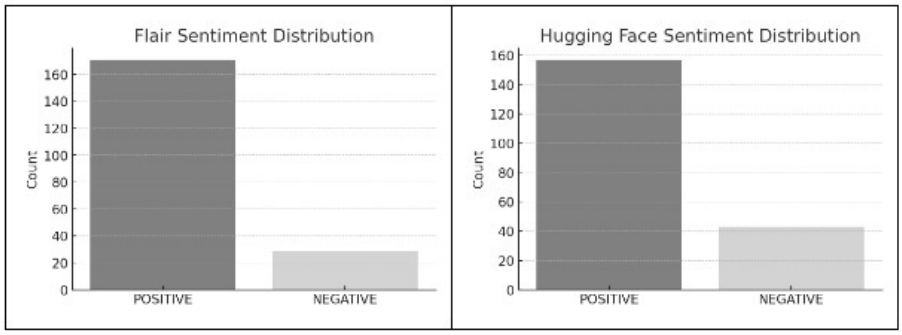


Fig. 9. Flair and HuggingFace sentiment Distribution

Table 3. Results of Cohen Kappa Score Matrix

TextBlob and VADER:	TextBlob and Flair	TextBlob and Hugging Face	VADER and Flair	VADER and Hugging Face	Flair and Hugging Face
K=0.21 K=0.21	K=0.09 K=0.09	K=0.10 K=0.10	K=0.40 K=0.40	K=0.33 K=0.33	K=0.77 K=0.77

- **TextBlob Polarity:** The polarity scores from TextBlob in Fig. 7 indicate a distribution with a slight skew towards positive sentiment.
- **TextBlob Subjectivity:** The subjectivity scores show a wide distribution, in Fig.7 indicating a mix of subjective and objective sentiments in the dataset.
- **VADER Scores:** The VADER sentiment analysis shows a variety of distributions for the positive, neutral, and negative scores. The compound score distribution indicates in Fig. 8 a skew towards positive sentiment overall.
- **Flair Sentiment:** The bar plot for Flair sentiment in Fig.9 shows the counts of each sentiment class.
- **Hugging Face Sentiment:** Similarly, the bar plot for Hugging Face sentiment in Fig. 9 shows the distribution of sentiment classes.
  The subsequent section of this analysis will focus on evaluating the extent of concordance between human expert judgments across different sentences from love letters.This examination is imperative to optimise and check the suitability of each tool's performance tailored to specific text genres such as love letter analysis.

4.2. Concordance Between Human Judges and Tools (Round 1 [100]) :

To analyse the concordance between human judges and tool scores, we will compute the correlation between the judges' ratings and the score of sentiments from the tools (TextBlob, Vader,and Flair). This will help us understand how well the sentiment analysis tools align with human judgement. For computations between the human judges themselves, we will calculate the correlation between the ratings of different judges to gauge the degree of agreement among them. This will give light on the consistency of human judgement across different individuals.

Concordance Between Human Judges and Tools Scores For this, the correlation between each judge's ratings and the sentiment score from TextBlob, VADER, and Flair is computed. Since the judge's ratings are binary, we will use methods suitable for binary data correlation with continuous data, such as point-biserial correlation (Figure 10).

- Correlations vary significantly across different judges and different tools in Fig.10 indicates that there is no consistent pattern of agreement or disagreement between human judges and automated sentiment analysis tools.
- TextBlob Polarity: Some judges show a weak to moderate positive correlation with TextBlob polarity, suggesting a slight agreement in assessing the overall sentiment (positivity/negativity) of the texts.

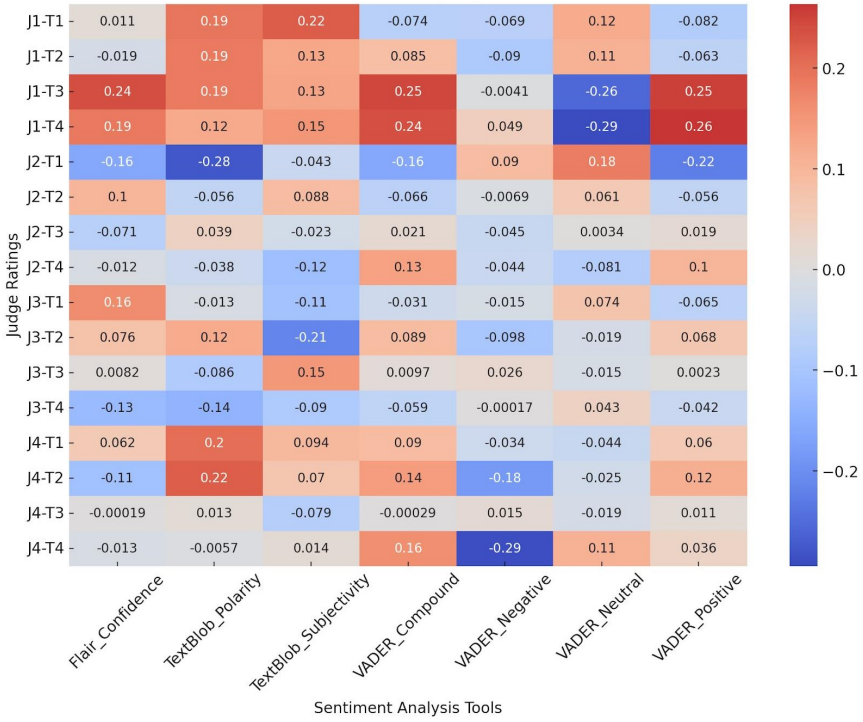


Fig. 10. Correlation between Human Judges and Tools

– VADER Scores: Correlations with VADER scores (positive, neutral, negative, compound) also vary. In some cases, there is a weak positive correlation, while in others, there is no significant correlation or even a weak negative correlation.
– Flair Confidence: The correlation with Flair's confidence in its sentiment predictions is generally low, indicating that the confidence level of the Flair model is not very aligned with human judgments.

Concordance Between Human Judges In this section, calculations regarding the correlation between the ratings of different judges are done (Figure 11) and for this correlating binary data, (Phi coefficient) is computed, [7] which is a measure of the association between two binary variables.

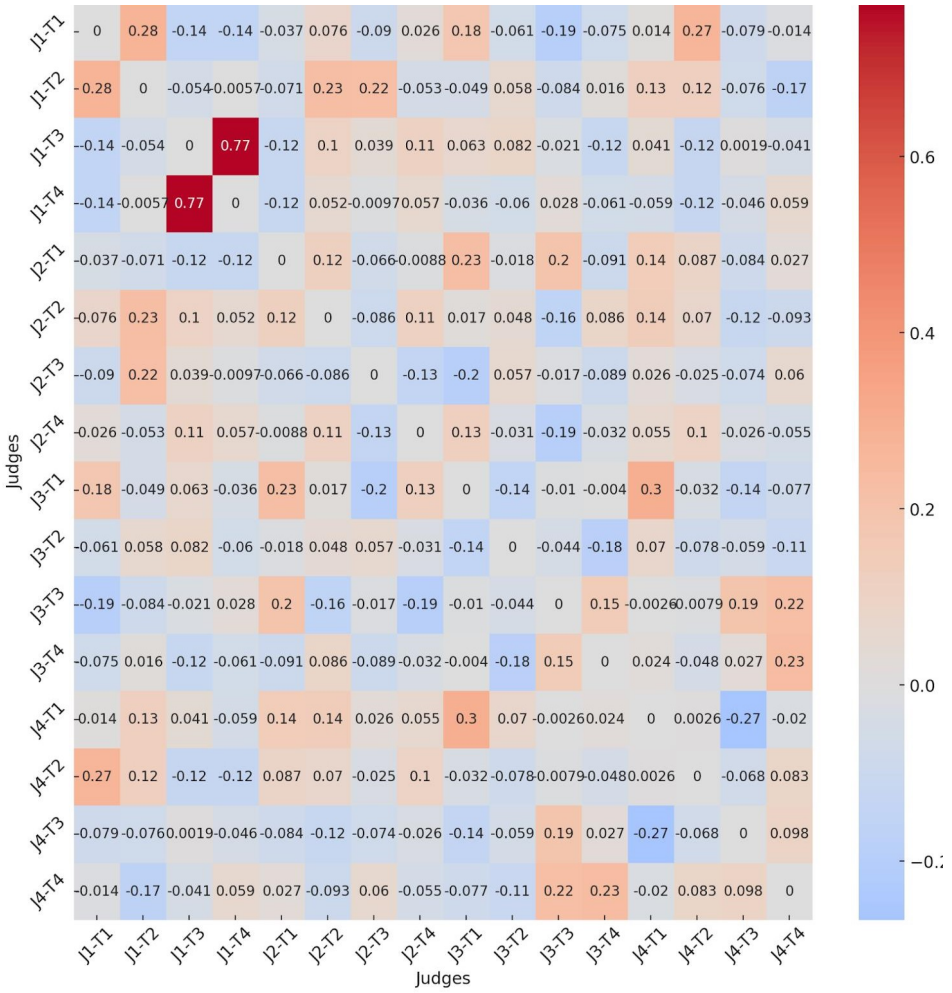


Fig. 11. Correlation between Human Judges

The Phi coefficient [7] values between pairs of judges are as follows:

– The correlations vary significantly, indicating differing levels of agreement between judges. However, in some cases, there is a moderate positive correlation, suggesting agreement between certain pairs of judges. At the same time, it is observed that many pairs of judges show low to negligible correlation, indicating a lack of consistent agreement in their ratings.
  item A few pairs show a high correlation in Fig.11 (e.g., 'J1-T3' and 'J1-T4' with a coefficient of 0.774), suggesting a strong agreement in their judgments and scoring.

4.3. Inferences from the Concordance Values

Concordance Between Judges and Tools

– There is no strong and consistent pattern of correlation between the judges' ratings and the sentiment analysis tool's scores. This suggests that there may be some level of agreement between human judges and automated tools, the concordance is not uniform across different tools or different judges.
– The variability in correlations suggests that sentiment analysis tools might interpret sentiment differently compared to human judgment, which can be due to the nuances and complexities of human language that automated tools may not fully capture.

Concordance Between Human Judges

– The varying levels of correlation between different pairs of judges highlight the subjective nature of sentiment analysis when performed by humans. Different judges may focus on different aspects of the text or interpret sentiments differently, leading to varying levels of agreement.
– The presence of both high and low correlations among judges pairs suggests that some judges may have similar criteria or thresholds for sentiment evaluation, others differ significantly in their judgement.

These findings demonstrate the challenges in sentiment analysis, both in terms of creating tools that can mimic human judgement and in achieving consistency among human evaluators. It highlights the need for clear guidelines and training for human judges and the importance of continuous improvement in sentiment analysis algorithms to better align with human interpretations.

4.4. Concordance Between Human Judges and Tools (Round II [200]):

In this round, 200 random sentences with their evaluations are analysed, similar to the previous round. It includes sentiment analysis scores from TextBlob, VADER, and Flair, along with evaluations from four judges.

Concordance Between Human Judges and Tools Scores in Round II The observations regarding the point-biserial correlation coefficients between the judge's ratings and the sentiment analysis tool scores in Round II are as follows:
It can be observed that generally, the Correlations are low, indicating a weak agreement between human judges and sentiment analysis tools. This is consistent across different judges and tools. TextBlob Polarity and Subjectivity: Some judges show a

slight positive correlation with TextBlob scores, but the correlation is generally weak. In VADER Scores, the Correlations with positive, neutral, negative, and compound scores vary and are generally low. In the case of Flair Confidence score analysis, the correlation with Flair's confidence level is also generally low, suggesting limited alignment with human judgments.

Concordance Between Human Judges in Round II The Phi coefficient values between pairs of judges in Round II are as follows:
The Variability in Correlations is similar to Round I, the correlations between judges in Round II vary, indicating differing levels of agreement. At the same time, it can be observed that some Moderate Correlations can be observed in case a few judge pairs show moderate positive correlations, suggesting agreement in their evaluations. Lastly, many Low Correlations can also be observed, clearly showing that many pairs of judges exhibit low to negligible correlation, indicating a lack of consistent agreement in their ratings. Based on the above observations, generally low correlations between human judges and analysis tools suggest that the tools' assessments of sentiment do not consistently align with human judgement. This might be due to the complexity of human language and sentiment, which automated tools might not fully capture.The variability in these correlations highlights the differences in how humans and machines interpret and evaluate sentiment. The variability in agreement among judges underscores the subjective nature of sentiment analysis. Different judges might prioritise different aspects of the text or interpret sentiments differently. The presence of both moderate and low correlations among judge pairs suggests varying criteria or thresholds for sentiment evaluation among different judges. The findings from Round II are inline with those from Round I, highlighting the challenges in achieving consistency in sentiment analysis, both among human evaluators and between human and automated tool assessments. This emphasises the guidelines for human sentiment analysis and continued improvement in automated sentiment analysis algorithms.
From the round I and round II of evaluation, it is abundantly clear that such tools are unsuitable for getting the sentimental value of the sentences embodied in the love letter. There is a necessity to customise and modify these tools to gain such availability.

5. Limitations and Future Applicability

Despite the significant advancements demonstrated in the reviewed sentiment analysis research, several limitations are evident, offering directions for future work. One recurrent challenge is the dependency on large, labeled datasets for training complex models. Although efforts like those described by Shi et al. [30] aim to mitigate this through adaptive prompt-based learning for few-shot scenarios, the generalization of such models across truly diverse or low-resource linguistic domains remains problematic. Moreover, while models such as those introduced by Yang et al. [33] improve multimodal sentiment analysis, the alignment of different modalities (text, audio, visual) and the effective fusion of such data without redundancy still pose considerable challenges.

The complexity of sentiment analysis models, especially those integrating deep learning and neural networks, often leads to issues with interpretability. Models like the RWESA-

GNNR proposed by Gu et al. [8] integrate sentiment analysis with graph neural networks to enhance recommendation systems, yet the interpretability of such models is not well addressed, which is crucial for trust and reliability in real-world applications. Furthermore, the performance of sentiment analysis tools across different cultural or demographic segments has not been thoroughly explored, which could affect the applicability and fairness of these tools in global or multicultural settings.

Future research could focus on developing more robust models that require fewer data yet are capable of high generalization across domains. This includes exploring unsupervised or semi-supervised methods that could leverage unlabeled data more effectively. There is also a pressing need for the development of models that provide better interpretability and explainability, which are essential for applications in fields such as healthcare, finance, and law where understanding model decisions is crucial. Additionally, enhancing the ability of sentiment analysis tools to handle multimodal data more efficiently, possibly through the development of novel neural network architectures or more sophisticated data fusion techniques, could significantly improve the accuracy and applicability of sentiment assessments. Lastly, the ethical implications and biases in sentiment analysis need rigorous examination to ensure these tools are fair and equitable across all user demographics.

Table 4. Sentiment Analysis Metrics for Love Letters

Metric	Definition	Calculation
Tenderness Index	A metric that measures the tool's accuracy in identifying and quantifying expressions of tenderness in love letters.	Evaluate the frequency and intensity of words associated with tenderness (e.g., gentle, affectionate, caring) in the sentiment predictions.
Passion Quotient	Assess the tool's effectiveness in capturing the passionate undertones within love letters.	Analyze the presence and intensity of words related to passion (e.g., fiery, intense, desire) in the sentiment predictions.
Nostalgia Score	A metric indicating how well the sentiment analysis tool captures nostalgic sentiments in love letters.	Consider the accuracy of identifying words and phrases associated with nostalgia (e.g., reminiscing, longing, memories) in the sentiment predictions.
Romantic Consistency	Measure the tool's ability to maintain sentiment consistency in segments of text expressing romantic feelings.	Evaluate the tool's performance in correctly predicting sentiment changes within the context of a romantic narrative.
Depth of Emotional Understanding	Assess the tool's capability to grasp the depth and complexity of emotions conveyed in the love letters.	Develop a metric that considers the tool's accuracy in capturing subtle shifts in sentiment intensity and emotion complexity.
Personalization Accuracy	Gauge how well the sentiment analysis tool adapts to and accurately reflects the unique personalization in individual love letters.	Evaluate the tool's precision in identifying sentiment nuances specific to the personal writing style and expressions of the letter.
Sentiment Diversity	Measure the variety and range of sentiments detected by each tool within the collection of love letters.	Assess the tool's performance in identifying a diverse array of sentiments, reflecting the multifaceted nature of love.

6. Conclusion

From the thorough analysis conducted during Round I and Round II evaluations, a clear pattern emerges, underscoring the inadequacy of existing sentiment analysis tools in capturing the nuanced sentimental value embedded within love letters. This revelation prompts a compelling argument for the imperative need to customise and modify these tools to enhance their suitability for discerning sentiments in the context of personal and emotionally charged texts. The key foundation of this argument lies in the examination of Phi coefficient values between pairs of judges in Round II. The Variability in Correlations, akin to Round I, persists in Round II, indicating divergent levels of agreement among the judges.

This variability is a crucial indicator of the challenges in achieving consensus, showcasing that different pairs of judges exhibit inconsistent levels of agreement in their evaluations.

A critical observation arises from the presence of Moderate Correlations in specific judge pairs, implying a degree of agreement in their assessments. However, the overarching trend reveals a multitude of Low Correlations, signifying that numerous judge pairs display low to negligible correlation. This discrepancy directly points to the lack of consistent alignment between human judges and sentiment analysis tools, implying that these tools' assessments do not reliably coincide with human judgments. The observed variability in correlations sheds insight on the complex structure of human language and sentiment, which automated tools may struggle to fully encapsulate. The disparities highlight the differing interpretations and evaluations of sentiment between humans and machines, emphasizing the nuanced and complex nature of emotional expression in love letters. Moreover, the consistency in findings between Round II and Round I accentuates the persistent challenges in achieving reliability and uniformity in sentiment analysis. This consistency underscores the ongoing struggle, both among human evaluators and between humans and automated tools, emphasizing the inherent difficulty in capturing the multifaceted emotional nuances present in love letters. Consequently, the argument strengthens the case for the necessity of clear guidelines for human sentiment analysis, facilitating a standardized approach and minimizing discrepancies among human judges. Simultaneously, it emphasizes the imperative for continuous enhancements in automated sentiment analysis algorithms. The presence of varying criteria and thresholds for sentiment evaluation among different judges, as suggested by the mixed correlations, reinforces the subjective nature of sentiment analysis and the need for further refinement in both manual and automated approaches. In essence, the argument emphasizes the pivotal role of customization and modification of sentiment analysis tools, drawing attention to the complex and subjective nature of sentiment analysis in the domain of personal and emotional expressions within love letters.

Table 5. Variable & Abbreviations

Variable	Abbreviations
Textblob_Polarity	TB_Pol
Textblob_Subjectivity	TB_Sub
Vader_Positive	VD_Pos
Vader_Neutral	VD_Neu
Vader_Negative	VD_Neg
Vader_Compound	VD_Comp
Flair_Sentiment	F_Sent
Flair_Confidence	F_Confd
Huggingface	HF
Judge	J
Tool	T

References

1. Angiani, G., Ferrari, L., Fontanini, T., Fornacciari, P., Iotti, E., Magliani, F., Manicardi, S., et al.: A comparison between preprocessing techniques for sentiment analysis in twitter. (2016)
2. Borrelli, F.M., Challiol, C.: Comparing and evaluating tools for sentiment analysis (2023)
3. Chhabra, A., Chaudhary, K., Alam, M.: Exploring Hugging Face Transformer Library Impact on Sentiment Analysis: A Case Study. Auerbach Publications (2024)
4. Chikersal, P., Poria, S., Cambria, E.: Sentu: Sentiment analysis of tweets by combining a rule-based classifier with supervised learning. SemEval 2015 Task 10 pp. 647–651 (2015)
5. Dhiman, P., Kaur, A.: Text pre-processing techniques in addressing fake news. In: Computational Methods in Science and Technology: Proceedings of the 4th International Conference on Computational Methods in Science & Technology (ICCMST 2024), 2–3 May 2024, Mohali, India. vol. 1, pp. 142–148. CRC Press (2024)
6. Do, H.H., Prasad, P., Maag, A., Alsadoon, A.: Deep learning for aspect-based sentiment analysis: A comparative review. Expert Syst. Appl. 118, 272–299 (2019)
7. Fleiss, J.L.: Measuring agreement between two judges on the presence or absence of a trait. Biometrics pp. 651–659 (1975)
8. Gu, J., Xu, Y., Liu, W.: Rwesa-gnnr: Fusing random walk embedding and sentiment analysis for graph neural network recommendation. Information Technology and Control 53(1), 146 – 159 (2024)
9. Gunasekar, M., Thilagamani, S.: Improved feature representation using collaborative network for cross-domain sentiment analysis. Information Technology and Control 52(1), 100 – 110 (2023)
10. Gupta, D., Madhukar, M., et al.: Analysis of machine learning approaches for sentiment analysis of twitter data. Journal of Computational and Theoretical Nanoscience 17(9-10), 4535–4542 (2020)
11. Hutto, C.J., Gilbert, E.: Vader: A parsimonious rule-based model for sentiment analysis of social media text. In: Proceedings of the International AAAI Conference on Web and Social Media (2014)
12. Illia, F., Eugenia, M.P., Rutba, S.A.: Sentiment analysis on pedulilindungi application using textblob and vader library 2021(1), 278–288 (2021)
13. Ilyas, S.H.W., Soomro, Z.T., Anwar, A., Shahzad, H., Yaqub, U.: Analyzing brexit's impact using sentiment analysis and topic modeling on twitter discussion. In: The 21st Annual International Conference on Digital Government Research. pp. 1–6 (2020)
14. Joharee, I.N., Hashim, N.N.W.N., Shah, N.S.M.: Sentiment analysis and text classification for depression detection. Journal of Integrated and Advanced Engineering (JIAE) 3(1), 65–78 (2023)
15. Kapočiūtė-Dzikienė, J., Damaševičius, R., Woźniak, M.: Sentiment analysis of lithuanian texts using traditional and deep learning approaches. Computers 8(1) (2019)
16. Karayiğit, H., Akdagli, A., Acı, : Bert-based transfer learning model for covid-19 sentiment analysis on turkish instagram comments. Information Technology and Control 51(3), 409 – 428 (2022)
17. Kaur, G., Kaur, A., Khurana, M., et al.: A review of opinion mining techniques. ECS Transactions 107(1), 10125 (2022)
18. Kaur, G., Kaur, A., Khurana, M., et al.: A stem to stern sentiment analysis emotion detection. In: 2022 10th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO). pp. 1–5. IEEE (2022)
19. Kumar, A., Hooda, S., Gill, R., Ahlawat, D., Srivastva, D., Kumar, R.: Stock price prediction using machine learning. In: 2023 International Conference on Computational Intelligence and Sustainable Engineering Solutions (CISES). pp. 926–932. IEEE (2023)

20. Kumar, R., Sharma, C.M., Chariar, V.M., Hooda, S., Beri, R.: Emotion analysis of news and social media text for stock price prediction using svm-lstm-gru composite model. In: 2022 International Conference on Computational Intelligence and Sustainable Engineering Solutions (CISES). pp. 329–333. IEEE (2022)
21. Kvålseth, T.O.: Note on cohen's kappa. Psychological reports 65(1), 223–226 (1989)
22. Li, J., Li, X., Du, Y., Fan, Y., Chen, X., Huang, D.: An aspect-category-opinion-sentiment quadruple extraction with distance information for implicit sentiment analysis. Information Technology and Control 52(2), 445 – 456 (2023)
23. Liang, S., Jin, J., Du, W., Qu, S.: A multi-channel text sentiment analysis model integrating pre-training mechanism. Information Technology and Control 52(2), 263 – 275 (2023)
24. Liu, H., Haig, E.: Fuzzy rule based systems for interpretable sentiment analysis. In: 2017 Ninth International Conference on Advanced Computational Intelligence (ICACI). pp. 129–136 (2017)
25. Mahrukh, R., Shakil, S., Malik, A.S.: Sentiments analysis of fmri using automatically generated stimuli labels under naturalistic paradigm. Scientific Reports 13(1), 7267 (2023)
26. Marsanich, G.: Comparing Predictions of YouTube Video Like to Dislike Ratios Using Sentiment Analysis Tools. Ph.D. thesis, Tilburg University (2022)
27. Prabowo, R., Thelwall, M.: Sentiment analysis: A combined approach. J. Informetrics 3, 143–157 (2009)
28. Pratibha, Kaur, A., Khurana, M., Damaševičius, R.: Multimodal hinglish tweet dataset for deep pragmatic analysis. Data 9(2) (2024)
29. Rosewelt, L.A., Raju, D.N., Sujatha, E.: A new sentiment and fuzzy aware product recommendation system using weighted aquila optimization and grnn in e-commerce. Information Technology and Control 52(3), 617 – 637 (2023)
30. Shi, C., Zhai, R., Song, Y., Yu, J., Li, H., Wang, Y., Wang, L.: Few-shot sentiment analysis based on adaptive prompt learning and contrastive learning. Information Technology and Control 52(4), 1058 – 1072 (2023)
31. Tesfagergish, S.G., Damaševičius, R., Kapočiūtė-Dzikienė, J.: Deep learning-based sentiment classification in amharic using multi-lingual datasets. Computer Science and Information Systems 20(4), 1459 – 1481 (2023)
32. Tesfagergish, S.G., Kapočiūtė-Dzikienė, J., Damaševičius, R.: Zero-shot emotion detection for semi-supervised sentiment analysis using sentence transformers and ensemble learning. Applied Sciences (Switzerland) 12(17) (2022)
33. Yang, Z., Li, Z., Zhu, D., Zhou, Y.: Tri-clt: Learning tri-modal representations with contrastive learning and transformer for multimodal sentiment recognition. Information Technology and Control 53(1), 206 – 219 (2024)
34. Zhang, L., Wang, S., Liu, B.: Deep learning for sentiment analysis: A survey. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 8 (2018)

Gaganpreet Kaur received her M.sc degree from Guru Nanak Dev University India,M.Tech degree from Guru Nanak Dev University India, and doing Ph.D. degree from Chitkara University, India. She has 12+ years of experience in academia. She is currently working as JRF in Chitkara University, India. She has published more than 10 research papers in reputed journals and conferences. She has 2 patents granted and published. Her primary research areas encompass machine learning, Deep Learning, Data Mining, NLP and artificial intelligence.

Amandeep Kaur currently holds the position of a Professor at the Chitkara University Institute of Engineering and Technology, Chitkara University Punjab. She earned her

doctorate degree from I. K. Gujral Punjab Technical University, Jalandhar. Dr. Kaur's academic achievements include receiving both her M.Tech (Computer Science and Engineering) and B.Tech (Computer Science and Engineering) degrees with distinction. Additionally, she has successfully qualified for the UGC-NET in Computer Science.Dr. Amandeep Kaur boasts an extensive research portfolio, with approximately 100 publications in renowned international journals and fully refereed international conferences. She has accumulated 24 years of valuable experience in her field and has filed and published more than 107 patents.Moreover, Dr. Kaur has played a significant role in mentoring the academic growth of over 30 Ph.D. and PG students. Her primary research areas encompass medical informatics, machine learning, IoT (Internet of Things), artificial intelligence, and cloud computing.Notably, Dr. Amandeep Kaur has been recognized for her exceptional contributions, winning the Excellence Award in the "Filing Patent" category for three consecutive years (2021, 2022, and 2023) and the Best Ph.D. Supervision Award in 2023.Furthermore, she has achieved recognition on a global scale, as she is included in Stanford University's prestigious list of the top 2

Meenu Khurana received her B.E. (Honors) degree from Punjab Engineering College (PEC University) India, M.E. in Computer Science from Panjab University, Chandigarh India and Ph.D. degree from Chitkara University, India. She has 28+ years of experience in industry and academia. She is currently working as Professor Pro Vice Chancellor in Chitkara University, India. She is a Senior Member of IEEE, Professional Member of ACM, Life Member of ISTE.She has published more than 80 research papers in reputed journals and conferences. She has 12 patents granted and 3 published. She has won best paper awards in various conferences few of them are IMEIS 2022, ICAC3N 2021, NCASM 2020 and more.She has been keynote speaker at various conferences, the recent ones are ICAETC, 2023, India; ICADEIS 2023, Indonesia. She has been Technical Program Committee member and session chair at conferences of repute, few of them are SmarTechCon 2023, IEEE DELCON 2022, ICAN 2022. Dr. Meenu Khurana served as judge in Smart India Hackathon in 2022. She is the guest editor for special issue of SN Computer Science. Her research area includes vehicular adhoc networks, fog computing, MIMO systems, MU-MIMO, data deduplication, machine learning algorithms. She has delivered expert sessions and conducted workshops in the area of vehicular adhoc networks, MIMO systems, MU-MIMO, 5G Networks in VANETs.

Robertas Damaševičius received the Ph.D. degree in informatics engineering from the Kaunas University of Technology, Lithuania, in 2005. He is currently a Professor with the Department of Software Engineering, Kaunas University of Technology and the Department of Applied Informatics, Vytautas Magnus University, Lithuania, and an Adjunct Professor with the Faculty of Applied Mathematics, Silesian University of Technology, Poland. He also lectures software maintenance, human–computer interface, and robot programming courses. He is the author of more than 500 articles and a monograph published by Springer. His research interests include sustainable software engineering, human–computer interfaces, assisted living, and explainability. He is also the Editor-in-Chief of the Information Technology and Control journal.

Received: March 28, 2024; Accepted: July 29, 2024.

Computer Science and Information Systems 21(4):1435–1455 https://doi.org/10.2298/CSIS230725042W

Advancing Crack Segmentation Detection:Introducing AAMC-Net Algorithm for Image Crack Analysis

Wang Xiaofang[1], Liu Chenfang[2], Hou Junliang[1], and Zhou Liang[1]

[1] Geely University of China,
Chengdu Sichuan, 641423
939549393@qq.com
houjunliang@guc.edu.cn
zhouliang@guc.edu.cn
[2] Chengdu College of University of Electronic Science and Technology of China,
Chengdu Sichuan, 611731
2975431869@qq.com

Abstract. This research paper presents an innovative solution to address the challenges of poor detail detection effectiveness and prolonged training time in image segmentation. The proposed approach leverages the Adaptive Attention Multiscale Convolution Network (AAMC-Net), incorporating a multi-scale dilated convolution VGG_L network for feature extraction and a deconvolution method for image segmentation. Extensive experiments demonstrate the superior performance of the proposed algorithm concerning intersection over Union (IOU), accuracy, precision, recall, F1, average training efficiency, and segmentation efficiency when compared to several traditional algorithms. On average, the proposed algorithm achieves remarkable improvements of 3.9%, 3.1%, 1.7%, 4.9%, 17.9%, 14.8% ,and 20.2% in these metrics. Moreover, the enhanced algorithm exhibits notable advantages in detail processing and real-time image segmentation detection.

Keywords: Image crack segmentation, Convolutional neural network,VGG_L, Attention mechanism.

1. Introduction

As a major infrastructure construction center, China has built a road network spanning 6 million kilometers by 2022 [2]. Effective road damage detection is crucial for ensuring road safety. Road cracks are early indicators of pavement deterioration [9]. Manually extracting cracks based solely on workers' experience and subjective judgments is inefficient and costly. Rapid and precise road crack detection is vital for reducing road maintenance expenses, enhancing driving safety, lowering fuel consumption, and prolonging road lifespan [27]. As a key technique for crack detection, effective crack segmentation can advance crack detection technology to some extent.

With the fast evolution of computer processors, image processing and computer vision have been extensively applied in crack segmentation, including threshold-based road crack extraction, morphological detection, region segmentation, and edge detection methods [12] [16]. Thresholding techniques like Otsu's method and histogram thresholding, despite being simple and fast, are prone to environment and illumination effects, resulting in poor performance on images with blurry crack edges. Morphological methods such

as erosion and dilation, multistructural and multiscale mathematical morphology, though effectively suppressing noise interference and achieving good detection with fast computation, struggle with detailed segmentation. Region-based crack segmentation methods like watershed and region growing enable automatic segmentation but perform poorly on faint and tiny cracks, requiring enhancement algorithms to suppress noise. Common edge detectors like Sobel, Roberts, gradient operators, Laplacian operators, Marr and Canny, though easy to implement and fast, are sensitive to noise due to template size and direction limitations.

With the development of artificial intelligence, deep learning and convolutional neural network (CNN) technologies have become a research trend for applying image crack segmentation. Models including FCN [34] [10], UNet [35], and UNet++ [33]have improved crack segmentation accuracy. However, these models have drawbacks like large computational loads and many network parameter settings, making them less ideal for real-time detection.

The shortcomings in fine-grain crack segmentation and the absence of real-time detection capabilities are pressing issues that directly impede advancements in intelligent crack detection systems. The VGG neural network is notable for its minimal computational demands and swift processing speeds, making it a comparatively lightweight model apt for rapid image analysis [23]. Attention mechanisms facilitate end-to-end learning, thereby aiding in the interpretation and understanding of data. When amalgamated with neural network models, these mechanisms markedly improve the precision of segmentation networks, demonstrating exceptional efficacy in image detection tasks [18]. Moreover, the Convolutional Block Attention Module (CBAM) enhances feature extraction in convolutional layers, accentuating key image information and excelling in local texture extraction. Dilated convolutions maintain spatial resolution whilst effectively capturing a broader contextual scope, all without augmenting the number of parameters or computational complexity. Consequently, this study aims to refine the VGG neural network by leveraging the benefits of CBAM and dilated convolutions to develop the AACM-Net model. The objective is to optimize the segmentation of crack details whilst concurrently reducing the time required for crack detection, thereby elevating the overall accuracy of segmentation.

2. Related Work

As the country places greater emphasis on intelligent transportation systems, the focus has shifted towards intelligent detection of road cracks. Image processing, particularly digital image processing, has witnessed significant advancements in crack detection [17]. Notably, researchers like Xiu et al. [31]proposed using the Sobel operator and CV model for image edge detection to address uneven gray image segmentation. Similarly, Jiang et al. [11] employed adaptive thresholding to extract crack edges and effectively mitigate noise interference. However, challenges persist due to complex lighting conditions, background variations, and other external factors, leading to suboptimal results.

With the rapid advancement of processors, artificial intelligence and computer vision have increasingly come to the fore in the digital economy era. Deep learning and convolutional neural networks point the way forward for road crack detection. Notably, like J.Long et al [15]utilized a semantic segmentation technique using a Fully Convolutional

Network(FCN8s). The FCN8s approach is quite adaptable, capable of handling images of any dimension. It streamlines the training regimen by producing classification labels in just one forward pass, enabling seamless training and prediction from start to finish. However, it's worth noting that this method demands a lot of computational power and falls short when it comes to detailing the textures of smaller objects. To address this, Dung et al. [4] proposed an FCN-based automatic crack identification algorithm, employing convolutional layers instead of fully connected layers to classify images of any scale. However, FCN's inefficiency and limited receptive field hinder it from capturing global and detailed information effectively. Further, Sun et al. [24] used a hybrid approach combining multi-resolution features with Transformers for crack semantic segmentation(Multi-Transformer). The method use a global receptive field for scene understanding and incorporates Transformer networks for semantic segmentation, overcoming the limitations of CNNs' small receptive fields. However, the use of Transformers and multi-resolution features increases the model's computational complexity.

At the same time, Wang et al [29] proposed an efficient road surface crack segmentation algorithm using a deep learning encoder-decoder structure. They used a pre-trained DenseNet121 as the encoder to extract road features and a global attention upsampling module as the decoder for crack segmentation. Although the method effectively improves training efficiency, it is highly dependent on the quality and quantity of the training data. The use of multiple attention modules and a pre-trained DenseNet121 adds to the model's complexity. Lin et al [13] used a deep semantic segmentation method combining Conditional Random Fields (CRF) and CNN (CRF-CNN) to capture contextual area information and background information for crack segmentation. The inclusion of contextual information enhances the model's image understanding capabilities but also increases its complexity and computational costs. Building on this, they introduced multi-scale inputs and sliding in pyramid pooling layers the following year to capture context for patch-background [14], improving semantic segmentation accuracy. While the introduction of efficient training methods somewhat optimizes the CRF and CNN architecture, the model remains complex and resource-intensive.

Furthermore, Fan [6] and his team introduced a road crack detection algorithm based on deep learning and adaptive image segmentation. The algorithm trains a deep convolutional network to determine whether an image contains cracks and uses adaptive thresholds to extract road surface cracks. Although the method performs well in extracting various types of road cracks, it requires substantial computational resources and has a long training time.

To overcome the limitations of existing methods and enhance crack detection accuracy, we propose a novel algorithm based on AAMC-Net image. The algorithm leverages the CBAM hybrid domain attention mechanism to strengthen global and local image features. Furthermore, we employ multi-scale dilated convolutions VGG_L for feature extraction and utilize deconvolution for fracture detection. The entire model training is based on MSELoss Adam for adaptive learning, optimizing performance and efficiency. Through extensive experiments, we demonstrate the superior capabilities of our AAMC-Net based algorithm in intelligent road crack detection, providing an innovative solution for infrastructure maintenance and safety enhancement.

3. Research Method

The CBAM attention mechanism is a lightweight, general-purpose module that can be seamlessly integrated into convolutional neural network architectures for end-to-end training and the capture of useful information. VGG achieves learning by deepening the network with multiple nonlinear layers, albeit with fewer network parameters. In this study, we use the VGG network as the foundational structure and propose a multi-scale dilated convolution variant, termed VGG_L. This is combined with the CBAM attention mechanism to construct the AACM-Net network model specifically designed for road crack segmentation. The architecture of this model is depicted in Figure 1.

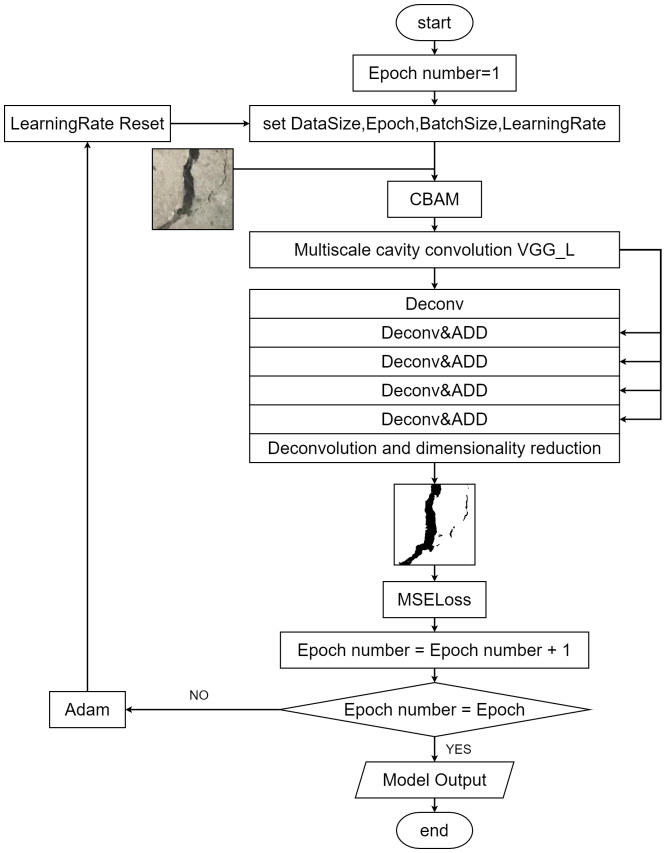


Fig. 1. Algorithm flow chart

A deep learning-based road crack image segmentation model is designed, with its training process illustrated in Figure 1. The initial step involves setting model parameters, including the size of the dataset, the number of training iterations, batch size, and the initialization of the learning rate. The model's input, crack images, first pass through the CBAM module, which enhances key features of the image using channel and spatial

attention mechanisms. Subsequently, feature extraction is performed using the VGG_L model, which employs multi-scale dilated convolutions to capture image details across different scales. Following feature extraction, the model reconstructs and segments features through a series of deconvolution and feature fusion steps (Deconv&ADD). At the end of each training cycle (Epoch), the model evaluates the segmentation effectiveness by calculating the Mean Squared Error (MSELoss) and adjusts parameters based on the loss using the Adam optimizer. This iterative process continues until the predetermined number of training iterations is reached, at which point the model outputs the final crack segmentation results, completing the training process. This training approach ensures that the model effectively learns to segment cracks from complex backgrounds, enhancing the accuracy and robustness of segmentation through the attention mechanism and multi-scale feature fusion. In this way, the model is equipped to handle crack detection challenges across various road surfaces and lighting conditions.

3.1. Image Processing

CBAM based on hybrid attention mechanism is used to preprocess the input crack feature image to enhance the global and local texture feature information. The hybrid attention mechanism based CBAM [30]model implements the image feature enhancement process by processing global and local texture features in channel features and spatial features. It consists of convolution of channel and spatial attention mechanisms, the overall process is shown in Figure 2.

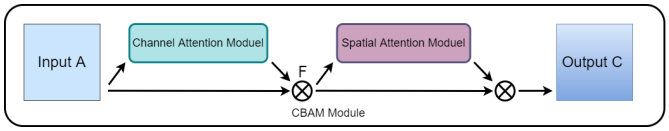


Fig. 2. Overall CBAM flow chart

The input A is initially processed through a channel attention module, resulting in a feature map that is element-wise multiplied (denoted as $\otimes$) with the input A to produce a feature map F. This process enhances features associated with specific channels. Subsequently, the feature map F undergoes further processing through a spatial attention module, and the resulting feature map is again element-wise multiplied with F, leading to the final output C. This illustration demonstrates how channel and spatial attention modules are sequentially employed to enhance the network's capacity to represent input images effectively.

This study integrates a Combined Channel and Spatial Attention Mechanism (CBAM) to enhance the precision of crack segmentation in images. As shown in Figure 2, the input image A initially undergoes processing through a channel attention module. This module focuses on the importance of different channels within the image, generating a feature map. The primary goal of this step is to highlight feature channels that are most relevant for crack detection.After processing by the channel attention mechanism, the feature map is element-wise multiplied with the original input A, resulting in the feature map F. This

approach applies channel weights to the original features, intensifying the focus on significant features. Subsequently, the feature map F is directed to a spatial attention module, concentrating on the crack regions within the image. The feature map, post spatial attention processing, is element-wise multiplied with the input F, further enhancing spatial focus. Ultimately, the input A, after undergoing channel and spatial attention weighting, transforms into the output C. This output contains a richer and more detailed representation of crack features, crucial for the following steps of crack segmentation. The model is capable of more precise crack detection and segmentation, particularly in complex crack images. Notably, throughout this process, the size of the output features aligns with that of the input features, maintaining dimensions of 214×214. The computational method for the CBAM cross-domain attention mechanism is presented in formula 1.

$$C = CBAM(A) = SAM(CAM(A) \otimes A) \otimes (CAM(A) \otimes A) \tag{1}$$

Among them, both C and $CBAM(A)$ are the results of CBAM mixed domain attention mechanism operation, A is the original crack image, CAM is the channel domain attention operation mechanism for crack image processing, SAM is the spatial domain attention mechanism operation for crack image processing, $\otimes$ is the matrix multiplication operation, and the matrix multiplication operation formula is shown in formula 2.

$$F_{ik \times jl} = A_{i \times j} \bullet A\prime_{k \times l} \tag{2}$$

Among them,the symbol $\bullet$ signifies element-wise matrix multiplication, $A_{i \times j}$ is the matrix A whose pixel size is $i \times j$, $A\prime_{k \times l}$is the matrix$A\prime$ whose pixel size is $k \times l$, and $F_{ik \times jl}$ is the result matrix F whose pixel size is $ik \times jl$.

Channel Domain Attention Mechanism The Channel Attention Mechanism (CAM) enhances the capability for global feature extraction in crack images by evaluating the importance of each channel feature map. It achieves this by compressing and reducing dimensions in the spatial domain to obtain the channel attention map. The computational process is illustrated in Figure 3.

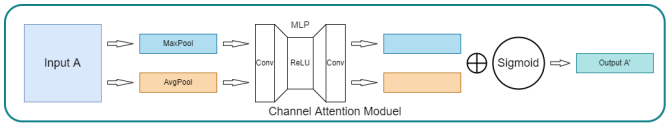


Fig. 3. CAM calculation flow chart

As can be seen from Figure 3, the original fissure image A is fed into the channel attention mechanism model, processed by the maximum pooled layer and the average pooled layer respectively to obtain two different feature graphs, which are then fed into the MLP neural network [25] for feature vector weight assignment. The whole process adopts the consistency principle, and finally obtains the global texture enhancement feature map through Sigmoid activation function [28]. The calculation is shown in formula 3.

$$A' = CAM\,(A) = Sigmoid\,(MLP\,(AvgPool\,(A)) + MLP\,(MaxPool\,(A))) \quad (3)$$

$Sigmoid$ represents Sigmoid activation function, $AvgPool$ represents average pooling operation, $MaxPool$ represents maximum pooling calculation, and MLP represents MLP neural network, which is used to extract attention map in CAM operation, essentially convolves two convolution layers with 1 convolution core with a ReLU activation layer [28], as shown in formula 4.

$$MLP\,(pool) = Conv(ReLU(Conv(pool)) \quad (4)$$

Where is $pool$ stands for pooled input, $Conv$ for convolution, and $ReLU$ for ReLU activation. The global texture feature image processed by channel attention is convolved with the original feature image to get the feature map F.

Spatial Attention Mechanism The spatial attention mechanism (SAM) enhances the ability of the model to extract local texture features by obtaining the importance of different regions in the feature map. SAM compresses the channel dimension of the feature graph to obtain the attention graph in the spatial domain. The calculation flow is shown in Figure 4. As shown in Figure 4, the maximum and average pooled layers are still used

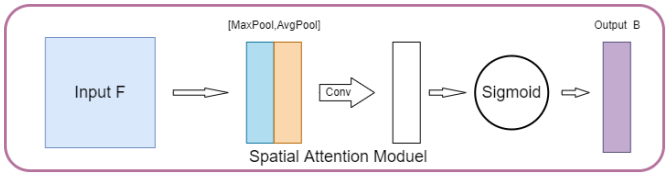


Fig. 4. SAM calculation process

in the input of Feature Diagram F into the spatial attention mechanism, resulting in two features with the same size and number of channels. Then, the two feature images are stitched together and processed by Sigmoid activation function after convolution operation of 7×7 to obtain the local texture enhanced crack image. As shown in formula 5.

$$B = SAM\,(F) = Sigmoid\left(Conv^{7\times7}\left([AvgPool\,(F) + MaxPool\,(F)]\right)\right) \quad (5)$$

Of which, $Conv^{7\times7}$ represents the convolution operation with a convolution nucleus size of 7×7.

3.2. Crack Image Feature Extraction

The study uses the VGG_16 [23] deep neural network as a base model, improves it, and proposes a multiscale null convolution VGG_L neural network for crack image feature

extraction. The network retains the first 10 convolutional layers and 4 pooling layers of VGG_16, and connects 1 layer of CBAM mixed-domain attention mechanism and 1 layer of dual-scale null convolution. The architecture lightens the VGG_16 model while enhancing the feature focusing ability of the network, the contextual information capturing ability, which can better extract the crack detail information and improve the recognition accuracy. The architecture is shown in Figure 5.

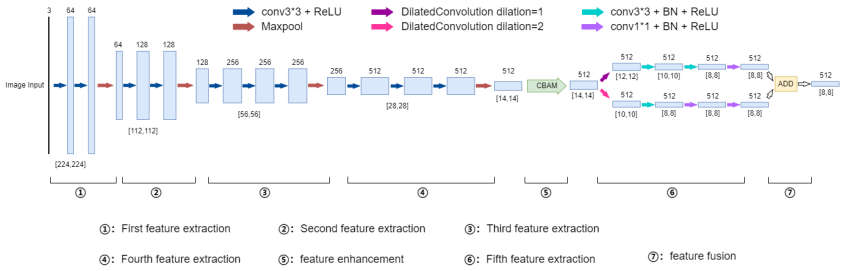


Fig. 5. VGG_L network architecture for multiscale void convolution

As shown in Figure 5, the network sequentially processes the input image through convolutional layers, max pooling layers, and dilated convolutional layers. The mid-section of the network includes convolutional layers with Batch Normalization (BN) and the ReLU activation function, along with an attention module (CBAM), for further refinement of features. After a series of convolutional and dilated convolutional layer processing, features are ultimately fused through an ADD operation, resulting in the generation of the output feature map. In the figure, the content within brackets [] indicates the size of the feature map, numbers denote the quantity of feature layers, and arrows of different colors represent various operations within the network.

The study inputs images of 214×214 dimensions into the VGG_L model, where they first undergo four feature extractions and one feature enhancement. The feature map, post-enhancement, is then subjected to a fifth feature extraction and one feature fusion. The first and second feature extraction processes are identical, involving two rounds of conv3*3+ReLU operations followed by one Maxpool operation. The third and fourth feature extractions are similar, consisting of three conv3*3+ReLU operations and one Maxpool operation each. The fifth feature extraction involves dual-scale dilated convolution, where the feature map enhanced by CBAM undergoes convolution operations at dilation rates of 1 and 2 simultaneously. After the dilation rate 1 convolution, the feature map undergoes two conv3*3+BN+ReLU operations, followed by one conv1*1+BN+ReLU operation to generate the feature map. Conversely, the feature map post dilation rate 2 convolution first undergoes one conv3*3+BN+ReLU operation, then two conv1*1+BN+ReLU operations to generate its feature map.

For feature fusion, the two feature maps obtained from the fifth feature extraction, being of equal size, are fused by adding corresponding width and height dimensions. This results in the final feature map from the feature extraction process.

The first feature extraction is the same as the second feature extraction. After two convolution operations with a 3×3 kernel and padding of 1, followed by a ReLU activation operation [28], the convolution kernel is pooled with a 3×3 size, a stride of 2, and a kernel of 2. The computation is as shown in formula 6.

$$C_i = MaxPool(ReLU\ \ (Conv\,(ReLU\,(Conv\,(C_{i-1}))))\quad i \in (1,2) \tag{6}$$

Among them, C_{i-1}represents the input fracture data set image, i represents the number of feature extraction processing times, C_i represents the feature extraction result, and when $i\ =\ 1$, C_0 is C, represents the fracture image after CBAM feature enhancement.

The third and the fourth feature extraction processes are the same, consisting of three layers of convolution core size is 3×3, padding 1 and ReLU activation, and the feature extraction results are obtained by the maximum pool size is 3×3, step is 2 and kernel is 2, as shown in formula 7.

$$C_i = MaxPool(ReLU\ (Conv(ReLU(Conv\,(ReLU\,(Conv\,(C_{i-1})))\ \ (i \in (3,4)) \tag{7}$$

After processing, the feature extraction results of crack map were obtained C_1, C_2, C_3 and C_4. The fourth feature extraction result C_4 is further enhanced by the CBAM mixed domain attention mechanism module to solve the problem of fine-grained information loss.

Dilated convolutions [20] keeps the image size constant while enhancing the receptive field of the convolution neural network. Double-scale dilated convolutions is used to achieve the fifth feature extraction to reduce the loss of context information and improve the accuracy of crack recognition. The processing involves a convolution operation with a void ratio of 1 and a void ratio of 2 and three ordinary convolution operations, as shown in Formula 8.

$$DC_\gamma(x,y,n,\gamma) = Conv(Conv(Conv(DConv\,(x,y,n,\gamma)))) \tag{8}$$

Among them, $DC_\gamma(x,y,n,\gamma)$represents the improved multi-scale convolution function, γ represents the void ratio, determines the sampling interval of the convolution kernel, x and y represent the length and width of the input crack image respectively, and n represents the edge length of the convolution core. $DConv$ represents the convolution operation of holes, the calculation is shown in formula 9.

$$DConv\,(x,y,n,\gamma) = \sum_{n}^{x}\sum_{n}^{y} I\,[x+\gamma n, y+\gamma n]\,w\,[n,n]) \tag{9}$$

Among them, $DConv$ represents the dilated convolutions operation, w is the void filter, I is the input image, and γ-1 represents the number of holes.

Dilated convolutions enhances the effect of image extraction by enlarging the receptive field of the convolution process without any additional parameters. The receptive field computation is shown in formula 10.

$$G = (m+1)(m-1)+m \tag{10}$$

Among them, m represents convolution nucleus size, G represents receptive field size.

In this paper, the dilated convolutions conv+BN+ReLU is composed of convolution core = 3×3, padding = 1, stride = 2. The image is processed by convolution with voidage 1 and voidage 2, respectively, and two features are extracted after three ordinary convolution operations. The convolution layer is composed of conv+BN+ReLU. Among them, the void ratio is 1, and the convolution parameter of the first two convolution treatments is 3×3, padding = 0, stride = 1. The convolution parameter of the third convolution is convolution kernel =1×1, padding = 0, stride = 1. When the void ratio is 1, the first convolution parameter is convolution kernel =3×3, padding = 0, stride = 1. The convolution parameters of the second and third convolution are convolution kernel =1×1, padding = 0, stride = 1. Because the sizes of the two feature maps are the same, the weighted operation is used to fuse the two feature maps. The result is the fifth feature extraction graph. , as described in formula 11.

$$MDC = \sum_{r=1}^{2} DC_{\gamma} \tag{11}$$

Among them, MDC is the fused crack feature extraction result.

3.3. Image Segmentation and Dimension Reduction

In order to reduce the channel dimension and improve the resolution of the feature map, the fused deconvolution and skip connection methods are used for five deconvolution operations [7] [8]. In deconvolution, which consists of ConvTanspose+BN+ReLU, the deconvolution layer parameter is set to convolution kernel =3×3, padding = 1, stride = 2, out_padding = 1. The output size of the deconvolution is calculated as Formula 12.

$$o = s\,(i-1) + 2p - k + 2 \tag{12}$$

Here, i represents the input size, k the convolution size, p the boundary extension, s the convolution size, and o is the output size.

The feature fusion feature map MDC is subjected to the deconvolution process. The processed feature map is skip connect with the first pooled processed feature map ,to achieve feature fusion. Then skip connect the output feature map with the second pooled feature map to realize feature fusion. According to this idea, sequential deconvolution processing and skip connect with the first four pooling results C_1,C_2,C_3,$C_4$1 to complete the image feature fusion. Because the dimension and channel dimension of the deconvolution feature map are the same as those of feature extraction. The method of wide-high correspondence addition is used to fuse the image features, and the MDC of fracture extraction is obtained. Although MDC has the same resolution as the input image, the channel dimension is much higher than the input image. Convolution reduction is used to reduce the high dimensional feature map to 3 dimensions. The number of convolution kernels is equal to the number of low-dimensional images, that is, the width and height of convolution kernels are 1. Thus obtains the final image segmentation result Y.

3.4. Merging MSELoss and Adam Parameter Optimization

SGD gradient descent learning rate optimization algorithm [19] has a slow convergence rate and local optimal solution. The model learning process was optimized using a fusion

of MSELoss and Adam methods. First, the loss value is calculated using MSELoss, and the loss function is used to calculate the mean square deviation, as shown in Formula 13.

$$MSELoss\,(x,y) = (x-y)^2 \tag{13}$$

Where x is the forecast and y is the label. During the model training process, MSELoss loss values are calculated every iteration and the results are iterated using the Adam adaptive learning rate algorithm [1] [26]. The handling steps are as follows:

1. Initialization. The initial chemical accessibility is θ ,the step length is α ,and the initialization matrix estimates the exponential rate of decay as β_1,β_2. β_1,β_2 as a value of [0,1]. Initialize the numeric constant δ, initialize the number of updates $t = 0$, and compute the MSELoss.

2. Gradient calculation. MSELoss is gradient computed to get the current loss gradient, as shown in Formula 14.

$$g = \nabla_{MSELoss} \tag{14}$$

Where g represents the MSELoss gradient, ∇ is the gradient calculation symbol.

3. Calculation of biased moment estimation. The second moment estimate is obtained by using the partial first moment estimate to calculate the partial first moment estimate of the gradient as shown in formula 15.

$$s = \beta_1 s + (1-\beta_1)\, g \tag{15}$$

Where s represents the partial first moment estimate of the gradient, and β_1 represents the exponential decay rate of the gradient partial first moment estimate.The partial first moment estimation of the gradient is introduced into the formula of partial second moment estimation to correct the influence factor. The calculation is shown in formula 16.

$$r = \beta_2 r + (1-\beta_2)\, g \odot g \tag{16}$$

Among them, r represents gradient second moment estimation, β_2 represents gradient second moment estimation exponential decay rate.

4. Deviation correction. The first moment estimation is modified to obtain the deviation coefficient. The calculation is shown in formula 17.

$$\hat{r} = \frac{r}{1-\beta_2^t} \tag{17}$$

The $\hat{s}$ represents the correction of the partial first moment deviation of the gradient, and t represents the number of iterations of the learning rate.The second-order deflection coefficients are obtained by revising the second-order moment estimation. The calculation is shown in formula 18.

$$\hat{r} = \frac{r}{1-\beta_2^t} \tag{18}$$

Among them, $\hat{r}$ represents the correction of deviation of gradient partial second moment.

5. Update learning rates. The first-order moment estimation correction coefficient and second-order moment estimation correction coefficient are incorporated into the learning

rate updating formula to calculate and update the learning rate.The calculation is shown in formula 19.

$$\theta_i = \theta_{i-1} - \alpha \frac{\hat{s}}{\sqrt{\hat{r}} + \delta} \tag{19}$$

Where θ_i is the updated learning rate, θ_{i-1} is the previous learning rate. After processing, the corresponding learning rate is obtained, which is used for the next training, and the whole process is repeated until the end of the training.

4. Experimental Results and Discussion

Research using AMD EPYC 7302 16-Core Processor CPU, NVIDIA GeForce RTX 3090GPU computing power resources. The development uses the Pytorch deep learning framework and the Numpy framework.

4.1. Data Source

The study used self-collected data, CFD datasets [21] [3],CRACK500 [32] and GAPs384 [5] datasets, a total of 5800, including different lighting, different scenes. In order to reduce the difference caused by the external environment, the scale normalization is carried out. Labelme is used to annotate the data, and the annotation information includes background and crack. Because of the deficiency of the public dataset and the self-mining dataset, the existing dataset is expanded by image augmentation. By means of rotation, mirroring, salt and pepper noise, Gaussian noise, random noise, etc. [22], the original data set is enlarged to 12800 images, and all image sizes are set to 224×224pixel.

Salt and pepper noise and random noise were added to the data augmentation, respectively 1.5% and 1%. Random Gaussian noise is used for noise amplification, as shown in Formula 20.

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \tag{20}$$

Among them, μ represents mean, σ represents standard deviation, $\mu = 0, \sigma = 0.05$. After processing, the experimental dataset is obtained. The dataset sample is shown in Figure 6.

Figure 6shows that A is the original figure, B is the random noise image, C is the salt and pepper noise image, and D is the Gaussian noise image.

4.2. Analysis of Results

Loss rate analysis of different algorithms The training parameters of UNet networks [35], UNet++ networks [33], FCN8s networks [15], CRF-CNN [13], Multi-Transformer [24] are shown in Table 1.

Based on the parameters set in Table 1, the model training of UNet networks [35], UNet++ networks [33], FCN8s networks [15], CRF-CNN [13],Multi-Transformer [24],

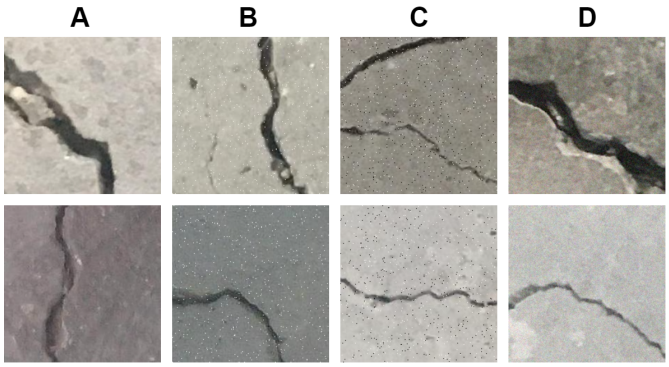


Fig. 6. Partial data set

Table 1. Training Parameters Configuration Table

Parameter Name	Value
InputSize	[3,224,224]
OutputSize	[3,224,224]
DataSize	12800
BatchSize	2
Epoch	16
LearningRate	10^{-6}
Max iteration	102400

and the training loss process of several algorithms is analyzed. The loss curve is shown in Figure 7.

As Figure 7 shows,the image illustrates the diminishing trend of iteration losses during the training process for six models: AAMCNet, UNet, Multi-Transformer, U-Net++, FCN8s, and CRF-CNN. It uses the horizontal axis to denote the number of iterations and the vertical axis to represent the loss rate. This effectively showcases each model's efficiency in reducing losses over successive training iterations. In the training process of the six models, with the increase of the number of iterations, the loss value decreases rapidly at first, and then decreases gradually, and finally approaches convergence. The initial loss of AAMC-Net model is the lowest, only 0.29. This is due to the use of CBAM mixed domain attention mechanism twice, which can more effectively focus the important features of crack image and enhance the network feature extraction ability. In addition, the AAMC-Net model converges nearly 10000 iterations, and AAMC-Net converges fastest. This is due to the use of lightweight VGG_L networks and the use of Adma's MSELoss adaptive function in model training, which adaptively adjusts the learning rate and helps the model find the minimum value of the loss function faster.

Subjective Evaluation Analysis In the same experimental environment, the image was segmented using improved Sobel [31], adaptive threshold algorithm [11], UNet [35], UNet++ [33], FCN8s [15], CRF-CNN [13],Multi-Transformer [24] and AAMC-Net network respectively. The segmentation effect was shown in Figure 8 and Figure 9.

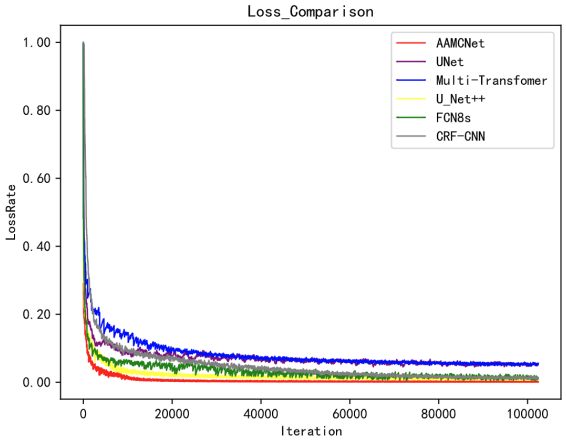
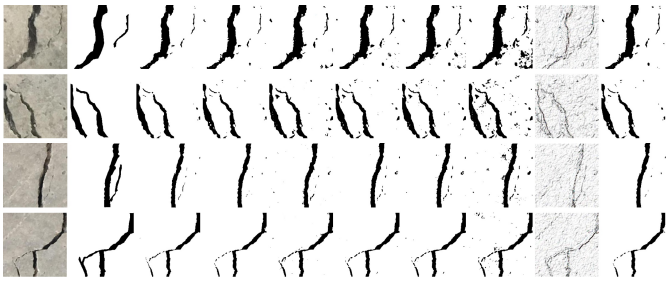


Fig. 7. Loss ratio chart

An analysis of algorithm performance was conducted using subjective human evaluation. As shown in Figures 8 and Figures 9, two digital image processing methods outperform the CFD dataset in image segmentation. The CFD dataset contains numerous road surface textures. However, the improved Sobel operator and adaptive thresholding are highly sensitive, effortlessly capturing subtle variations within the images. In road crack detection, identifying road surface textures as granular cracks in the CFD dataset adversely affects the effectiveness of image segmentation.

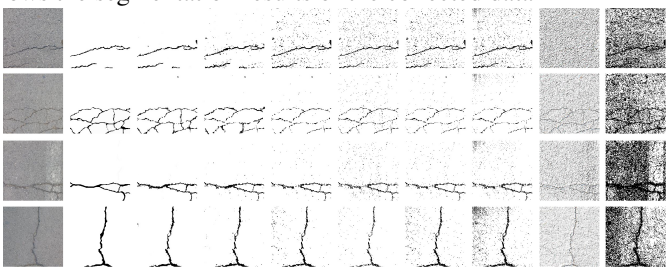
Within the neural network models, particularly in the segmentation results of UNet, stemming from the CFD dataset, significant noise is present, which to some extent hampers the segmentation performance. Even though the upgraded version, UNet++, demonstrates clear crack segmentation results, it struggles in handling image details. This is attributed to UNet++ filtering redundant information during model training, inadvertently filtering out crack details. Consequently, UNet++ performs poorly in applications involving small crack details. While FCN8s progressively integrates Pool3 and Pool4 layers, thereby effectively retaining some crack details, it still retains a considerable amount of noise. Although CRF-CNN and Multi-Transformer excel in segmenting image details, residual noise still impacts the segmentation performance.

In contrast, AAMC-Net effectively mitigates noise interference and simultaneously captures intricate crack segmentation details. This achievement is owed to the integration of CBAM during preprocessing and feature extraction stages, enabling the capture and enhancement of both global and local information. Moreover, the fusion of CBAM and multi-scale atrous convolution modules for feature extraction facilitates multi-layer perception of crack features at varying scales, effectively mitigating information loss resulting from pooling. Additionally, this approach mitigates interference from non-crack information during crack image segmentation processing.



1.Original drawing 2.Label 3. Improved algorithm 4.UNet++ 5.Muti-Transfomer 6.CRF-CNN 7.FCN8s 8.UNet 9.Improved Sobel operator 10.Adaptive threshold

Fig. 8. shows the segmentation results of the collected data



1.Original drawing 2.Label 3. Improved algorithm 4.UNet++ 5.Muti-Transfomer 6.CRF-CNN 7.FCN8s 8.UNet 9.Improved Sobel operator 10.Adaptive threshold

Fig. 9. shows the segmentation results of the CFD data

4.3. Objective Performance Analysis

The study used improved Sobel [31], adaptive threshold algorithm [11], UNet [35], UNet++ [33], FCN8s [15], CRF-CNN [13],Multi-Transformer [24] and AAMC-Net network to segment 10 road crack images randomly selected in the validation concentration. After processing, the processing images shall be evaluated by using cross and merge ratio, accuracy ratio (Acc), precision ratio, recall ratio and F1 value [36], the processed average results will be retained to 3 decimal places, as shown in Table 2.

From the table 3, we can see that among the eight comparison algorithm models, the combination ratio, accuracy ratio, precision ratio, recall ratio and F1 value of the improved algorithm are the best, 96.310%, 96.379%, 99.550% and 97.008% respectively. Multitransformer takes second place, and the two digital image processing algorithms are the worst. The worst of the six models is the Unet network. Compared with the five network models, the improved algorithm has an average increase of 2.650%, 1.860%, 0.626%, 3.092% and 1.731% respectively.

The AAMC-Net model exhibits superior performance in image crack segmentation, attributed to the utilization of dual CBAM processes that enhance the crack feature maps through image strengthening, thereby augmenting the network's feature capturing capabilities. Additionally, the model benefits from dual-scale atrous convolutions for feature extraction, where atrous convolutions expand the image's receptive field. The dual-scale atrous convolutions effectively capture crack feature information at different scales, en-

Table 2. Training Parameters Configuration Table

Algorithms	IOU	Acc	Precision	Recall	F1
AAMC-Net	96.310	96.321	96.379	99.550	97.008
UNet	90.118	91.205	96.134	92.668	94.112
FCN8s	94.014	94.623	95.501	95.486	94.418
UNet++	94.808	95.812	95.806	97.912	95.542
CRF-CNN	95.163	95.513	95.328	98.116	96.231
Multi-Transformer	95.217	95.826	96.134	98.915	96.532
Adaptive Threshold	93.450	94.685	95.215	98.695	83.115
Improved Sobel	86.650	86.693	86.690	97.195	70.126

compassing local details and global structures, while maintaining robustness against various interferences. This enhancement amplifies the model's expressive capacity. Through analysis of five objective evaluation metrics, the algorithm's proficiency in crack detail handling and segmentation performance is once again affirmed.

4.4. Reasoning Speed

In order to verify the real-time detection performance of the algorithm, the real-time performance of road crack processing is evaluated by using a single image crack segmentation time, the calculation is shown in formula 21.

$$T = \frac{1}{h}\sum_{i=1}^{h} Q_h \tag{21}$$

Among them, T represents the average time of single image segmentation, h represents the number of images processed in batches, and Q_h represents the time used to segment the nth image.

The study used UNet [35], UNet++ [33], FCN8s [15], CRF-CNN [13], Multi-Transformer [24] and AAMC-Net network to train in the same experimental environment. The training time of six network models is analyzed and the average segmentation time of single crack image after training is compared. The result is 3 decimal places, as shown in Table 3.

Table 3. Training Parameters Configuration Table

Model	AAMC-Net	UNet	FCN8s	UNet++	CRF-CNN	Multiscale CRF
Average split time(s)	0.240	0.242	0.256	0.276	0.332	0.344
Training time(min)	241	245	278	306	332	345

As can be seen from Table 4, among the six algorithms, AAMC-Net has the shortest model training time and average detection time per image, followed by UNet, while Multi-Transformer takes the most time. When analyzed in conjunction with Table 3, although the training and average detection time of the UNet model are not significantly

different from those of AAMC-Net, the average accuracy rate of crack segmentation increased by 5.609%. Compared to the most effective Multi-Transformer, the average accuracy of crack segmentation increased by 0.517%, but the time for crack detection and model training decreased by 30.233% and 30.145% respectively. In real-time scenarios like road monitoring or instant image analysis, the response speed and accuracy of algorithms are paramount. The efficiency advantage of the improved algorithm is confirmed through actual model training and random crack image segmentation detection. The time advantage of the AAMC-Net architecture stems from reducing convolution and pooling layers in the VGG_16 model and integrating CBAM to optimize the output of traditional convolutions.

By decreasing the number of convolution and pooling layers in the VGG_16 model, AAMC-Net significantly reduces computational complexity, leading to a marked decrease in model training time. This is especially valuable in real-time scenarios, enabling quicker retraining of the model to adapt to environmental changes like weather or lighting conditions.

The integration of CBAM in the algorithm optimizes feature extraction, focusing more on key image areas and enhancing crack detection capabilities. The inclusion of this attention mechanism heightens sensitivity to crack features, maintaining high accuracy even in complex backgrounds or various crack types.

AAMC-Net, using multi-scale dilated convolutions, captures a broader context without adding extra parameters. This means comprehensive image content understanding without increasing computational burden, crucial for analyzing images at different resolutions and aiding real-time systems to adapt to diverse input conditions.

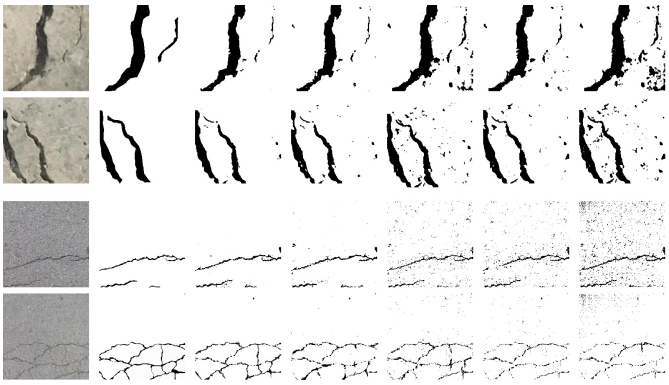
The algorithm also simplifies the training process and speeds up model convergence by automatically adjusting the learning rate with an adaptive MSELoss learning function. In real-time applications, this translates to rapid transfer training across different task scenarios, quicker convergence, and reduced model training time.

In experiments, AAMC-Net, compared with other advanced algorithms, displayed shorter average detection times and higher accuracy in crack segmentation. These results demonstrate its potential in real-time applications, as it quickly identifies and locates cracks while maintaining high accuracy. This is critical for applications requiring immediate response, such as monitoring road conditions in intelligent traffic systems or rapidly identifying and assessing road damage in disaster management systems for swift deployment of repair work and emergency measures.

4.5. Ablation Experiment

In order to study the effectiveness of the algorithm, the image segmentation of VGG_16, VGG_10, VGG_L, VGG_10 + CBAM and the improved algorithm model were studied under the same experimental environment. The segmentation results are shown in Figure 10, and the ablation analysis results are shown in Table 4.

The results of ablation experiments in Table 4 show that the average segmentation time of IOU, ACC, Precision, Recall and fracture images of AAMC-Net is the best. Compared with the ablation experiment, the VGG_10 fused with CBAM is better than VGG_10. The results show that CBAM can enhance the ability of image feature extraction and improve the performance of image segmentation. VGG_10 image fused with CBAM attention mechanism module has better performance of crack segmentation and



1.Original image 2.Label Image 3.AAMC-Net Algorithm 4.VGG_L 5.VGG_16 6.CBAM+VGG_10 7.VGG_10

Fig. 10. Ablation experimental segmentation

Table 4. Training Parameters Configuration Table

Algorithms	IOU	Acc	Precision	Recall	F1	Average split time(s)	Training time(min)
VGG_16	94.221	94.332	94.618	96.126	95.012	0.271	298
VGG_10	89.112	88.604	88.482	88.909	87.826	0.226	221
VGG_L	94.815	95.321	95.212	98.132	96.810	0.235	236
VGG_10+CBAM	94.151	94.214	94.371	95.825	93.712	0.229	230
AAMC-Net	96.25	96.351	96.382	99.532	97.008	0.240	241

less interference in image segmentation. The training time of the model and the average segmentation time of single image increase little. The results showed that multi-scale cavity convolution could enhance the receptive field in convolution process. The aim is to enhance the ability of different size feature extraction and improve the performance of image segmentation. Compared AAMC-Net with VGG_16, the performance of AAMC-Net is better than VGG_16. The model training time and detection time are better than VGG_16. Ablation experiments show that AAMC-Net is robust and optimized based on VGG_16.

5. Conclusion

In this paper, we have introduced the AAMC-Net convolutional neural network algorithm, which leverages the mixed domain attention mechanism to enhance feature extraction from the feature map. Through the integration of MSE loss and Adam during global learning and iterative training, the network model achieves improved performance for road crack segmentation and demonstrates its applicability in various image segmentation domains.

Our algorithm exhibits notable strengths in detail processing, model training, and real-time detection on single images. It successfully addresses the challenge of road crack segmentation with promising results. Furthermore, the flexibility of its application to other image segmentation tasks highlights its potential in broader contexts.

However, it is essential to acknowledge that the algorithm demands substantial computing resources, necessitating further optimization efforts. Additionally, the algorithm can only directly process images that conform to the input dimensions, which may not meet the general requirements of real-time applications. It's worth noting that there is still room for improvement in terms of multi-object real-time segmentation, in order to enhance segmentation efficiency. In the future, we plan to concentrate on optimizing the algorithm to enhance its computational performance, explore a universal detection architecture for multi-scale images, all while maintaining its effectiveness.

In conclusion, the AAMC-Net algorithm showcases significant advancements in image crack segmentation detection. As we continue to refine and optimize this approach, we anticipate its continued relevance and applicability to various image segmentation tasks, contributing to the advancement of the field.

References

1. Adam, K.D.B.J.: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 1412 (2014)
2. Cajas, Y.R.A., Guisado, Y.Z., Vergaray, A.D.: Identify faults in road structure zones with deep learning. Journal of System and Management Sciences 12(6), 163–191 (2022)
3. Cui, L., Qi, Z., Chen, Z., Meng, F., Shi, Y.: Pavement distress detection using random decision forests. In: International Conference on Data Science. pp. 95–102. Springer (2015)
4. Dung, C.V., et al.: Autonomous concrete crack detection using deep fully convolutional neural network. Automation in Construction 99, 52–58 (2019)
5. Eisenbach, M., Stricker, R., Seichter, D., Amende, K., Debes, K., Sesselmann, M., Ebersbach, D., Stoeckert, U., Gross, H.M.: How to get pavement distress detection ready for deep learning? a systematic approach (2017)
6. Fan, R., Bocus, M.J., Zhu, Y., Jiao, J., Wang, L., Ma, F., Cheng, S., Liu, M.: Road crack detection using deep convolutional neural network and adaptive thresholding. In: 2019 IEEE Intelligent Vehicles Symposium (IV). pp. 474–479. IEEE (2019)
7. Gao, H., Yuan, H., Wang, Z., Ji, S.: Pixel transposed convolutional networks. IEEE transactions on pattern analysis and machine intelligence 42(5), 1218–1227 (2019)
8. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
9. Hu, G.X., Hu, B.L., Yang, Z., Huang, L., Li, P.: Pavement crack detection method based on deep learning models. Wireless Communications and Mobile Computing 2021, 1–13 (2021)
10. Islam, M.M., Kim, J.M.: Vision-based autonomous crack detection of concrete structures using a fully convolutional encoder–decoder network. Sensors 19(19), 4251 (2019)
11. Jiang, X., Yang, X., Ding, X.: An efficient and reliable approach based on adaptive threshold for road defect detection. International Journal of Innovative Computing and Applications 12(5-6), 321–329 (2021)
12. Kheradmandi, N., Mehranfar, V.: A critical review and comparative study on image segmentation-based techniques for pavement crack detection. Construction and Building Materials 321, 126162 (2022)
13. Lin, G., Shen, C., Van Den Hengel, A., Reid, I.: Efficient piecewise training of deep structured models for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3194–3203 (2016)
14. Lin, G., Shen, C., Van Den Hengel, A., Reid, I.: Exploring context with deep structured models for semantic segmentation. IEEE transactions on pattern analysis and machine intelligence 40(6), 1352–1366 (2017)

15. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3431–3440 (2015)
16. Mohan, A., Poobal, S.: Crack detection using image processing: A critical review and analysis. alexandria engineering journal 57(2), 787–798 (2018)
17. Munawar, H.S., Hammad, A.W., Haddad, A., Soares, C.A.P., Waller, S.T.: Image-based crack detection methods: A review. Infrastructures 6(8), 115 (2021)
18. Niu, Z., Zhong, G., Yu, H.: A review on the attention mechanism of deep learning. Neurocomputing 452, 48–62 (2021)
19. Ruder, S.: An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747 (2016)
20. Schmidt, C., Athar, A., Mahadevan, S., Leibe, B.: D2conv3d: Dynamic dilated convolutions for object segmentation in videos. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 1200–1209 (2022)
21. Shi, Y., Cui, L., Qi, Z., Meng, F., Chen, Z.: Automatic road crack detection using random structured forests. IEEE Transactions on Intelligent Transportation Systems 17(12), 3434–3445 (2016)
22. Shorten, C., Khoshgoftaar, T.M.: A survey on image data augmentation for deep learning. Journal of big data 6(1), 1–48 (2019)
23. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
24. Sun, Z., Zhou, W., Ding, C., Xia, M.: Multi-resolution transformer network for building and road segmentation of remote sensing image. ISPRS International Journal of Geo-Information 11(3), 165 (2022)
25. Tolstikhin, I.O., Houlsby, N., Kolesnikov, A., Beyer, L., Zhai, X., Unterthiner, T., Yung, J., Steiner, A., Keysers, D., Uszkoreit, J.: Mlp-mixer: An all-mlp architecture for vision. Advances in neural information processing systems 34, 24261–24272 (2021)
26. Tong, Q., Liang, G., Bi, J.: Calibrating the adaptive learning rate to improve convergence of adam. Neurocomputing 481, 333–356 (2022)
27. Tran, V.P., Tran, T.S., Lee, H.J., Kim, K.D., Baek, J., Nguyen, T.T.: One stage detector (retinanet)-based crack detection for asphalt pavements considering pavement distresses and surface objects. Journal of Civil Structural Health Monitoring 11, 205–222 (2021)
28. Villmann, T., Ravichandran, J., Villmann, A., Nebel, D., Kaden, M.: Activation functions for generalized learning vector quantization-a performance comparison. arXiv preprint arXiv:1901.05995 (2019)
29. Wang, W., Su, C.: Convolutional neural network-based pavement crack segmentation using pyramid attention network. IEEE Access 8, 206548–206558 (2020)
30. Woo, S., Park, J., Lee, J.Y., Kweon, I.S.: Cbam: Convolutional block attention module. In: Proceedings of the European conference on computer vision (ECCV). pp. 3–19 (2018)
31. Xiu, C., Yin, H., Liu, Y.: Image segmentation of cv model combined with sobel operator. In: The 32nd China Control and Decision Making Conference. pp. 124–128 (2020)
32. Yang, F., Zhang, L., Yu, S., Prokhorov, D., Mei, X., Ling, H.: Feature pyramid and hierarchical boosting network for pavement crack detection. IEEE Transactions on Intelligent Transportation Systems 21(4), 1525–1535 (2019)
33. Yang, Q., Ji, X.: Automatic pixel-level crack detection for civil infrastructure using unet++ and deep transfer learning. IEEE Sensors Journal 21(17), 19165–19175 (2021)
34. Yang, X., Li, H., Yu, Y., Luo, X., Huang, T., Yang, X.: Automatic pixel-level crack detection and measurement using fully convolutional network. Computer-Aided Civil and Infrastructure Engineering 33(12), 1090–1109 (2018)
35. Zhang, L., Shen, J., Zhu, B.: A research on an improved unet-based concrete crack detection algorithm. Structural Health Monitoring 20(4), 1864–1879 (2021)

36. Zhao, F., Dong, B., Pan, H., Anqi, S.: A mining algorithm to improve lstm for predicting customer churn in railway freight traffic. Studies in Informatics and Control 32, 25–38 (2023)

Wang Xiaofang is with Geely University of China, located in Eastern New District, Chengdu, Sichuan Province (641423). Research interests include computer vision and data mining.

Liu Chenfang is with Chengdu College of University of Electronic Science and Technology of China, located in Gaoxin West District, Chengdu, Sichuan Province (611731). Research interests include convolutional neural network.

Hou Junniang is with Geely University of China, located in Eastern New District, Chengdu, Sichuan Province (641423).Research interests include data analysis and mining.

Zhou Liang is with Geely University of China, located in Eastern New District, Chengdu, Sichuan Province (641423).Research interests include image processing.

Received: July 25, 2023; Accepted: February 07, 2024.

Computer Science and Information Systems 21(4):1457–1481 https://doi.org/10.2298/CSIS230725043Z

Enhancing Architectural Image Processing: A Novel 2D to 3D Algorithm Using Improved Convolutional Neural Networks

Qianying Zou[1], Fengyu Liu[2,⋆], and Yuan Liao[1]

[1] Geely University of China
Chengdu, 641423, China
{56471843,29497521}@qq.com
[2] Geely University of China
Chengdu, 641423, China
i@liufengyu.cn

Abstract. In light of the escalating advancements in architectural intelligence and information technology, the construction of smart cities increasingly necessitates a higher degree of precision in architectural measurements. Conventional approaches to architectural measurement, characterized by their low efficiency and protracted execution time, need to be revised to meet these burgeoning demands. To address this gap, we introduce a novel architectural image processing model that synergistically integrates Restricted Boltzmann Machines (RBMs) with Convolutional Neural Networks (CNNs) to facilitate the conversion of 2D architectural images into 3D. In the implementation phase of the model, an initial preprocessing of the architectural images is performed, followed by depth map conversion via bilateral filtering. Subsequently, minor voids in the images are rectified through a neighborhood interpolation algorithm. Finally, the preprocessed 2D images are input into the integrated model of RBMs and CNNs, realizing the 2D to 3D conversion. Experimental outcomes substantiate that this novel model attains a precision rate of 97%, and significantly outperforms comparative algorithms in terms of both runtime and efficiency. These results compellingly corroborate our model's superiority in architectural image processing, enhancing measurement accuracy and drastically reducing execution time.

Keywords: Building image; Boltzmann machine; Convolution neural network; Bilateral filtering; Neighborhood difference; 2D to 3D.

1. Introduction

As smart cities undergo rapid development, the digitization of architecture has emerged as a pivotal trend in modern urban development and transformation within the real estate industry. In this context, the enhancement of architectural measurement accuracy becomes increasingly critical. However, traditional methods of measurement, such as manual and mechanical techniques, are inefficient and cumbersome, failing to meet the demands of contemporary society [25]. These approaches are labor-intensive and time-consuming and

⋆ Corresponding Author

fall short in ensuring accuracy when dealing with complex architectural forms. Furthermore, the increasing complexity of modern architectural forms exacerbates the limitations of conventional image processing methods, including low accuracy and extended execution time [33].

In the realm of three-dimensional reconstruction, deep learning has manifested significant potential and a broad spectrum of application prospects. Initially, theoretical advantages of deep learning can be observed. For instance, the article [23] introduces a Convolutional Restricted Boltzmann Machine model for image segmentation. Despite its high computational complexity and slow training, hindering its widespread application in 3D reconstruction, the theoretical merits of this approach warrant attention. Secondarily, deep learning technologies have also unveiled new possibilities in practical applications. The article [37] employs an AlexNet Convolutional Neural Network model infused with transfer learning principles, maintaining high accuracy even with limited data, thereby offering a novel perspective for 3D reconstruction. However, certain technical challenges remain to be addressed. For example, the image transformation time in the 11-layer deep Convolutional Neural Network model built upon the GoogLeNet model, as described in the article [43], is excessively lengthy, compromising real-time application efficiency. A small DNN architecture called SqueezeNet is proposed in the literature [30], which deep neural networks for 3D reconstruction applications have improved accuracy, achieving comparable accuracy to AlexNet on the ImageNet dataset but with 50 times fewer number of parameters, which opens up a new path for efficient and accurate 3D model construction, but the architecture suffers from size constraints, computationally intensive problems. A new channel pruning method, Xception, is proposed in the literature [28], which not only significantly reduces the cumulative error, but also enhances the compatibility of the network with a variety of architectures, and can be used to accelerate very deep convolutional neural networks, which significantly improves the efficiency of the network operation while maintaining the accuracy, but the model has limitations in the small-sample constraints, and has a large number of parameters. An efficient model MobileNet for mobile and embedded vision applications is proposed in the literature [29], which uses depth-separable convolution to build lightweight deep neural networks and has wide applicability and superior performance in advanced vision tasks such as 3D reconstruction, but has significant limitations in terms of computational and spatial complexity.

Further research includes the multi-view stereo matching method based on deep learning proposed in the article [10], achieving precise 3D reconstruction through multi-angle image fusion; the in-depth exploration of deep learning in point cloud data processing in the article [24], resulting in fine-grained 3D model reconstruction; and the article [32] overcoming some of the challenging issues in traditional 3D reconstruction by combining deep learning with geometric learning. While these studies offer new perspectives and methods, such as those discussed in articles [31] and [9] concerning indoor and outdoor architectural 3D reconstruction, challenges persist in model generalization, data acquisition, and processing. Consequently, there remains a necessity to explore more efficient and precise methodologies in the field of architectural measurement and to identify suitable deep learning models for solving associated problems. This will not only enhance the accuracy of architectural measurements but also contribute to breakthroughs in the development of architectural digitization. Within this context, the processing of architectural images from two to three dimensions is of particular significance. By transforming

2D architectural images into 3D models, one can both improve measurement accuracy and significantly enhance efficiency, thereby fulfilling the needs of contemporary society [26] [13] [1]. When selecting deep learning models, factors such as complexity, training speed, and accuracy should be considered [5] [20], and the impact of training data volume on model performance should be noted [35] [17].

This study introduces an innovative approach based on a deep learning model aimed at generating corresponding 3D architectural images from inputted 2D architectural images. Compared to existing deep learning methods for 3D reconstruction, the distinctiveness and innovative contributions of this research can be summarized in three main aspects. Firstly, in terms of model architecture, this study amalgamates Restricted Boltzmann Machines (RBMs) with Convolutional Neural Networks (CNNs), creating a stark contrast with standalone deep learning models such as GoogLeNet and AlexNet mentioned in the literature. Secondly, the image preprocessing stage in this research incorporates bilateral filtering algorithms and neighborhood interpolation algorithms, specific preprocessing techniques not explicitly discussed in existing articles. Lastly, this research focuses on the transformation of 2D architectural images into 3D, whereas the scope of literature encompasses a broader array of 3D reconstruction applications, including image segmentation and multi-view stereo matching. The confluence of these three aspects ensures a marked distinction of this study in terms of model architecture, preprocessing techniques, and application domain when compared to extant research, thereby exhibiting unique innovative value. More specifically, the workflow of the model is divided into two modules: image preprocessing and model transformation. In the image preprocessing module, depth acquisition and void-filling are achieved for the original 2D architectural images through bilateral filtering and neighborhood interpolation. Subsequently, in the model transformation module, these preprocessed images are input into the fused RBM and CNN model for 3D conversion. This targeted and specialized approach not only enhances the accuracy of measurements but also significantly boosts efficiency, thus better catering to the specific demands of architectural measurement in modern society.

2. Related Work

2.1. Bilateral Filtering Algorithm

Filtering constitutes a crucial technique in image processing [11], effectively eliminating image disturbances such as noise and blur to enhance image quality. Filters, encompassing both frequency-domain and spatial-domain filters, smooth the image by summing or convolving the pixels, thereby reducing noise and unnecessary detail. Frequency-domain filters operate in the image frequency domain, eliminating specific frequency components; spatial-domain filters operate in the spatial domain, altering pixel values, as indicated in equation1.

$$GC[I]_p = \sum G_\sigma(\|p - q\|) I_q \tag{1}$$

Gaussian filtering represents a significant image processing technique that accomplishes linear smoothing of the image through weighted averaging, effectively eliminating Gaussian noise and enhancing image quality. In Gaussian filtering, the magnitude

of the weights is associated with the spatial distance between pixels, implying that pixels closer to the center pixel exert greater influence on the filtering outcome due to their larger weights. However, this approach has its limitations. If the distance between pixels is excessively large, even adjacent pixels could exhibit considerable disparities in their weights [39] This could result in a loss of detail in the filtering outcome, diminishing the local visual effects of the image. More importantly, such blurring in the filtering results could adversely impact subsequent image processing steps. Therefore, when employing Gaussian filtering, it is imperative to meticulously select appropriate weights to ensure the quality of the filtering outcome.

Bilateral filtering is a nonlinear filtering technique extensively employed in image processing, introduced by Tomasi et al. in 1998 [38]. The uniqueness of bilateral filtering lies in its simultaneous consideration of both the image domain and the value range, enabling it to better preserve image detail during processing. The definition of bilateral filtering closely parallels that of Gaussian filtering, both being realized through weighted averaging based on neighborhood definitions [19]. However, the method of weight measurement in bilateral filtering diverges from that in Gaussian filtering. In bilateral filtering, researchers opt for dual weight values to handle pixel values, thereby more adequately accounting for the variances between adjacent pixels. This approach, nevertheless, has its limitations. If the discrepancies between two pixels are excessively large, they may reciprocally affect each other, inducing unwarranted errors. To circumvent this, researchers introduce a sufficient number of influencing factors in both pixel values and spatial positions to ensure the quality of the filtering result. This method yields favorable outcomes when applied to images with complex textures and details. The definition of bilateral filtering is as shown in equation 2.

$$BF[I]_p = \frac{1}{W_p} \sum\nolimits_{q \in S} G_{\sigma s}(\|p - q\|) G_{\sigma r}(|I_p - I_q|) I_q \tag{2}$$

Herein, W_p represents the normalization factor; I_q denotes the input image; $BF[I]_p$ signifies the filtered image; $\sum\limits_{q \in S} G_{\sigma s}\,||\mathrm{p} - \mathrm{q}||$ is the spatial weight function; and $G_{\sigma r}\,(|I_p - I_q|)\,I_q$ is the distance weight, with the sum of the weights equating to 1, as illustrated in equation 3.

$$W_p = \sum\nolimits_{q \in S} G_{\sigma s}(\|p - q\|) G_{\sigma r}(|I_p - I_q|) \tag{3}$$

Bilateral filtering is particularly well-suited for the preprocessing of depth maps [21]. Depth maps, whether directly acquired through cameras or generated algorithmically, are susceptible to noise. Such noise may result in small voids in the target image during virtual viewpoint synthesis utilizing depth maps. To mitigate this noise, research employs filtering to preprocess depth maps, effectively reducing distortions and voids within them. However, some conventional filtering methods, such as Gaussian or median filtering, risk the loss of edge information in images, leading to cracks and voids. Therefore, the retention of edge information is of paramount importance, as it allows for image smoothing while preserving salient features. Bilateral filters consist of two filter components: one associated with geometric spatial distance and the other with pixel differences. This design enables the bilateral filter to effectively reduce noise and voids while preserving edge in-

formation when processing depth maps, thereby enhancing the overall image processing outcome. The complete bilateral filter is as delineated in equation 4.

$$f(x,y) = \frac{\sum_{(i,j)\in S_{x,y}} w(i,j)g(i,j)}{\sum_{(i,j)\in S_{x,y}} w(i,j)} \tag{4}$$

Herein, $S_{x,y}$ represents the neighborhood of the central point at coordinates (x, y) with a dimension of $(2N + 1) * (2N + 1)$, N is an integer, $g(i, j)$ denotes the original image, and $f(x, y)$ signifies the image post-filtering. The weight coefficient $w(i, j)$ is composed of the product of two parts: the range filtering coefficient $w_r(i, j)$ and the spatial filtering coefficient $w_s(i, j)$, as illustrated in equations 5 and 6.

$$W_r(x,y) = \exp(-\frac{\left|g(i,j) - g(x,y)^2\right|}{2\sigma_r^2}) \tag{5}$$

$$W_s(i,j) = \exp(-\frac{(i-x)^2 + (j-y)^2}{2\sigma_S^2}) \tag{6}$$

Herein σ_S is the spatial proximity factor and σ_r is the luminance similarity factor. For a given image, its value is fixed.

2.2. RBM

Restricted Boltzmann Machines (RBMs) are a derivative of Boltzmann Machines (BMs), which are generative stochastic neural network models capable of learning and representing complex data distributions, thereby generating new samples similar to the training data [42]. However, their training process typically requires extensive computational resources and time. To address this issue, researchers introduced Restricted Boltzmann Machines (RBMs). RBMs substantially simplify the model's architecture and learning process by removing connections within the same layer. In the present study, RBMs are primarily employed for feature extraction from images. The energy function serves as the foundational descriptor of the overall structure and operations of the RBM, as delineated in equation 7.

$$\mathrm{E}(v,h) = -\sum_i a_i v_i - \sum_j b_j h_j - \sum_{i,j} v_i h_j w_{ij} \tag{7}$$

Herein, v and h represent the states of the visible and hidden layers, respectively. a_i and b_i are the bias terms, and w_{ij} denotes the weights.

Layers of the RBM The Restricted Boltzmann Machine (RBM) is composed of two primary layers: the visible layer and the hidden layer [12]. The visible layer serves as the input interface of the RBM and is alternatively referred to as the observational layer. This layer comprises all observable data nodes, each of which represents a specific feature of the data. The principal function of the visible layer is to receive input data and relay this information to the hidden layer. By forming weighted connections with the hidden layer, the visible layer is capable of capturing the rudimentary structure and patterns inherent in the data.

The term "restricted" in Restricted Boltzmann Machine refers to the absence of connections between nodes within the same layer. This distinguishes RBMs from traditional Boltzmann Machines, where intralayer node connections are fully permitted. By eliminating these intralayer connections, RBMs substantially reduce both model complexity and computational requirements. This streamlined architecture not only simplifies the training process but also helps mitigate the risk of overfitting, making the model more scalable to large datasets. RBM achieves effective learning and representation of data distributions through the collaborative training of its visible and hidden layers, and by optimizing the weights between them. The advanced features and concepts captured by the hidden layer facilitate the model's understanding and reconstruction of complex data structures. This makes RBMs a potent tool for feature extraction and generative modeling, and they are widely employed in tasks related to unsupervised learning and deep learning.

The integration of a two-layer architecture—comprising the visible and hidden layers—and restricted connectivity design offers an efficient, flexible, and robust approach for learning and representing complex data distributions. The visible layer directly engages with the input data, while the hidden layer is responsible for capturing higher-order features and structures within the data. This design imbues RBMs with extensive applicability and superiority in handling a wide array of tasks in unsupervised learning and deep learning.

RBM activation function In RBMs, the probability distribution of the nodes in the hidden layer can be articulated using the Sigmoid activation function. Given the state v of the visible layer, the activation probability for the hidden layer nodes h_i is delineated as shown in equation 8.

$$P(h_i = 1|\, v) = \sigma(\sum_i v_i w_{ij} + b_j) \tag{8}$$

Herein, σ represents the Sigmoid activation function, v_i denotes the state in the visible layer nodes i, w_{ij} is the weight between the visible layer nodes i and the hidden layer nodes j, and b_j signifies the bias term for the hidden layer nodes j.

The Sigmoid activation function possesses the characteristic of mapping continuous inputs into a probability distribution. By employing the Sigmoid function, RBMs are capable of converting the linear combinations between the visible and hidden layers into probability values, representing the activation probabilities of the hidden layer nodes. These probability values are subsequently utilized in the sampling and inference processes, aiding the model in learning and generating the underlying data distribution. Through such an activation mechanism, RBMs effectively capture complex patterns and structures within the data, encoding and interpreting them probabilistically.

Loss Function RBMs employ Contrastive Divergence (CD) as their loss function. This study analyzes the limitations of the Contrastive Divergence method and introduces a feedback mechanism to optimize the training process.

The Contrastive Divergence loss function is described as shown in equation 9.

$$CD_n = \langle E(v,h)\rangle_{data} - \langle E(v',h')\rangle_{model} \tag{9}$$

2.3. Convolutional Neural Networks (CNN)

Layers Convolutional Neural Networks (CNNs) are a form of deep feed-forward neural networks [2], where n represents the number of Gibbs sampling steps, and $\langle \cdot \rangle$ symbolizes expectation. The training procedure primarily comprises three key phases: the forward pass, starting from the visible layer and calculating the probabilities in the hidden layer using the Sigmoid activation function, followed by sampling; the backward pass, commencing from the hidden layer and employing the same Sigmoid function to compute the reconstruction probabilities of the visible layer, also followed by sampling; and finally, weight updates, where gradients are computed based on the results of the forward and backward passes and applied using gradient descent methods.

However, two significant drawbacks exist in this training regimen. Firstly, Contrastive Divergence (CD) employs approximation techniques, which may not accurately capture the true data distribution. Secondly, CD is susceptible to local optima, leading to suboptimal training results. To mitigate these issues, this study introduces a feedback mechanism to optimize the training process. Specific improvements include: dynamic adjustment of the learning rate by monitoring loss variations to avoid local minima and accelerate convergence; the incorporation of regularization techniques, specifically L1 regularization, to enhance the model's generalization capabilities and prevent overfitting; and an increase in the number of Gibbs sampling steps to improve the accuracy of CD training and better approximate the true data distribution. This comprehensive approach ensures training accuracy and efficiency, effectively overcoming the limitations of traditional CD methods.

Artificial Neural Networks [7], particularly Convolutional Neural Networks (CNNs), are exceptionally suited for image recognition tasks [6]. The fundamental architecture of CNNs comprises convolutional layers, pooling layers, and fully connected layers.

The objective of the convolutional layer is to extract features from the image through convolution operations, as indicated in equation 10.

$$F_{i,j} = \sum_{m} \sum_{n} K_{m,n} \cdot I_{i+m,j+n} + b \tag{10}$$

Herein, $F_{i,j}$ represents an element of the feature map, $K_{m,n}$ denotes an element of the convolutional kernel, $I_{i+m,j+n}$ signifies an element of the input image, and b is the bias term.

This study employs max pooling, as illustrated in equation 11.

$$P_{i,j} = \max_{m,n} I_{i+m,j+n} \tag{11}$$

The fully connected layer transforms the output from the preceding layer into a one-dimensional array, which is then fed into the subsequent layer. Given N inputs and M outputs, the mathematical representation of the fully connected layer is as indicated in equation 12.

$$O_j = \sum_{i=1}^{N} W_{i,j} \cdot I_i + b_j \tag{12}$$

Herein, O_j represents the value of the output node, $W_{i,j}$ denotes an element of the weight matrix, I_i signifies the value of the input node, and b_j is the bias term.

Activation Functions This study employs the ReLU (Rectified Linear Unit) [3] nonlinear activation function to enhance the nonlinear expressive capability of the Convolutional Neural Network (CNN) model. The mathematical expression of the ReLU activation function is $f(x) = \max(0, x)$, such that its graphical representation appears as a piecewise linear curve. Specifically, the output is zero when the input is less than zero, and the output is equal to the input when it is greater than or equal to zero. Although the ReLU function may appear linear graphically, it is inherently nonlinear, enabling it to handle complex problems, especially within deep learning models. The computation of ReLU is remarkably simple and efficient, requiring only a check to determine whether the input is greater than zero, thereby reducing the demands on training time and computational resources. Additionally, given that the gradient of the ReLU function is 1 in the positive interval, it helps to mitigate the vanishing gradient problem, particularly in deep networks. In CNNs, the ReLU activation function is commonly applied following the convolutional and fully connected layers to increase the model's nonlinear expressive power, thereby enhancing predictive accuracy and generalization capabilities. Overall, due to its straightforward mathematical formulation, nonlinear properties, computational efficiency, and ability to alleviate the vanishing gradient problem, the ReLU activation function finds extensive application in the realm of deep learning. By utilizing ReLU in CNNs, this study aims to construct a more effective and robust model [15].

Loss Function The loss function serves as a metric for assessing the disparity between the model's predictions and the actual labels, and its selection is contingent upon the nature and objectives of the task at hand [14]. In this study, which focuses on classification tasks, Cross-Entropy Loss [34] is employed as the loss function. Cross-Entropy Loss is commonly used in multi-classification problems, and its mathematical expression is denoted as $Loss = -\sum_i y_i \log(\hat{y}_i)$, where y_i represents the i th element of the actual label, usually taking values of 0 or 1, and $\hat{y}_i$ is the i th element of the model's prediction, with a value range between 0 and 1. The role of Cross-Entropy Loss is to measure the divergence between the predicted probability distribution and the true probability distribution of the labels, making it suitable for scenarios where the model's output is a probability distribution. In the context of binary classification, it is equivalent to the log-likelihood loss in logistic regression and imposes a more severe penalty for confidently incorrect classifications. By judiciously selecting and applying Cross-Entropy Loss, this study aims to accomplish accurate classification tasks and effectively guide model training.

3. Method

3.1. Improved Neighborhood Interpolation Method for Small Hole-Filling

This study employs a neighborhood interpolation method for filling small holes, as cited in [27]. Given that most small holes are vertically contiguous—meaning holes exist both above and below the target hole—relying solely on the 8 adjacent pixels of the hole would result in underutilization of surrounding information. Therefore, we extend laterally to include two additional pixels, denoted as A_2 and E_2, as reference pixels for hole-filling, as proposed in [22]. Considering that the correlation between a pixel and the hole increases as they are closer to each other, the weights are assigned in such a manner that pixels

farther away from the hole have smaller weights, while those closer have larger weights, thereby ensuring effective hole-filling. The DIBR method, as per references [40] [36], generates left and right views of the image with different disparities, employing the same filling methodology. The relationship between the holes and the adjacent pixels used for filling is summarized in Table 1.

Table 1. Relationship Diagram between Holes and Adjacent Pixels Used for Filling

A_1	B_1	C_1	D_1	E_1
A_2	B_2	C_2	D_2	E_2
A_3	B_3	C_3	D_3	E_3

Assuming C_2 is a hole point, it is filled using the ten pixels B_1, C_1, D_1, B_2, D_2, B_3, C_3, D_3, A_2 and E_2. However, it is possible that some of the pixels used for filling may also be holes, and not all of these ten pixels may be empty. Therefore, it is essential to discuss the number of hole points within these pixels. Different filling methods are proposed based on the number of filled points within these pixels. If none of the ten pixels are holes, the filling formula is as indicated in equation 13.

$$C_2 = (B_1 + C_1 + D_1 + B_2 + D_2 + B_3 + C_3 + D_3)\,\omega_1 + (A_2 + E_2)\,\omega_2 \tag{13}$$

Where ω_1 and ω_2 are the filling parameters. Let m denote the number of holes in the 8-pixel neighborhood of C_2, and n denote the number of holes in the pixels A_2 and E_2. The number of hole points may have some influence on the filling outcome. If m is less than 8 and n is less than or equal to 2, the hole-filling formula is as indicated in equation 14.

$$C_2 = \frac{8\omega_1}{8-m}\,(B_1 + C_1 + D_1 + B_2 + D_2 + B_3 + C_3 + D_3) + \frac{2\omega_2}{2-n}\,(A_2 + E_2) \tag{14}$$

If $m = 8$ and $n < 2$, the hole-filling formula is as indicated in equation 15.

$$C_2 = \frac{10\omega_2}{2-n}\,(A_2 + E_2) \tag{15}$$

If $m < 8$ and $n = 2$, the hole-filling formula is as indicated in equation 16.

$$C_2 = \frac{10\omega_1}{8-m}\,(B_1 + C_1 + D_1 + B_2 + D_2 + B_3 + C_3 + D_3) \tag{16}$$

Through the methods described above, holes within the image can be effectively filled by utilizing the adjacent pixels, thereby resolving issues related to the presence of holes in the image. However, if the area of the hole is large and there is insufficient pixel information surrounding it, such holes are considered unrepairable large holes. Edge portions generally do not contain critical information; thus, they can be filled using corresponding positions from the original image.

3.2. RBM+CNN for Image Recognition

This study integrates Restricted Boltzmann Machines (RBM) with Convolutional Neural Networks (CNN) algorithms for training on input images. The RBM is responsible for extracting crucial features from the raw images, while the CNN takes on the task of transforming these features into specific recognition outcomes. The architecture of the integrated network is illustrated in Figure 1.

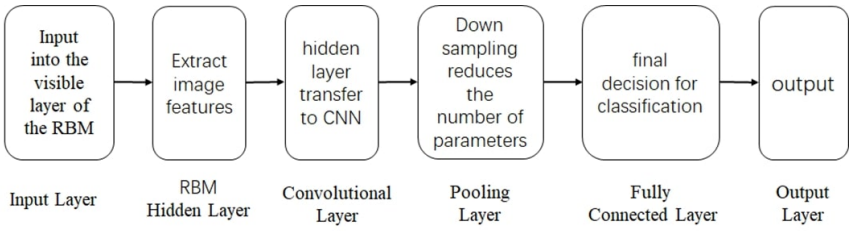


Fig. 1. Conversion effect

As shown in Figure 1, the RBM and CNN integrated network in this study comprises the following key components:

1) Input Layer: Image data are fed into the visible layer of the RBM, with each pixel corresponding to a visible unit. This layer serves to present the raw image data to the network.
2) RBM Hidden Layer: The hidden layer of the RBM is responsible for extracting features from the images. Each hidden unit acts as a feature detector and is not interconnected with other hidden units, only sharing bidirectional connections with the visible units. This design enhances computational efficiency and feature extraction capabilities.
3) Convolutional Layer: Transfers from the RBM's hidden layer to the CNN's convolutional layer for further spatial feature extraction. Each neuron in the convolutional layer is connected to a localized region in the preceding layer and utilizes a shared weight matrix (i.e., convolutional kernel) for computation. This helps capture spatial dependencies within the image.
4) Pooling Layer: Performs downsampling to reduce the number of parameters. Each neuron in the pooling layer connects to a small region in the preceding layer and employs a max-pooling function for computation. This layer minimizes computational demands while retaining feature significance.
5) Fully Connected Layer: Serves as the final decision-making layer for classification. Each neuron in this layer is connected to all neurons in the preceding layer and employs a ReLU activation function for computation. This ensures the network can perform complex nonlinear mappings.
6) Output Layer: Presents the final classification outcome. Each neuron in the output layer corresponds to a specific class and employs a softmax activation function for computation. This ensures that the output is represented as a probability distribution, with each element indicating the predicted probability of the corresponding class.

Overall, the integrated RBM and CNN network structure leverages the strengths of both models, offering a hierarchical approach for image feature extraction and classification, and demonstrating considerable flexibility and performance [16].

This study begins with an analysis of the shortcomings of the Contrastive Divergence method using a Restricted Boltzmann Machine (a sub-model of the Boltzmann Machine) and introduces a feedback mechanism, as cited in [41]. Given the high computational cost of direct gradient use, stochastic sampling is employed for gradient estimation. A subset X' of the original X set is obtained to ensure that their distributions are approximately consistent. The gradient parameters (e, f, v) indicated here require continuous updates in the Contrastive Divergence method of the RBM algorithm. The learning process in the Boltzmann Machine $(\mathrm{e}, \mathrm{f}, \mathrm{v})$ can also be represented as an updating process. Therefore, RBM (e, f, v) can likewise be depicted as an iterative process.

The training and learning process of the model relies on samples obtained through sampling, which directly influence the effectiveness of the training classification. Upon completion of sampling, a feedback mechanism [4] can be introduced to better optimize the parameters. The RBM model is considered in terms of network input (X) and network output (Y), and sampling is performed for both inputs and outputs. The sampling process for X. and Y is defined as sample_positive. It calculates the similarity between Y, X, and X', and the sampled output of Y given X is defined as sample_negative. To obtain optimal parameters, multiple samples are required. The best parameters are identified when the sample distance is reduced or approaches the first node. When the distance changes, the bias could have optimal parameters, and sampling terminates. The number of iterations (C) is arbitrary and depends on the treatment of the sampled examples. For any training sample Xi, the current position of the sample is represented as $X\,(0) = Xi$. The Contrastive Divergence method based on the feedback mechanism is as indicated in equation 17.

$$\Delta W_{ji} = Or_j b_{ji} \tag{17}$$

In the convolutional computations, r_j represents the adjustment coefficient of the sampling element j in the overall connection gradient bias. b_{ji} represents the amount of data acquired by element w_{ji} in a given direction. The corresponding learning rate is denoted by O. The adjustment amount for the i th input weight of the sampling element j can be represented as indicated in equation 18.

$$\Delta P^{l}_{fmap-i} = \sum_{j-1}^{N} convn\left(\Delta P^{l+1}_{fmap-j}, inv\left(K^{l+1}_{ij}\right)\right) \tag{18}$$

In the convolutional computation process, K denotes the convolutional kernel matrix. The feature changes in the convolutional layer must be acquired through backtracking methods. If I represents the sampling layer, then I signifies the convolutional connections. Node i, corresponding to the non-wired layer, has N convolutional rollbacks and can be viewed as the sum of N elements. During hierarchical transmission, the forward convolutional matrix is $PI + 1$ with dimensions $\mathrm{Size}\left(\mathrm{P^I}, 1\right) - \mathrm{Size}\left(\mathrm{K}, 1\right) + 1$. The backward convolutional matrix is PiPi, with corresponding dimensions described as $\mathrm{Size}\left(\mathrm{P^{I+1}}, 1\right) + \mathrm{Size}\left(\mathrm{K}, 1\right) - 1$. Convolutional kernel flipping is employed, and dur-

ing this flipping process, the bias matrix of the ith feature map in the lth layer of M-dimensional convolutional operation is as indicated in equation 19.

$$\Delta K_{ij}^{l} = convn\Delta P_{fmap-j}^{l}, inv\left(P_{fmap-i}^{l-1}\right) \tag{19}$$

The output bias (ΔP_{fmap-i}^{l-1}) is derived from the flipped convolutional kernel $inv\left(K_{ij}^{l+1}\right)$ and the input bias matrix (ΔP_{fmap-i}^{l-1}). Using backtracking methods, the bias of the convolutional kernel matrix can be further obtained from the flipped input bias matrix and output bias matrix, as indicated in equation 20.

$$K_{ij}^{(t)} = K_{ij}^{(t-1)} - \Delta K_{ij} \tag{20}$$

When progressing through layers from $I = 2$ forward, the kernel state at a given time $t - 1$ is obtained by applying the convolutional kernel bias to the kernel state ${K_{ij}}^{(t-1)}$ at the preceding time t, as indicated in equation 21.

$$\Delta W_{ji}^{(t)} = W_{ij}^{(t-1)} - \Delta W_{ij} \tag{21}$$

Through the derived kernel bias function, one can further extend to the states of the ith feature volume matrix at the first layer for different time instances, as indicated in equation 22.

$$\Delta W_{ji}^{(t)} = Or_j b_{ji} + \beta \Delta W_{ji}^{(t-1)} \tag{22}$$

Here, a momentum term is introduced to describe the state of the corresponding feature volume bias matrix, aiming to accelerate convergence. β represents the momentum coefficient, with a value range of [0, 1]. When $\beta = 0$, it is capable of representing the state of the corresponding feature volume bias matrix at time t.

3.3. Image Transformation Results

In the experiment, we selected an indoor corridor as the scene and achieved the transformation from 2D to 3D through different stages of image processing. As depicted in Figure 2, the four stages are as follows: Fig0.2(a) The original 2D grayscale image: This serves as the starting point of the experiment and showcases the basic appearance and shape of the corridor. Fig0.2(b) Foreground and background segmentation: At this stage, the image is segmented into foreground and background, with white representing the foreground and black representing the background. This helps to emphasize the main features of the corridor. Fig0.2(c) Pre-processed depth map: The image undergoes pre-processing to generate a depth map, which contains spatial depth information of the corridor and lays the groundwork for 3D reconstruction. Fig0.2(d) 2D-to-3D reconstruction: Utilizing the previous processing steps, the 2D image is successfully transformed into a 3D effect image, revealing the three-dimensional appearance of the corridor.

Through these four sequential stages, we are able to start with the original 2D grayscale image, progressively unveil the structure of the corridor, and ultimately achieve a 2D-to-3D transformation. Each stage provides necessary information and a foundation for the subsequent stage, forming a coherent and logical processing workflow.



(a) The original 2D grayscale image (b) Foreground and background segmentation (c) Pre-processed depth map (d) 2D-to-3D reconstruction

Fig. 2. Effects of the 2D-to-3D Processing Workflow

4. Result and Discussion

4.1. Experimental Environment

The experiments are conducted using a Dell Precision T7960 graphics workstation, the main parameters of which include the use of an Intel Xeon W7-3465X 28-core processor with a frequency of 2.5 GHz, equipped with a 1 TB or 2 TB SSD and 16 TB mechanical hard disk storage solution, as well as a high-end four NVIDIA RTX 6000 Ada graphics cards. The experiments runs on a Linux operating system, the datasets utilized include the GlobalMLBuildingFootprints open-source dataset, architectural images from ADE20K, and the WHU Building Dataset for building detection.

The model was trained on a curated subset of the GlobalMLBuildingFootprints open-source dataset and validated on architectural images from ADE20K as well as the WHU Building Dataset. In total, 1,317,000 architectural images were used, with 790,000 images in the training set, 263,000 images in the test set, and 261,800 images in the validation set.

4.2. Comparative Experiment on Hole-filling

To enhance the accuracy of image transformation, a hole-filling algorithm was introduced for image pre-processing, specifically targeting the repair of small holes at the edges of the depth map [8]. The evaluation metrics used in this experiment are Peak Signal-to-Noise Ratio (PSNR) and Mean Squared Error (MSE), defined by equations 23, 24, respectively.

$$PSNR = 10 \cdot \log_{10}\left(\frac{MAX_I^2}{MSE}\right) = 20 \cdot \log_{10}\left(\frac{MAX_I^2}{\sqrt{MSE}}\right) \tag{23}$$

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\left|\left|I\left(i,j\right) - K\left(i,j\right)\right|\right|^2 \tag{24}$$

In these equations, $I\left(i,j\right)$ and $K\left(i,j\right)$ represent the pixel values at corresponding coordinates. m and n denote the height and width of the image, respectively. MAX_I is the maximum numerical value representing the color of a pixel in the image.

As well as the structural similarity index (SSIM) and the mean absolute error (MAE), which are defined by the formulas shown in equations 25 and 26:

$$SSIM(P_1, P_2) = \frac{(2\mu_{P_1}\mu_{P_2} + C_1)(2\sigma_{P_1P_2} + C_2)}{(\mu_{P_1}^2 + \mu_{P_2}^2 + C_1)(\sigma_{P_1}^2 + \sigma_{P_2}^2 + C_2)} \tag{25}$$

Where P_1, P_2 represents the original and enhanced images respectively; μ_{P_1}, μ_{P_2} represents the mean of the images P_1 and P_2 respectively; $\sigma_{P_1}^2, \sigma_{P_2}^2$ represents the variance of the images P_1 and P_2 respectively; $\sigma_{P_1P_2}$ represents the covariance of the P_1 and P_2; and C_1, C_2 represents the small constant used to stabilize the denominator.

$$MAE = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}|P_{1ij} - P_{2ij}| \tag{26}$$

Where P_{1ij}, P_{1ij} represent the pixel values of the original and enhanced images at the location (i, j) respectively; m and n are the height and width of the image respectively. The main objective of this study is to measure the quality of the enhanced image by calculating the difference between the gray values of the corresponding pixel points of the enhanced image and the original image by using a statistical method.Both PSNR and MSE measure the good quality of the image by calculating the global magnitude of the pixel error between the enhanced image and the original image. Specifically, a larger value of PSNR indicates that the distortion between the enhanced image and the original image is smaller and the image quality is better. On the contrary, a smaller value of MSE indicates better image quality.SSIM considers the brightness, contrast and structure of an image, and a value closer to 1 indicates better image quality. This metric is useful for evaluating the visual quality of an image during 2D to 3D conversion, and can help determine whether the converted image visually maintains structural features similar to those of the original image.MAE measures the average error at the pixel level, with a smaller value indicating a smaller error, and this metric is valid for evaluating the overall pixel-level error in 2D to 3D conversion.

To verify the effectiveness of this paper's algorithm, by comparing with SqueezeNet algorithm , Xception algorithm and MobileNet algorithm , as shown in Table 4.2. In terms of PSNR, the algorithm in this paper is 38.67 dB, which is an improvement of 2.44% compared with SqueezeNet algorithm, 1.55% compared with Xception algorithm, and 1.31% compared with MobileNet algorithm. Since PSNR is a measure of the quality aspect of image reconstruction, its improvement means that the error of image reconstruction is reduced and the image quality is improved. Although the improvement effect is not obvious, it still indicates a certain progress, which is due to the optimization of the algorithm in image feature extraction and reconstruction model in this paper. In terms of MSE, the mean square error of this algorithm is 8.92, which is 27.71% lower than the SqueezeNet algorithm, 16.24% lower than the Xception algorithm, and 2.19% lower than the MobileNet algorithm. Since MSE is a measure of the difference between images, the lower its value indicates that the error between the reconstructed image and the original image is smaller, the effect of this paper's algorithm is more obvious compared with the previous two algorithms, which should be attributed to the more accurate model training and more appropriate loss function selection. Although the effect is almost the same with MobileNet, there is also a slight improvement. In terms of SSIM, the structural similarity index of this paper's algorithm is 0.9277, which is improved by 7.43% compared with SqueezeNet algorithm, 5.21% compared with Xception algorithm, and 3.72% compared with MobileNet algorithm. Since SSIM is a metric in evaluating the structural fidelity of

an image, its improvement means that the image reconstruction process better preserves the structural features of the original image. Although the improvement is not extremely significant, it shows that the algorithm in this paper has made some progress in this area. This effect is due to the optimization of the algorithm in using more efficient activation functions and more efficient data enhancement techniques. The more efficient activation function may help the network to better capture and convey important features in the image, while the efficient data enhancement technique may enhance the robustness of the model to various changes and perturbations, and thus perform better in preserving the image structure. In terms of MAE, the mean square error of the algorithm in this paper is 3.29, which is 10.35% lower compared to SqueezeNet algorithm, 11.56% lower compared to Xception algorithm and 3.52% lower compared to MobileNet algorithm. Since the MAE is a measure of the difference between images, the lower its value indicates that the error between the reconstructed image and the original image is smaller, the effect of this paper's algorithm is more obvious compared with the previous two algorithms, which should be attributed to the more optimized network architecture of this paper's algorithm and optimization of the specific dataset. Although the effect of this algorithm is similar to that of MobileNet, there is a slight improvement. This suggests that the algorithm in this paper may be more refined in terms of network design, which can handle specific types of image data more effectively, and the optimization for specific datasets may help the algorithm better adapt to and handle these data, and thus achieve better results in terms of MAE metrics.

Table 2. Comparative Experimental Results on Parameters for Hole-filling

Algorithms	PSNR/DB	MSE	SSIM	MAE
SqueezeNet	37.75	12.34	0.8635	3.67
Xception	38.08	10.65	0.8818	3.72
MobileNet	38.17	9.12	0.8944	3.41
Algorithms in this article	38.67	8.92	0.9277	3.29

4.3. Comparison Experiments

Comparison of quantitative results of measurements In this study, the performance of SqueezeNet algorithm, Xception algorithm, MobileNet algorithm and the algorithm proposed in this paper are compared on WHU Building Dataset dataset. To ensure the fairness of the comparison, all algorithms are tested on the same training, test and validation sets. Specifically, each training contains 400 samples and the training process of each algorithm is capped at a maximum of 1000 iterations. The comparison results of the four algorithms are shown in Table 3.

The comparison in terms of number of iterations is discussed first. The number of iterations of this paper's algorithm needs more compared to SqueezeNet algorithm and Xception algorithm, this is because the model structure and optimization strategy of this paper's algorithm needs more iterations to reach convergence. However, this paper's algorithm requires fewer iterations compared to MobileNet algorithm, which indicates that

Table 3. Comparative Experimental Results of Three Algorithmic Models

Algorithms	Number of iterations	Training set loss function	Training set accuracy	Validation set loss function	Validation set accuracy	Measurement accuracy
SqueezeNet	790	0.0231	0.9891	0.2100	0.9579	95.32
Xception	650	0.0229	0.9913	0.0056	0.9714	98.59
MobileNet	920	0.0089	0.9985	0.0039	0.9932	99.21
Algorithms in this article	830	0.0214	0.9946	0.0041	1.0000	99.44

this paper's algorithm outperforms MobileNet algorithm in terms of optimization strategy. Next, the performance of the three aspects of training set loss function, training set accuracy and validation set loss function are analyzed. The algorithm in this paper outperforms SqueezeNet and Xception algorithms in these aspects, which is mainly attributed to the effective loss function optimization, accurate model parameter tuning, excellent feature extraction, and better model generalization ability of the algorithm in this paper. In contrast, the performance of this paper's algorithm in these three aspects is slightly lower compared to the MobileNet algorithm, indicating that the MobileNet algorithm has a slight advantage in these aspects. Finally, consider the performance in both validation set accuracy and measurement accuracy. This paper's algorithm outperforms the other three algorithms in both aspects, which indicates that this paper's algorithm is more accurate and effective in processing validation set data, and has more robustness in practical applications.

In summary, the algorithm in this paper performs well in several aspects, especially in validation set accuracy and measurement accuracy, showing its advantages in practical applications. Although it is slightly inferior to MobileNet in some aspects, overall, the algorithm in this paper demonstrates its effectiveness and robustness in image processing tasks.

Table 4. Quantitative Comparison Experiment Results for 2D Images Across Different Algorithmic Models

Algorithms	Check Accuracy Rate	IoU	Recall Rate	F1 Score	Error rate
SqueezeNet	0.912	0.933	0.954	0.931	0.048
Xception	0.927	0.938	0.96	0.947	0.041
MobileNet	0.958	0.945	0.962	0.962	0.042
Algorithms in this article	0.97	0.948	0.964	0.967	0.041

The experiments in this study involved inputting 8,000 2D images taken in the field into different models for recognition and analyzing them in depth in terms of five aspects: checking accuracy, IoU (intersection and concatenation ratio), recall, F1 score (reconciled mean), and error rate. In terms of detection rate, the detection rate of SqueezeNet algorithm is only 0.912, and after observing the pictures with prediction errors in detail, it is found that the possibility of prediction errors is increased in special areas such as high reflective areas formed by little difference in color, strong external light, or broken concrete floors. In comparison, the algorithm in this paper improves the checking accuracy with

Xception algorithm and MobileNet algorithm by 4.64% and 1.25%, respectively, showing better robustness and discriminative ability for these complex situations. In terms of IoU (intersection-to-union ratio), this paper's algorithm shows a slight improvement compared to the other three algorithms. This result proves that all four algorithms have better ability in locating the target object boundary, while this paper's algorithm can locate the target object boundary more precisely and capture the target morphology more accurately. In terms of recall, this paper's algorithm also has a slight improvement compared with the other three algorithms, this result shows that the four algorithms have comparable abilities in capturing positive samples, and also highlights the slight advantage of this paper's algorithm in recalling positive samples. In terms of F1 scores, this paper's algorithm improves by 3.87%, 2.11% and 0.52% compared to the SqueezeNet algorithm, the Xception algorithm and the MobileNet algorithm, respectively. Since the F1 score is the reconciled average of the detection and recall rates, this improvement reflects the simultaneous optimization of this paper's algorithm in terms of detection and recall rates. In terms of error rate, this paper's algorithm has the lowest error rate with Xception algorithm, which is 14.58% and 2.38% lower than SqueezeNet algorithm and MobileNet algorithm, respectively. This result proves the superiority of this paper's algorithm in providing more accurate classification decisions and more robust feature extraction.

Combining the experimental results in Tables 3 and 4, this paper's algorithm performs better in several key performance metrics, especially in validation set accuracy, measurement accuracy, checking accuracy, and F1 scores than the other compared algorithms. These results show that this paper's algorithm has advantages in model training, generalization ability and accuracy in practical applications. The subtle advantages of this paper's algorithm in other metrics also reflect its overall algorithmic robustness and efficiency. These performance improvements are attributed to the innovation of algorithm design, refinement of feature engineering, and optimization of the training process.

Table 5. Experimental results comparing 2D to 3D running time and efficiency of different algorithm models

Algorithms	Total Time(s)	FPS	FLOPS
SqueezeNet	59.22	51	53.97
Xception	56.37	55	57.81
MobileNet	51.93	62	67.38
Algorithms in this article	36.11	87	87.27

Runtime & Efficiency Comparison In this study, the total time taken by four different algorithms to convert 2000 2D architectural images into 3D images with the same training, test, and validation sets is compared. The study observes the conversion duration (s), frames per second (FPS) of processed images and peak speed per second (FLOPS). In terms of conversion duration, the conversion time of this paper's algorithm is 36.11s, which is 39.02% lower than the running time of SqueezeNet algorithm, 35.94% lower than the running time of Xception algorithm, and 28.06% lower than the running time of

MobileNet algorithm; this result is due to the fact that the algorithm of this paper has improved the traditional CNN This result is attributed to the improvement of the traditional CNN network structure and the precise setting of the weight initialization value of this algorithm. In terms of FPS, the algorithm in this paper improves significantly compared with the other three algorithms, which is attributed to the improvements in image preprocessing, feature extraction, reduction of redundant computation, optimization of memory access, and enhancement of parallel computation capability. In terms of FLOPS, the algorithm in this paper has a significant advantage over the other three algorithms in terms of processing speed per second, which is attributed to the optimization of the computational density of the algorithm, which reduces unnecessary floating-point operations, improves the parallelism of the algorithm, and optimizes the computationally intensive part of the code. In summary, the significant improvement in conversion time, FPS and FLOPS of this paper's algorithm reflects the optimization and improvement of the algorithm in several key aspects, such as computational efficiency, data preprocessing, feature extraction, parallel computing, resource management and code implementation, which proves that this paper's algorithm can efficiently and accurately implement the 2D to 3D conversion technology.

Ablation Study The present study introduces a novel image reconstruction algorithm that amalgamates the advantages of both Convolutional Neural Networks (CNN) and Restricted Boltzmann Machines (RBM). The objective is to achieve more precise image reconstruction. To validate the superiority of the proposed algorithm over traditional CNN and RBM techniques in terms of image reconstruction quality, we have designed a comparative experiment.

Table 6. 2D picture quantitative ablation comparison experiment results

Algorithms	Check Accuracy Rate	IoU	Recall Rate	F1 Score	Error rate
CNN	0.665	0.701	0.782	0.719	0.081
RBM	0.73	0.715	0.759	0.745	0.076
Algorithms in this article	0.97	0.948	0.964	0.967	0.041

The experiment involved the input of 8,000 on-site captured 2D images into three different algorithms for identification. The results, as shown in Table 6, indicate that the algorithm proposed in this study outperforms traditional CNN and RBM models across five key metrics, substantiating its effectiveness. This superiority can be elucidated from several perspectives:

Architectural Advantage: The proposed algorithm integrates the convolutional layers of CNN with the generative model features of RBM. This amalgamation leverages the strengths of both, resulting in higher precision and robustness in image reconstruction.

Efficient Training: By adopting more precise weight initialization, the training process for our algorithm is both efficient and stable. This enhances the model's ability to fit the

training data more effectively, thereby improving its 3D reconstruction performance on the on-site captured 2D images.

Function Selection: The algorithm employs loss and activation functions that are particularly suited to this specific task. The appropriate choice of loss function better guides model learning, while the optimal activation function enhances the model's non-linear representational capability.

In summary, the proposed algorithm's superiority over traditional CNN and RBM models is primarily attributable to its comprehensive optimization in structural design, training strategy, and function selection. These collectively contribute to a significant advantage in 3D reconstruction tasks for on-site captured 2D images.

Table 7. Ablation Study on Run Time and Efficiency for 2D-to-3D Conversion

Algorithms	Total Time(s)	FPS	FLOPS
CNN	124.89	31	27.92
RBM	119.54	34	31.31
Algorithms in this article	36.11	87	87.27

Under a unified framework of identical training, testing, and validation sets, this study meticulously contrasts the efficacy of three disparate algorithms—our proposed algorithm, traditional Convolutional Neural Networks (CNN), and Restricted Boltzmann Machines (RBM)—in the task of transforming 2,000 2D architectural images into 3D models, as detailed in Table 7. Our observations indicate that the proposed algorithm significantly outperforms the traditional counterparts across various metrics, including total conversion time, frames processed per second (Fps), and peak floating-point operations per second (FLOPs). Firstly, this substantial advantage can be attributed to the architectural refinements incorporated into the proposed algorithm. By synergistically leveraging the convolutional attributes of CNNs and the generative capabilities of RBMs, the algorithm enhances both the efficiency and accuracy of image conversion tasks. Secondly, the marked elevation in computational efficiency underscores the algorithm's prowess in diminishing computational complexity and overall operational time. This not only expedites the training phase but also augments the model's performance in handling intricate image data. Finally, the flexibility and precision of the proposed algorithm are also manifest. Optimized for task-specific needs, the algorithm minimizes unnecessary floating-point calculations and redundant computations. As a result, it significantly amplifies operational speed, establishing new benchmarks in both frames processed per second and peak operational speed.

In summary, the comprehensive optimization of the proposed algorithm in terms of architecture, efficiency, and flexibility renders it exceptionally proficient in 2D-to-3D image conversion tasks. It substantially transcends traditional CNN and RBM algorithms, thereby demonstrating its potent potential and applicability.

Comparative Performance of 2D-to-3D Image Reconstruction Across Algorithms As shown in Figure 3, the results of three different algorithms are demonstrated on the

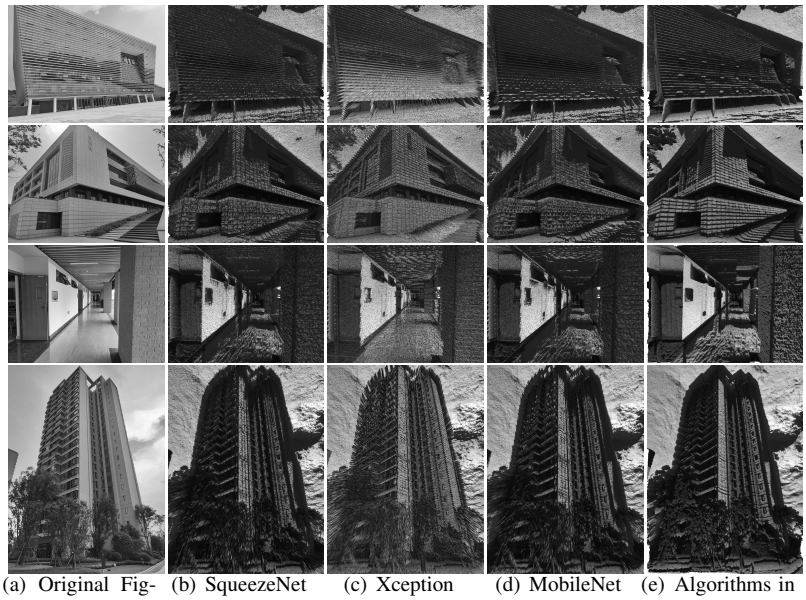
task of 2D to 3D image reconstruction on library image, academic building image, corridor image and residential building image. Image.3(a) is the original image, and images Image.3(c) to Image.3(e) show the processing results of SqueezeNet algorithm, Xception algorithm, MobileNet algorithm, and this paper's algorithm, respectively. SqueezeNet algorithm, as an earlier deep learning model, has a relatively simple structure and cannot capture all the subtle features of the image, so the detail features are weak. The Xception algorithm, with a more complex structure and more parameters, is better than the SqueezeNet algorithm but still has room for improvement. The MobileNet algorithm, an efficient neural network model, achieves minimizing the demand for computational resources while maintaining good performance by employing depth-separable convolutions and specific network structural adjustments, but it is not effective in dealing with extremely complex or fine-grained tasks, there is still room for improving its performance.

In contrast, our proposed algorithm excels in 2D-to-3D image reconstruction, manifesting a marked enhancement in detail fidelity. This superiority can be attributed to three key facets: (1)Task-Specific Optimization: The algorithm has been fine-tuned for specific types of images and content, enabling it to better capture intricate details. (2)Loss Function and Optimization Strategy: The selection of a loss function and optimization techniques particularly tailored for 2D-to-3D conversion tasks further augments the quality of image reconstruction. (3)Precise Weight Initialization and Efficient Training: The algorithm employs a highly accurate weight initialization process and an optimized training methodology, contributing to efficient and stable training.

In summation, the proposed algorithm stands as the most proficient in reconstructing 3D images from 2D precursors, owing to its comprehensive optimization across task-specific requirements, loss function selection, and training procedures. Its exemplary performance underscores its robustness and applicability in complex image transformation tasks.

In this study, we incorporate various performance indicators such as the TOP-5 [18] error rate, the number of convolutional layers, the size of the convolutional kernels, and the number of fully connected layers to compare the efficacy of different algorithmic models. The TOP-5 error rate serves as a critical evaluation metric that considers a prediction as correct if the true label is among the top five probabilities in the final output vector. A lower TOP-5 error rate is indicative of better model performance. Taking the library image reconstruction effect as an example, the Top-5 error rates of different algorithm models can be observed in Table 8, and the four algorithms show different performances. the SqueezeNet algorithm has a relatively high error rate of 6.7%, which is not able to adequately capture the complex features of the image due to its relatively simple structure. the Xception algorithm and MobileNet algorithm have reduced the error rates to 5.9% and 4.9%, which is a significant improvement compared to the first algorithm, but there is still room for improvement. Proposed Algorithm: Demonstrates a significantly reduced TOP-5 error rate of 3.1%, emphasizing the algorithm's superiority in image reconstruction tasks.

The decline in error rates can be attributed to increased algorithmic complexity and task-specific optimizations. The proposed algorithm, through its architectural refinements and innovative methods, achieves a more accurate image reconstruction. Its substantially lower error rate compared to the other algorithms corroborates its excellence in image reconstruction tasks.



(a) Original Figure (b) SqueezeNet (c) Xception (d) MobileNet (e) Algorithms in this article

Fig. 3. Algorithm effect diagram comparison

In the subsequent section, we delve into the significance of convolutional layers. Images are composed of pixels, which may contain noise. Convolutional layers serve the purpose of extracting features from these noisy pixels. As the number of convolutional layers increases, these rudimentary features combine to form higher-level patterns. Consequently, a deeper convolutional network is capable of capturing more abstract and comprehensive features. With 151 convolutional layers, our proposed algorithm excels in exhaustive detail extraction, contributing to the increased accuracy of the generated 3D images.

Furthermore, to reduce computational and parameter overhead while accelerating model training, smaller convolutional kernels are beneficial. They allow for the inclusion of additional convolutional layers, thereby enhancing the network's nonlinear capabilities. On another note, a single fully connected layer is insufficient for condensing feature data to a degree where accurate judgments can be made. The presence of two layers in our algorithm ensures more concentrated feature data.

In summary, when considering multiple evaluation metrics such as the TOP-5 error rate, the number of convolutional layers, the size of the convolutional kernels, and the number of fully connected layers, our proposed algorithm outperforms other models in terms of overall efficacy. The architecture of our model is well-balanced in terms of convolutional kernel size and the number and size of fully connected layers, making it a more potent solution for the task at hand.

Table 8. Experimental results of performance comparison of different algorithms for library image reconstruction

Algorithms	TOP-5 Error Rate	Number Of Convol -utional Layers	Convol -utional Kernel Size	Number Of Fully Connected Layers
SqueezeNet	6.7%	115	7,1,3,5	2
Xception	5.9%	118	7,1,3,5	2
MobileNet	4.9%	132	11,7,5,3	2
Algorithms in this article	3.1%	151	7,1,3,5	2

5. Conclusion

A new model for architectural image processing is developed in this study. This model combines a Restricted Boltzmann Machine (RBM) and a Convolutional Neural Network (CNN) specifically designed to convert two-dimensional (2D) architectural images into three-dimensional (3D) models. The input building images are first preprocessed to ensure their suitability. Then, a bilateral filtering technique is used to convert the image into a depth map, laying the foundation for subsequent 3D reconstruction. Next, small voids in the image are repaired by a domain difference algorithm, thus improving the integrity and quality of the image. The last key step is to input the processed 2D image into the research-designed model. The model incorporates the features of RBM and CNN to effectively realize the conversion from 2D image to 3D model. After thorough testing, the study found that the model achieved 97% in terms of checking accuracy and significantly outperformed other existing algorithms in terms of runtime and efficiency.

However, it is crucial to acknowledge the limitations of our current algorithm. For instance, the likelihood of prediction errors increases when dealing with areas of subtle color variations, strong external lighting causing high reflectivity, or surfaces with damage mimicking water stains. This suggests that further optimizations are required for our algorithm to adeptly handle these challenging scenarios. Looking ahead, we anticipate continual refinement of our algorithm in two primary directions:

Enhanced Models and Techniques: We aim to incorporate more advanced deep learning models and techniques to improve both the accuracy and robustness of our algorithm. Diverse Applications: We will explore the utility of our algorithm in processing other types of architectural images and even in other image processing tasks, with the goal of widespread application and dissemination.

By addressing these aspects, we aim to extend the applicability and fortify the resilience of our algorithm, thereby contributing to its broader adoption in the domain of image processing.

Acknowledgments. Funding project: 2022 Sichuan Province Science and Technology Innovation Project in the field of housing and urban-rural construction, "Research and Application of Intelligent Detection of Underground Space Leakage Based on Computer Vision" (NO: SCJSKJ2022-05).

References

1. 3D scene description and recording mechanis: A comparative study of various 3D create tools. Journal of System and Management Sciences
2. Deep learning to detect image forgery based on image classification. Journal of System and Management Sciences
3. Evaluation of algorithmic metrics with a focus on server Cyber-Risks. Evaluation of Algorithmic Metrics with A Focus on Server Cyber-Risks. Journal of System and Management Sciences
4. Factors affecting students' intention to use mobile learning at universities: an empirical study. Journal of System and Management Sciences
5. Identify faults in road structure zones with deep learning. Journal of System and Management Sciences
6. Inventory policy for electric vehicle battery swapping stations in beijing. Journal of System and Management Sciences
7. rediction of Vibration in the Discharge Ring of a River Type Hydroelectric Power Plant with Bulb Turbine Using Artificial Neural Networks and Support Vector Machine
8. Root cause analysis for it incident using artificial neural network (ANN). Journal of System and Management Sciences
9. 3D reconstruction using SfM with sequence images in hadoop. Journal of System and Management Sciences (2019)
10. Image and graph restoration dependent on generative adversarial network algorithm. Tehnicki Vjesnik 28(6) (2021)
11. Image completion based on edge prediction and improved generator. Tehnicki Vjesnik 28(5) (2021)
12. Neural network driven automated guided vehicle platform development for industry 4.0 environment. Tehnicki Vjesnik 28(6) (2021)
13. An innovative photogrammetric system for 3D digitization of dental models. Tehnicki Vjesnik 29(5) (2022)
14. Motor imagery EEG recognition using deep generative adversarial network with EMD for BCI applications. Tehnicki Vjesnik 29(1) (2022)
15. Corporate online learning as shared service based on ADDIE model. Journal of System and Management Sciences 14(4) (2023)
16. Machine learning approaches for precision medicine: A review. Journal of System and Management Sciences 14(4) (2023)
17. Aggarwal, H.K., Mani, M.P., Jacob, M.: MoDL: Model-based deep learning architecture for inverse problems. IEEE Trans. Med. Imaging 38(2), 394–405 (2019)
18. Ayachi, R., Afif, M., Said, Y., Atri, M.: Strided convolution instead of max pooling for memory efficiency of convolutional neural networks. In: Smart Innovation, Systems and Technologies, pp. 234–243. Springer International Publishing, Cham (2020)
19. Chen, B.H., Tseng, Y.S., Yin, J.L.: Gaussian-Adaptive bilateral filter. IEEE Signal Process. Lett. 27, 1670–1674 (2020)
20. Chen, Y., Hu, S., Mao, H., Deng, W., Gao, X.: Application of the best evacuation model of deep learning in the design of public structures. Image Vis. Comput. 102(103975), 103975 (2020)
21. Chi, X., Meng, Q., Wu, Q.: A laser data compensation algorithm based on indoor depth map enhancement. J]. Electronics 12(12) (2023)
22. Cho, S.I., Kang, S.J.: Temporal incoherence-free video retargeting using foreground aware extrapolation. IEEE Trans. Image Process. 29, 4848–4861 (2020)
23. Eslami, S.M.A., Heess, N., Williams, C.K.I., Winn, J.: The shape boltzmann machine: A strong model of object shape. Int. J. Comput. Vis. 107(2), 155–176 (2014)
24. Fei, B., Yang, W., Chen, W.M., Li, Z., Li, Y., Ma, T., Hu, X., Ma, L.: Comprehensive review of deep learning-based 3D point cloud completion processing and analysis. IEEE Trans. Intell. Transp. Syst. 23(12), 22862–22883 (2022)

25. Figliola, R.S., Beasley, D.E.: Theory and design for mechanical measurements. Meas. Sci. Technol. 7(7) (1996)
26. Gimenez, L., Robert, S., Suard, F., Zreik, K.: Automatic reconstruction of 3D building models from scanned 2D floor plans. Autom. Constr. 63, 48–56 (2016)
27. Guo, W.Y., Wang, Y., Dai, F., Xu, P.: Improved sine cosine algorithm combined with optimal neighborhood and quadratic interpolation strategy. Eng. Appl. Artif. Intell. 94(103779), 103779 (2020)
28. He, Y., Zhang, X., Sun, J.: Channel pruning for accelerating very deep neural networks. In: 2017 IEEE International Conference on Computer Vision (ICCV). IEEE (2017)
29. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: MobileNets: Efficient convolutional neural networks for mobile vision applications (2017)
30. Iandola, F.N., Han, S., Moskewicz, M.W.: SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and¡ 0.5 MB model size (2016)
31. Kang, Z., Yang, J., Yang, Z., Cheng, S.: A review of techniques for 3D reconstruction of indoor environments. ISPRS Int. J. Geoinf. 9(5), 330 (2020)
32. Luo, Y., Mi, Z., Tao, W.: DeepDT: Learning geometry from delaunay triangulation for surface reconstruction. Proc. Conf. AAAI Artif. Intell. 35(3), 2277–2285 (2021)
33. Mahdjoubi, L., Moobela, C., Laing, R.: Providing real-estate services through the integration of 3D laser scanning and building information modelling. Comput. Ind. 64(9), 1272–1281 (2013)
34. Mlayeh, H., Othman, K.B.: Nonlinear accommodation of a DC-8 aircraft affected by a complete loss of a control surface(2022). Studies in Informatics and Control 31(3), 107–116 (2022)
35. O'Mahony, N., Campbell, S., Carvalho, A., Harapanahalli, S., Hernandez, G.V., Krpalkova, L., Riordan, D., Walsh, J.: Deep learning vs. traditional computer vision. In: Advances in Intelligent Systems and Computing, pp. 128–144. Springer International Publishing, Cham (2020)
36. Pamucar, D., Simic, V., Lazarević, D., Dobrodolac, M., Deveci, M.: Prioritization of sustainable mobility sharing systems using integrated fuzzy DIBR and fuzzy-rough EDAS model. Sustain. Cities Soc. 82(103910), 103910 (2022)
37. Rafiq, M., Rafiq, G., Agyeman, R., Jin, S.I., Choi, G.S.: Scene classification for sports video summarization using transfer learning. Sensors (Basel) 20(6), 1702 (2020)
38. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271). Narosa Publishing House (2002)
39. Vijaya Kumar, S., Nagaraju, C.: T2FCS filter: Type 2 fuzzy and cuckoo search-based filter design for image restoration. J. Vis. Commun. Image Represent. 58, 619–641 (2019)
40. Wang, G., Wang, Z., Gu, K., Li, L., Xia, Z., Wu, L.: Blind quality metric of DIBR-synthesized images in the discrete wavelet transform domain. IEEE Trans. Image Process. 29, 1802–1814 (2019)
41. Xiong, W., Zhu, Y., Zeng, Q., Du, J., Wang, K., Luo, J., Yang, M., Zhou, X.: Dose-effect relationship analysis of TCM based on deep boltzmann machine and partial least squares. Math. Biosci. Eng. 20(8), 14395–14413 (2023)
42. Xu, A., Tian, M.W., Firouzi, B., Alattas, K.A., Mohammadzadeh, A., Ghaderpour, E.: A new deep learning restricted boltzmann machine for energy consumption forecasting. Sustainability 14(16), 10081 (2022)
43. Zaidi, S.S.A., Ansari, M.S., Aslam, A., Kanwal, N., Asghar, M., Lee, B.: A survey of modern deep learning based object detection models. Digit. Signal Process. 126(103514), 103514 (2022)

Qianying Zou (born in 1980) is a Professor from Chengdu, Sichuan, China. She holds a Master's degree and specializes in the fields of big data applications and artificial intelligence. Geely University of China, No. 123, Section 2, Chengjian Avenue, East New District, Chengdu, Sichuan Province. E-mail: 56471843@qq.com

Fengyu Liu (born in 1996) is an Assistant Lecturer from Xi'an, Shaanxi, China (Corresponding author). He holds a Bachelor's degree and focuses on data mining and front-end and back-end design. Geely University of China, No. 123, Section 2, Chengjian Avenue, East New District, Chengdu, Sichuan Province E-mail: i@liufengyu.cn

Yuan Liao (born in 1982) is a Lecturer from Guang'an, Sichuan, China. She holds a Master's degree and her research interests include engineering management and virtual reality. Geely University of China, No. 123, Section 2, Chengjian Avenue, East New District, Chengdu, Sichuan Province. E-mail: 29497521@qq.com

Received: July 25, 2023; Accepted: January , 2024.

Computer Science and Information Systems 21(4):1483–1498 https://doi.org/10.2298/CSIS230829044L

Learning Discriminative Representations through an Attention Mechanism for Image-based Person Re-identification

Jing Liu[1] and Guoqing Zhou[2]

[1] School of Computer Science, Weinan Normal University,
Weinan 714099, Shaanxi, China
liujing8318@mail.nwpu.edu.cn
[2] School of Computer Science, Northwestern Polytechnical University,
Xi'an, 710072, Shaanxi, China
zhouguoqing@nwpu.edu.cn

Abstract. Over the past years, person re-identification has been obtaining various attentions in computer vision tasks. However, existing methods mainly focus on building massive number of deep architecture layers, which is unsuitable for extracting the robust features for person re-ID. In this paper, we present a novel hybrid framework PGAN, through which the discriminative representations can be learned for person re-ID. Specifically, a novel self-attention method named channel-wise attention mechanism is adopted to learn the informative representations from the patch-network and global network, respectively. In addition, CSwin Transformer is exploited to re-extract the discriminative features from the residual blocks. We obtain a mAP of 81.8% and 80.3% of the labeled and detected dataset on the CUHK0-NP dataset. And we obtain a mAP of 83.4% and 91.3% on the DukeMTMC and Market-1501 datasets respectively. Comprehensive experiments are performed on the three datasets, (Market-1501, DukeMTMC-reID and CUHK03-NP), demonstrating the efficiency of the introduced approach.

Keywords: person re-identification, channel-wise attention, deep learning.

1. Introduction

Person re-identification (re-ID) task aim to identify a person through video samples from different direction cameras. Along with the rapid development of deep learning (DL), various researchers have shifted from hand-crafted feature engineering [26] to deep learning techniques [19,39,50,54,44,46,45,40,12]. For example, in [53,21,51], the authors adopt DL technology to capture the robust features from the video samples for person re-ID. In addition, the attention mechanism attempts to be adopted in numerous studies [23,20,18,23,15,49,35] to represent the notable parts (e.g., spatial locations and etc.) by convolutional responses and filter the useless parts (e.g., background). Now, different attention schemes are adopted to represent great performances for person re-ID [20]. Yet, the general attention approaches can only be adopted to capture the coarse features, which is not able to learn complicated features from the visual sections for person re-ID. Moreover, several features can include some informative representations of attentional feature maps for person re-ID. Hence, the authors use a new attention mechanism to learn high-level representations. In addition, to extract the robust representations with all instances,



Fig. 1. These images sampled from four person re-ID datasets, representing that informative representations are important for person re-ID. Each sub-figure group consists of the three sub-figures (query, true, false)

the researchers consider that all data instances can be seen as a a similarity graph in person re-ID task. Also they combine the high-level and spectral features to promote the accuracy for person re-ID. Especially, through [27,56,41,48], the important features of person re-ID are also learned. And [27] proposes a deep graph metric learning (DGML) for person re-ID, which can learn the consistency from intra-video spatial graphs of consecutive frames, then the spatial graph learns neighborhood relationships of the detected person instances of each frame. For the another aspect, DGML can capture the distinguished features among the inter-video spatial graphs for person re-ID. In addition, the development of hand-crafted features needs a great number of factors (e.g., domain experience, time, etc.). Hence, bags and books can result in the poor performances of person re-ID. Furthermore, to learn the features of bags, scientists should learn domain-specific knowledge, which needs lots of time. This is the motivation of our work to design a robust architecture for person re-ID.

Up to now, through the various end-to-end DL methods, promising performances has been achieved in person re-ID, and the following problems still existed. The first problem is the representation of discriminative features. Current methods are mainly focused into two aspects by the point of feature representations (i.e., hand-crafted and deep learned features.). Though hand-crafted features have been proven to be obtaining in person re-ID based on hand-crafted features. Yet, there are still exists several problems. For instance, the designed of hand-crafted features acquire numerous aspects (e.g., domain experience, labor, etc). For example, bags and books can influence the accuracy for person re-ID. In addition, to capture the features of bags, researchers need to acquire task-specific knowledge to waste a lot of time. As shown in Fig. 1, to capture the informinative representations in the images, one should overcome this difficulties (such as the black bag in (a)) in the databases. For the rest of the examples in Fig. 1, we can observe that the bags can bring some problems to feature learning. More importantly, different databases have the same problems while extracting the informative features, in which a robust architecture also needs to be designed for feature learning.

In addition, another problem is the attentive feature learning. Although most of the DL methods for person re-ID have a discriminative feature extraction ability, the importance of the association between high-level semantic feature features and channels is ignored in

the feature learning step in person re-ID. For example, as shown in Fig. 1 (a), to extract the informative features, we should filter the bags of noisy features and capture the discriminative features for person re-ID. Also, the useless features lead to a reduced scalability of the model, which makes it hard for real-world large-scale usage. Meanwhile, the useless features can result in the problem of overfitting in the process of model training, which also brings some barriers for the real-world application.

In our work, to address the above-mentioned issues, we propose a novel architecture, named patch-global attention network (PGAN), for addressing person re-ID. Specifically, PGAN contains two components, i.e., patch-network and global-network, which also contains a channel-wise attention mechanism to learn the patch-global feature representations. For the patch-network, our aim is to learn the patch features, e.g., the face, the body, and other patches. This is because that different parts of the body contain different semantic information of features. Therefore, these different convolutional neural networks (CNN) branches can be encoded in the patch-network and informative feature representations for person re-ID can be captured. Through the global-network the global representations for person re-ID can be learned. More importantly, a channel-wise attention mechanism is adopted to learn the local and global features as well as fuse the different scale features for person re-ID. In addition, Swin Transformer is used to re-extract the discriminative features from the residual blocks.

The contributions of the presented paper are three-aspects:

- We introduce a novel framework, named PGAN, for modeling the multi-scale features for person re-ID. CSwin Transformer is adopted to extract the informative features from the residual blocks.
- In the architecture, a novel attention method, named channel-wise attention mechanism is adopted to learn the discriminative information through the patch-network and global-network, respectively.
- Comprehensive experiments are performed on some benchmark datasets, i.e., Market-1501, DukeMTMC-reID and CUHK03-NP, demonstrating the efficiency of the introduced approach.

The remainder of the presented paper is structured as follows. Several existing works are described in Section 2. In Section 3, the proposed deep architecture is described. The used databases and experimental performances are described in Section 4. In Section 5, conclusions and future works are described.

2. Related Works

In this part, we briefly overview the existing works on person re-ID. We first introduce the hand-crafted features, and describe the deep features for person re-ID.

2.1. Hand-crafted Features of Person Re-ID

Hand-crafted features are extracted for person re-ID in various studies [13]. A weighted color topology (WCT) feature is designed to model the spatial distribution features. In [47], it is found that the fusion of histogram features of oriented gradient (HOG) descriptor, HSV, and color histogram features is important for person re-ID. Meanwhile, in

[9], the authors also propose to use region-based feature salience for person re-ID. However, there still exists the problem that we have not extracted the high-level features, e.g., context information and semantic features. Hence, various studies are emerged with the development of DL. We briefly introduce the deep-learned features of person re-ID as follows.

2.2. Deep-learned Features in Person Re-ID

In the work of [28], Rao et al. propose a self-supervised gait encoding method through which unlabeled skeleton data can be learned to capture gait representations for person re-ID. The core idea of the introduced approach is that high-level semantics information is learned through a self-supervision mechanism. In addition, they also propose a locality-aware attention mechanism and a locality-aware contrastive learning scheme to model and maintain locality-awareness on an intra-sequence level and inter-sequence level in the self-supervised learning step. Finally, a new feature named constrastive attention-based gait encodings (CAGEs) is proposed to characterize gait effectively.

Bai et al. [1] collect a new large-scale dataset for person re-ID. Additionally, they also proposed a domain generalization re-ID architecture named dual-meta generalization network (DMG-Net), to learn the merits of meta-learning in the training step.

Two constructive modules are proposed in [2], i.e., a rectification domain-specific batch normalization (RDSBN) module and a graph convolutional network (GCN) based multi-domain information fusion (MDIF) module for the domain-specific view and domain-fusion view, respectively. RDSBN can be adopted to simultaneously reduce domain-specific characteristics and increase the distinctiveness of personal features. MDIF can be used to minimize field distances via combing representations of different fields.

In [42], the authors introduce a new pairwise loss function, through which fine-grained features can be learned by enforcing an exponential penalization on the images of small differences and a bounded penalization on the images of large differences.

Deep graph metric learning (DGML) is introduced of [27] for person re-ID, through which the consistency from intra-video spatial graphs of consecutive frames can be re-weighted, which the spatial graph capture neighborhood relationship in the detected person instances of each frame. From the another perspective, by using the DGML approach, the inter-video spatial graphs learned by numerous camera views at different aspects are identified. In order to further represent deploy a weak supervision into the DGML and solve the problem of weakly supervised person re-ID, weakly supervised regularization (WSR) is introduced to learn discriminative features by means of a weak identity loss and a cross-video alignment loss.

In [11], Hou et al. introduces a spatial-temporal architecture for video-based person re-ID. Concretely, a bilateral complementary network (BiCnet) is proposed to model spatial complementary information. The combination of a detail branch and context branch is used to model the original and long-range characteristic information. For every branch, biCnet can combine multiple parallel and diverse attention modules to discover divergent body parts for consecutive frames to obtain an integral characteristic of target identity. Moreover, a new block is designed to capture different scale features. The new block can be easily embed into BiCnet for spatial-temporal modeling.

Modeling static images with identity-related and unrelated features are proposed in [7]. Generative adversarial network is adopted to represent identity-related and unrelated

features of person images. Specifically, an identity-shuffling method is used to reduce the noisy labels without any additional methods.

It tries to overcome the unsupervised domain adaptation in [14] from a novel perspective for person re-ID. A cluster-wise contrastive learning algorithm (CCL) is adopted to maintain the feature extraction and to learn noise-tolerant features by a new unsupervised scheme. Meanwhile, they adopt a progressive domain adaptation (PDA) method is used to narrow the domain gap between source and target data. After that, fourier augmentation scheme is also utilized to maximize the size of class separability of re-ID models.

In [56], the authors propose a new architecture, named omni-scale network (OSNet) to model the different spatial scale features. The combination of multiple convolutional streams in the basic building block is used to learn the deep features. A unified aggregation gate of the omni-scale feature learning is adopted to add multi-scale features based on channel-wise weights. Meanwhile, the problem of cross-dataset discrepancies is also addressed on instance normalisation (IN) layers.

In [43], the authors introduce a new architecture based on GNN to overcome the above two studies on re-ID, i.e., group-aware person re-ID and group re-ID. Concretely, the authors formulate a context graph with group members as its nodes to consider dependencies among different people. Also, the authors develop a multi-level attention that can be formed in two group contexts, with an additional self-attention module for robust graph-level representations by attentively aggregating node-level features.

Based on the above-mentioned methods, only the spatial attention is considered in most of the works to extract the distinctiveness features. To learn the discriminative information from the images, we propose a channel-wise attention to learn the representations for person re-ID. In the rest part, we give a detailed description of the proposed method in Section 3.

3. Our Method

In this part, an overview of the proposed architecture is introduced, then we describe the proposed method for person re-ID.

3.1. Framework Overview

The framework for person re-ID is illustrated in Fig. 2. To obtain a robust feature representation, a novel network PGAN, for person re-ID, is designed. In the PGAN framework, we design a deep hybrid convolutional neural network and transformer structure to extract the distinctive features, which consists of a channel-wise attention mechanism, a residual block, and cswin transformer. In the following, a brief introduction of the CNN method is introduced.

3.2. CNN

Recently, the CNN has obtained great attentions in the DL community, which was originally proposed by Fukushima [8] based on “Neocognitron”, and was motivated from Hubel and Wiesel’s hierarchical receptive field of the visual cortex. Subsequently, LeCun

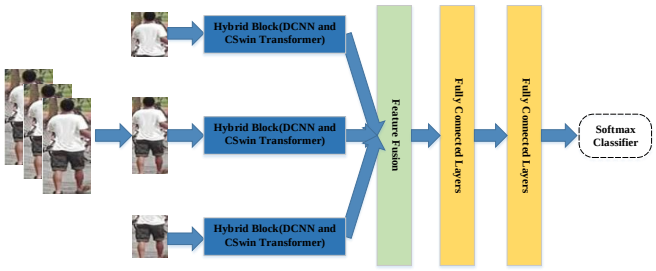


Fig. 2. The proposed deep architecture PGAN, in which Hybrid Block (DCNN and CSwin Transformer) is adopted to extract the deep features from the images and patches.

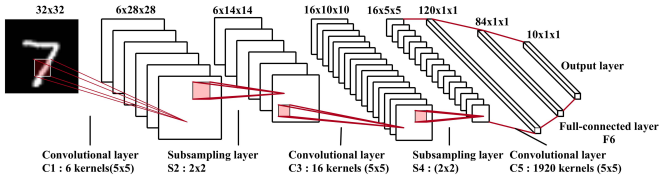


Fig. 3. LeNet-5 network. From [17]. The network contains three types of layers, i.e., convolutional, pooling, and fully-connected layers.

et al. [17] developed a framework for document recognition. In our work, we use LeNet-5 as an example to illustrate the architecture. Three layers is consisted in the LeNet-5: i.e., convolutional, pooling, and fully-connected layers. As illustrated in Fig. 3, we can discover that the convolution layer consists of a series of convolution kernels to compute the feature maps. The pooling layers are adopted to depress the spatial resolution of the feature maps. Commonly, the fully connect layers are used to generate the high-level feature representations [31]. Specifically, all the neurons of the fully-connected layers are used in the previous layer to link each single neuron of the current layer and obtain the output information. In some situations, the fully-connected layers can be modified with a 1×1 convolution layer to obtain the discriminative feature representations [22]. The main point of CNNs is that all the receptive fields share weights in a layer, needing only several parameters compared to fully-connected neural networks. Up to now, numerous CNN architectures have been designed to mitigate studies on image classification studies, i.e., AlexNet [16], ResNet [10], and VGGNet [31], etc.

3.3. Channel-wise Residual Block Based on CNN

The residual block structure of the channel-wise layer attention mechanism is illustrated in Fig. 4, which consists of two stacked basic residual network structures and a channel layer attention mechanism in the output of the last residual network structure. Suppose that the input feature map size is $\frac{C}{2} \times H \times W$ (where C represents the number of feature map channels, and H and W represent the height and width of the feature map, respectively). Batch normalization is performed in the two convolutional layers. The number of feature map channels is doubled in the first convolutional layer of the first residual

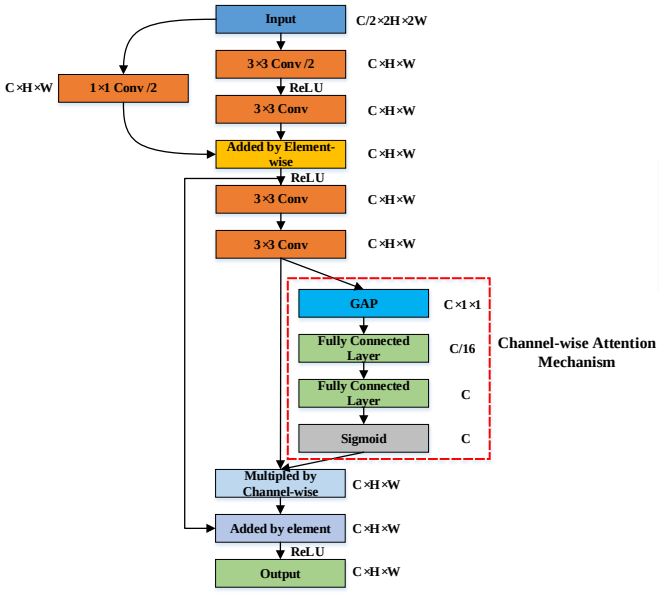


Fig. 4. The illustration of residual block.

network structure, and the feature map size is reduced. To ensure the obtaining of feature maps of the same size, we first use the 1×1 convolutional operation to double the number of the channels with a half number of the feature maps.

Firstly, the output feature map of the last convolutional layer is compressed at the spatial dimension, and the global average pooling is used to change the size of the feature map from the original $C \times H \times W$ to $1 \times 1 \times C$,

$$F_c(X) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} X(i,j) \tag{1}$$

Where X denotes the output feature map of the last convolutional layer.

Then, we learn the relationships among different channels, and the dimension of the channel features is reduced to 16. The ReLU is activated and then the size of the feature channels rises to the original feature channel dimension through a fully-connected layer. Advantages of this procedure include a higher non-linearity, a better fitting of the complex correlations among many channels, and reduction of the number of parameters as well as computations.

For the FC layer F_c, we have the following equation:

$$F_o = \mathrm{Re}LU(F_C \times w + b) \tag{2}$$

Where F_o represents the output of the FC layers, w denotes the weights of the FC layer, b represents the bias of the FC layer.

For the output of the last fully-connected layer, sigmoid activation function is used for nonlinear transform to obtain the normalized weight of each channel within 0-1, and then

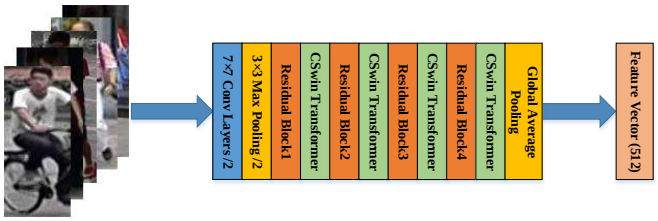


Fig. 5. The main architecture of the combination of Residual Block and Swin Transformer for feature learning.

Table 1. The size of feature maps.

Network	The size of input feature map	The size of output feature map
Conv 7 × 7, stride 2	3×256×256	64×128×128
Max Pooling 3×3, stride 2	64×128×128	64×64×64
Residucal Block 1 + CSwin Transformer	64×64×64	64×64×64
Residucal Block 2 + CSwin Transformer	64×64×64	128×32×32
Residucal Block 3 + CSwin Transformer	128×32×32	256×16×16
Residucal Block 4 + CSwin Transformer	256×16×16	512×8×8
Global Average Pooling	512×8×8	512×1×1

multiply back to the original channel features to complete the recalibration of the original features at the channel dimension. The procedure is written as:

$$\bar{X} = X \times Sigmoid(F_o) \tag{3}$$

Where $\bar{X}$ represents the feature map after recalibration, X denotes the output feature map of the last convolutional layer, and F_o is the output of the last FC layer.

The architecture of the deep convolutional neural network is shown in Fig. 5, which consists of a convolutional layer with kernel size of 7×7, a maximum pooling layer 3×3, residual blocks of four channel layer attention mechanisms and a global average pooling layer, with a final output feature vector of dimensions of 512. To reduce computational quantities and obtain large receptive fields near the input, the feature map size is downsampled twice consecutively using 7×7 convolution and 3×3 maximum pooling layers. Meanwhile, to ensure that not too much information is lost, the feature map size remains unchanged at the first residual block, and in the remaining three residual blocks, the feature map channels are doubled and the height as well as width is halved at the first convolution. For example, with the input feature map dimension of $3 \times 256 \times 256$, the input and output feature map dimension of each layer of the neural network is shown in Table 1.

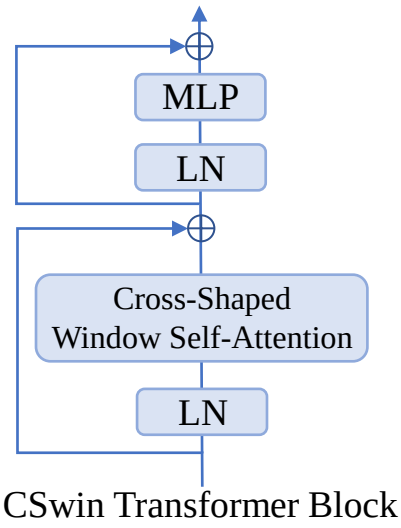


Fig. 6. The main architecture of CSwin Transformer.

3.4. PGAN

The ensemble pipeline of the proposed architecture is illustrated in Fig. 2. Firstly, we crop the images into the upper part of the body and the lower part of the body. Then the whole body and the upper as well as lower part of the body are input into the DCNN architecture to obtain the discriminative feature maps. To make a clear illustration, the three features are denoted as F_w, F_u and F_l, respectively. After input the DCNN, the size of feature maps is 512.

To learn the complimentary information in the three branches, we concatenate the F_w, F_u and F_l to generate the feature vector F_{all},

$$F_{all} = [F_w, F_u, F_l] \tag{4}$$

Finally, we feed the feature vector into the FC layers, and perform the final classification performance.

3.5. Swin Transformer

Swin Transformer is proposed by [24] to model a hierarchical representation by starting from small-sized patches and gradually merging neighboring patches in deeper Transformer layers. As shown in Fig. 6, let assume an input of the size of $h \times w \times c$, Swin Transformer first adapt the input with the size of $\frac{hw}{M^2} \times M^2 \times c$ with window partitioning, where $\frac{hw}{M^2}$ denotes the number of windows (with the size $M \times M$). Then Swin Transformer performs self-attention operation with h times in parallel on each window. Let assume a local window feature $Fea \in \mathbb{R}^{M^2 \times c}$, the Q, K and $V \in \mathbb{R}^{M^2 \times d}$ are calculated by:

$$Q = FeaW_Q, K = FeaW_K, V = FeaW_V \tag{5}$$

Where d is the query/key dimension, W_Q, W_K, W_V represent the $query, keyandvalue$ matrices. The attention matrix $Attention(Q, K, V)$ is calculated by self-attention mechanism of the local window:

$$Attention(Q, K, V) = SoftMax(QK^T/\sqrt{d} + B)V \tag{6}$$

Where B represents the learnable relative positional encoding. After that, a two-layer MLP with GELU activation followed. Then layer norm (LN) layer is added before multi-head self-attention (MSA) and MLP. The detailed description of Swin Transformer, please refer to [24].

Based on Swin Transformer, Dong et al. [6] propose a new transformer named CSwin Transformer as backbone for general-purpose vision tasks. Given an input image with the size of $H \times W \times 3$, we adopt the the overlapped convolutional token embedding (7×7 convolution operatrion with stride 4) to generate $\frac{H}{4} \times \frac{W}{4}$ patch tokens with the dimension C. Based on the self-attention mechanism and positional embedding mechanism, CSwin Transformer block can be written as:

$$\begin{aligned} \hat{Z}^i &= CSWin - Attention(LN(Z^{i-1})) + Z^{i-1}, \\ Z^i &= MLP(LN(\hat{Z}^i)) + \hat{Z}^i \end{aligned} \tag{7}$$

where Z^i represents the output of the $i-$ th CSWin Transformer block.

4. Experiments

4.1. Datasets

To make a fair comparison with the existing works, we adopted three databases to valid our proposed method. To verify the accuracy of the introduced architecture, extensive experiments were performed on the three benchmark database, i.e., Market-1501 [50], DukeMTMC-ReID [29,54] and CUHK03-NP [19,55]. Market-1501 consists of 12,936 images from 751 different identities. Query and gallery set consisted of 3,368 and 19,732 images from another 750 identities. DukeMTMC includes 16,522 data samples from 702 identities, including 2,228 and 17,661 images for the query and gallery set, respectively. CUHK03-NP is a subset of CUHK03, which included two series of data, i.e., labeled and detected images. The detected set of CUHK03 includes 7,365, 1,400 and 5,332 images for the training, query, and gallery partition, respectively. The labeled set of CUHK03 includes for the training, query and gallery partition of the number of 7,368, 1,400 and 5,328, respectively.

4.2. Experimental Settings and Evaluation Standards

4.3. Implementation Details:

To obtain a fast convergence for the deep models, an Adam optimizer with the parameters β_1, β_2 and ε value 0.9, 0.999 and 1×10^{-8} was used. The parameters of Adam had a momentum of 0.9, and a number of epochs of 50. The learning rate of 0.001 is set on the CUHKO-NP and Market-1501 dataset. While a learning rate of 0.0001 is set on the DukeMTMC-ReID dataset. The batch size was set as 32 on 1080Ti GPU. Pytorch platform is used in our experiment. To address the issues of overfitting, an early stop strategy is also adopted. The training process took 24 hours.

4.4. Results

In this part, we describe the performance for person re-ID. The performance of PGAN is introduced, and then discuss the results of person re-ID with those state-of-the-art methods.

Table 2. Performance of PGAN on the three benchmark databases.

Methods	CUHK0-NP				DukeMTMC-ReID		Market-1501	
	Labeled		Detected					
	R-1	mAP	R-1	mAP	R-1	mAP	R-1	mAP
	83.1	81.8	79.4	80.3	92.5	83.4	97.0	91.3

Performance of PGAN The performance of PGAN on Market-1501, DukeMTMC-ReID and CUHK03-NP are illustrated in Table 2. From Table 2, one can see that comparable performances are obtained on three databases. Based on the CUHK0-NP dataset, we obtain the a mAP of 80.2% and 79.1% of the labeled and detected dataset, respectively. For the performance of DukeMTMC-ReID dataset, we obtain a mAP of 82.9% for person re-ID. For the Market-1501 dataset, we obtain the mAP of 90.2% for person re-ID. In our study, we try to use a patch-network and a global-network to extract the discriminative features of person re-ID. The effectiveness of the proposed method is also demonstrated from the videos.

Table 3. Comparison with the state of the art methods on CUHK03-NP.

CUHK03-NP (%)					
Methods	Ref	Labeled		Detected	
		R-1	mAP	R-1	mAP
BoW+XQDA [50]	ICCV15	7.9	7.3	6.4	6.4
SVDNet [33]	ICCV17	–	–	41.5	37.3
DaRe(De)+RE [37]	CVPR18	66.1	61.6	63.3	59.0
MLFN [3]	CVPR18	54.7	49.2	52.8	47.8
HA-CNN [20]	CVPR18	44.4	41.0	41.7	38.6
PCB+RPP [34]	ECCV18	–	–	63.7	57.5
Mancs [33]	ECCV18	69.0	63.9	65.5	60.5
CASN+PCB [52]	CVPR19	73.7	68.0	71.5	64.4
RGA [48]	CVPR20	81.1	77.4	79.6	74.5
NFormer [36]	CVPR22	80.6	79.1	79.0	76.4
Ours		83.1	81.8	79.4	80.3

Comparison with the State-of-the-Art Methods Table 3, 4 and 5 illustrate the comparable accuracy with the state-of-the-art methods for person re-ID. Based on these observations, one can find that comparable results are obtained through our proposed method,

Table 4. Comparison with the state of the art methods on DukeMTMC.

DukeMTMC-ReID (%)			
Methods	Ref	R-1	mAP
BoW+kissme [50]	ICCV15	25.1	12.2
SVDNet [33]	ICCV17	76.7	56.8
DaRe(De)+RE [37]	CVPR18	80.2	64.5
MLFN [3]	CVPR18	81.0	62.8
KPM [30]	CVPR18	80.3	63.2
HA-CNN [20]	CVPR18	80.5	63.8
DNN-CRF [4]	CVPR18	84.9	69.3
PABR [32]	ECCV18	84.4	69.3
PCB+RPP [34]	ECCV18	83.3	69.2
Mancs [33]	ECCV18	84.9	71.8
CASN+PCB [52]	CVPR19	87.7	73.7
PSTA [38]	ICCV21	98.3	97.4
NFormer [36]	CVPR22	90.6	85.7
—- [25]	2023	88.3	75.1
Ours		92.5	83.4

Table 5. Comparison with the state of the art methods on Market-1501.

Market-1501 (%)			
Methods	Ref	R-1	mAP
BoW+kissme [50]	ICCV15	44.4	20.8
SVDNet [33]	ICCV17	82.3	62.1
DaRe(De)+RE [37]	CVPR18	89.0	76.0
MLFN [3]	CVPR18	90.0	74.3
KPM [30]	CVPR18	90.1	75.3
HA-CNN [20]	CVPR18	91.2	75.7
DNN-CRF [4]	CVPR18	93.5	81.6
PABR [32]	ECCV18	91.7	79.6
PCB+RPP [34]	ECCV18	93.8	81.6
Mancs [33]	ECCV18	93.1	82.3
CASN+PCB [52]	CVPR19	94.4	82.8
RGA [48]	CVPR20	96.1	88.4
NFormer [36]	CVPR22	95.7	93.0
PPLR [5]	CVPR22	94.3	84.4
—- [25]	2023	94.8	84.8
Ours		97.0	91.3

showing the capability of our architecture. The reason is that only the deep models are trained from scratch in our research through our proposed method. However, we leverage the attention mechanism to learn the informative patterns from the videos. Using a channel-wise attention mechanism with a patch-network and a global-network, the main features of person re-ID can be learned.

5. Conclusion

In the presented paper, a new framework is introduced, named PGAN, which can disentangle representations can be learned for person re-ID. Of the proposed architecture, a novel attention method named channel-wise attention mechanism is used to learn the informative representations through the patch-network and global network. In addition, CSwin Transformer is also adopted to learn the informative information from the residual blocks. Extensive experiments are performed on the three public datasets, i.e., Market-1501, DukeMTMC-reID and CUHK03-NP, demonstrating the efficiency of the proposed method. Next, we will explore the deep learning methods to extract the deep-learned features and robust models for the person re-ID task.

Acknowledgments. This work was supported by the Science and Technology Project of the State Grid Corporation of China (5700-202019186A-0-0-00), the Shaanxi Provincial Natural Science Foundation (grant 2022JQ-718), and the National Statistical Research Project (grant 2021LY037).

References

1. Bai, Y., Jiao, J., Ce, W., Liu, J., Lou, Y., Feng, X., Duan, L.Y.: Person30k: A dual-meta generalization network for person re-identification. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2123–2132 (2021)
2. Bai, Z., Wang, Z., Wang, J., Hu, D., Ding, E.: Unsupervised multi-source domain adaptation for person re-identification. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12914–12923 (2021)
3. Chang, X., Hospedales, T.M., Xiang, T.: Multi-level factorisation net for person re-identification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2109–2118 (2018)
4. Chen, D., Xu, D., Li, H., Sebe, N., Wang, X.: Group consistent similarity learning via deep crf for person re-identification. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8649–8658 (2018)
5. Cho, Y., Kim, W.J., Hong, S., Yoon, S.E.: Part-based pseudo label refinement for unsupervised person re-identification. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7308–7318 (2022)
6. Dong, X., Bao, J., Chen, D., Zhang, W., Yu, N., Yuan, L., Chen, D., Guo, B.: Cswin transformer: A general vision transformer backbone with cross-shaped windows. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12124–12134 (2022)
7. Eom, C., Lee, W., Lee, G., Ham, B.: Is-gan: Learning disentangled representation for robust person re-identification. IEEE transactions on pattern analysis and machine intelligence (2021)
8. Fukushima, K., Miyake, S.: Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In: Competition and cooperation in neural nets, pp. 267–285. Springer (1982)
9. Geng, Y., Hu, H.M., Zeng, G., Zheng, J.: A person re-identification algorithm by exploiting region-based feature salience. Journal of Visual Communication and Image Representation 29, 89–102 (2015)
10. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
11. Hou, R., Chang, H., Ma, B., Huang, R., Shan, S.: Bicnet-tks: Learning efficient spatial-temporal representation for video person re-identification. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2014–2023 (2021)

12. Hou, R., Ma, B., Chang, H., Gu, X., Shan, S., Chen, X.: Feature completion for occluded person re-identification. IEEE Transactions on Pattern Analysis and Machine Intelligence (2021)
13. Hu, H.M., Fang, W., Zeng, G., Hu, Z., Li, B.: A person re-identification algorithm based on pyramid color topology feature. Multimedia Tools and Applications 76(24), 26633–26646 (2017)
14. Isobe, T., Li, D., Tian, L., Chen, W., Shan, Y., Wang, S.: Towards discriminative representation learning for unsupervised person re-identification. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8526–8536 (2021)
15. Kalayeh, M.M., Basaran, E., Gökmen, M., Kamasak, M.E., Shah, M.: Human semantic parsing for person re-identification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1062–1071 (2018)
16. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems 25, 1097–1105 (2012)
17. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE 86(11), 2278–2324 (1998)
18. Li, S., Bak, S., Carr, P., Wang, X.: Diversity regularized spatiotemporal attention for video-based person re-identification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 369–378 (2018)
19. Li, W., Zhao, R., Xiao, T., Wang, X.: Deepreid: Deep filter pairing neural network for person re-identification. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 152–159 (2014)
20. Li, W., Zhu, X., Gong, S.: Harmonious attention network for person re-identification. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2285–2294 (2018)
21. Liao, S., Hu, Y., Zhu, X., Li, S.Z.: Person re-identification by local maximal occurrence representation and metric learning. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2197–2206 (2015)
22. Lin, M., Chen, Q., Yan, S.: Network in network. arXiv preprint arXiv:1312.4400 (2013)
23. Liu, X., Zhao, H., Tian, M., Sheng, L., Shao, J., Yi, S., Yan, J., Wang, X.: Hydraplus-net: Attentive deep features for pedestrian analysis. In: Proceedings of the IEEE international conference on computer vision. pp. 350–359 (2017)
24. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10012–10022 (2021)
25. Ma, X., Lv, W., Zhao, M.: A double stream person re-identification method based on attention mechanism and multi-scale feature fusion. IEEE Access 11, 14612–14620 (2023)
26. Matsukawa, T., Okabe, T., Suzuki, E., Sato, Y.: Hierarchical gaussian descriptor for person re-identification. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1363–1372 (2016)
27. Meng, J., Zheng, W.S., Lai, J.H., Wang, L.: Deep graph metric learning for weakly supervised person re-identification. IEEE Transactions on Pattern Analysis and Machine Intelligence (2021)
28. Rao, H., Wang, S., Hu, X., Tan, M., Guo, Y., Cheng, J., Liu, X., Hu, B.: A self-supervised gait encoding approach with locality-awareness for 3d skeleton based person re-identification. IEEE Transactions on Pattern Analysis and Machine Intelligence (2021)
29. Ristani, E., Solera, F., Zou, R., Cucchiara, R., Tomasi, C.: Performance measures and a data set for multi-target, multi-camera tracking. In: European conference on computer vision. pp. 17–35. Springer (2016)
30. Shen, Y., Xiao, T., Li, H., Yi, S., Wang, X.: End-to-end deep kronecker-product matching for person re-identification. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 6886–6895 (2018)

31. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. Computer Science (2014)
32. Suh, Y., Wang, J., Tang, S., Mei, T., Lee, K.M.: Part-aligned bilinear representations for person re-identification. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 402–419 (2018)
33. Sun, Y., Zheng, L., Deng, W., Wang, S.: Svdnet for pedestrian retrieval. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 3800–3808 (2017)
34. Sun, Y., Zheng, L., Yang, Y., Tian, Q., Wang, S.: Beyond part models: Person retrieval with refined part pooling (and a strong convolutional baseline). In: Proceedings of the European conference on computer vision (ECCV). pp. 480–496 (2018)
35. Varior, R.R., Haloi, M., Wang, G.: Gated siamese convolutional neural network architecture for human re-identification. In: European conference on computer vision. pp. 791–808. Springer (2016)
36. Wang, H., Shen, J., Liu, Y., Gao, Y., Gavves, E.: Nformer: Robust person re-identification with neighbor transformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7297–7307 (2022)
37. Wang, Y., Wang, L., You, Y., Zou, X., Chen, V., Li, S., Huang, G., Hariharan, B., Weinberger, K.Q.: Resource aware person re-identification across multiple resolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8042–8051 (2018)
38. Wang, Y., Zhang, P., Gao, S., Geng, X., Lu, H., Wang, D.: Pyramid spatial-temporal aggregation for video-based person re-identification. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 12026–12035 (October 2021)
39. Wei, L., Zhang, S., Gao, W., Tian, Q.: Person transfer gan to bridge domain gap for person re-identification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 79–88 (2018)
40. Wu, J., Yang, Y., Lei, Z., Wang, J., Li, S.Z., Tiwari, P., Pandey, H.M.: An end-to-end exemplar association for unsupervised person re-identification. Neural Networks 129, 43–54 (2020)
41. Xu, W., Liu, H., Shi, W., Miao, Z., Lu, Z., Chen, F.: Adversarial feature disentanglement for long-term person re-identification. In: Proceedings of the International Joint Conference on Artificial Intelligence (2021)
42. Yan, C., Pang, G., Bai, X., Liu, C., Xin, N., Gu, L., Zhou, J.: Beyond triplet loss: person re-identification with fine-grained difference-aware pairwise loss. IEEE Transactions on Multimedia (2021)
43. Yan, Y., Qin, J., Ni, B., Chen, J., Liu, L., Zhu, F., Zheng, W.S., Yang, X., Shao, L.: Learning multi-attention context graph for group-based re-identification. IEEE Transactions on Pattern Analysis and Machine Intelligence (2020)
44. Yang, Y., Tan, Z., Tiwari, P., Pandey, H.M., Wan, J., Lei, Z., Guo, G., Li, S.Z.: Cascaded split-and-aggregate learning with feature recombination for pedestrian attribute recognition. International Journal of Computer Vision pp. 1–14 (2021)
45. Yang, Y., Tiwari, P., Pandey, H.M., Lei, Z., et al.: Pixel and feature transfer fusion for unsupervised cross-dataset person reidentification. IEEE Transactions on Neural Networks and Learning Systems (2021)
46. Yang, Y., Zhang, T., Cheng, J., Hou, Z., Tiwari, P., Pandey, H.M., et al.: Cross-modality paired-images generation and augmentation for rgb-infrared person re-identification. Neural Networks 128, 294–304 (2020)
47. Yuan, L., Tian, Z.: Person re-identification based on color and texture feature fusion. In: International conference on intelligent computing. pp. 341–352. Springer (2016)
48. Zhang, Z., Lan, C., Zeng, W., Jin, X., Chen, Z.: Relation-aware global attention for person re-identification. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2020)

49. Zhao, H., Tian, M., Sun, S., Shao, J., Yan, J., Yi, S., Wang, X., Tang, X.: Spindle net: Person re-identification with human body region guided feature decomposition and fusion. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1077–1085 (2017)
50. Zheng, L., Shen, L., Tian, L., Wang, S., Wang, J., Tian, Q.: Scalable person re-identification: A benchmark. In: Proceedings of the IEEE international conference on computer vision. pp. 1116–1124 (2015)
51. Zheng, L., Yang, Y., Hauptmann, A.G.: Person re-identification: Past, present and future. arXiv preprint arXiv:1610.02984 (2016)
52. Zheng, M., Karanam, S., Wu, Z., Radke, R.J.: Re-identification with consistent attentive siamese networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5735–5744 (2019)
53. Zheng, W.S., Gong, S., Xiang, T.: Reidentification by relative distance comparison. IEEE transactions on pattern analysis and machine intelligence 35(3), 653–668 (2012)
54. Zheng, Z., Zheng, L., Yang, Y.: Unlabeled samples generated by gan improve the person re-identification baseline in vitro. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 3754–3762 (2017)
55. Zhong, Z., Zheng, L., Cao, D., Li, S.: Re-ranking person re-identification with k-reciprocal encoding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1318–1327 (2017)
56. Zhou, K., Yang, Y., Cavallaro, A., Xiang, T.: Learning generalisable omni-scale representations for person re-identification. IEEE Transactions on Pattern Analysis and Machine Intelligence (2021)

Liu Jing, Female, PhD in Engineering, professor, works in Weinan Normal University. Her research interests include Computational photography and Light field imaging and processing.

Zhou Guoqing, Male, PhD in Engineering, associate professor, master's supervisor, and visiting scholar at Johns Hopkins University (JHU). His research directions include theories and issues in international frontier fields such as machine vision, computational photography, and global optimization theory.

Received: August 29, 2023; Accepted: February 07, 2024.

Computer Science and Information Systems 21(4):1499–1522 https://doi.org/10.2298/CSIS230907045J

Semantic Feature-Based Test Selection for Deep Neural Networks: A Frequency Domain Perspective

Zhouxian Jiang[1], Honghui Li[1], Xuetao Tian[2], and Rui Wang[1,⋆]

[1] School of Computing and Information Technology, Beijing Jiaotong University, Beijing, China
zhouxianjiang@bjtu.edu.cn
hhli@bjtu.edu.cn
rui.wang@bjtu.edu.cn
[2] Faculty of Psychology, Beijing Normal University, Beijing, China
xttian@bnu.edu.cn

Abstract. While deep neural networks (DNNs) have great potential for applications in security and safety-critical domains, their limited robustness to adversarial samples and out-of-distribution (OOD) samples raise significant concerns. In the software engineering community, significant efforts have been devoted to devising testing techniques that verify the robustness of DNNs. This paper investigates semantic feature-based test selection for DNNs from a frequency domain perspective and propose a novel method called SaFeTS. Specifically, we leverage saliency detection techniques, such as Fourier Phase Transform to extract semantic features from test cases. These features are then clustered to select diverse test cases to evaluate the robustness of DNNs and model retraining. Experiments on CIFAR-10 and SVHN datasets demonstrate that SaFeTS exposes more varied model errors compared to baseline methods. Further, retraining with SaFeTS-selected samples significantly improves adversarial and out-of-distribution robustness over state-of-the-art test selection methods.

Keywords: DNN testing, test selection, semantic feature, frequency domain, robustness.

1. Introduction

Deep learning (DL) systems are susceptible to adversarial samples [48] by malicious attacks, as well as out-of-distribution (OOD) samples [1] originating from natural environments. These vulnerabilities present potential security and safety threats, raising concerns about the application of DL technologies in security and safety-critical areas [5, 8, 38, 43, 44]. Researchers in the software engineering community have focused on verifying and improving the robustness of DL systems through testing to mitigate these threats. However, a pivotal element of a DL system is the deep neural network (DNN), which fundamentally differs from conventional software in several vital aspects: programming paradigms, source of faults, and repair approach. Conventional software is often developed based on specific requirements, while DNNs predominantly operate within a data-driven programming paradigm, making decisions through non-deterministic logic

⋆ Corresponding Author

[3]. Therefore, the faults in conventional software often stem from specific functions or lines of code. In contrast, the faults in DNNs typically manifest as deviations in model behavior, often linked to the data, learning algorithm, or training process. In addition, addressing faults in conventional software often involves pinpointing and correcting faulty code [9]. However, repairing faults in DNNs is more intricate, as they cannot be localized to a specific component (e.g., neuron or layer in the model) for direct fixing. A common approach to enhance the robustness of DNN is to expand the data set to retrain the model.

Early works have proposed structural adequacy criteria for DNN testing [35, 40], as well as test methods based on these criteria [17, 36, 40, 46, 56]. Recent efforts have extended test prioritization or selection methods to reduce computational and labeling costs. These methods are based primarily on specific rules, such as uncertainty [14, 55] and gradient-based rules [53]. However, existing test selection methods often focus on boosting the quantity of the model errors disclosed during the test rather than the diversity of misbehavior [15]. This leads to inadequate coverage of model erroneous behavior that potentially exists in real-world scenarios. Additionally, some work shows that existing test selection methods have a minimal impact on improving DNN's robustness [34, 53].

To mitigate these issues, we explore an alternative strategy to enhance the test selection for DNNs. Given the data-driven nature of DNNs, we aim to increase the diversity of test cases, not only due to its results in software testing [6, 13, 21, 32], but also due to the call for many work in DNN testing [15,31,61,62]. Essentially, relying on diverse test cases should expend the exploration scale of sample space and thus improve fault detection rate for a given test set [2]. Further, incorporating diverse samples in the retraining/repair process helps the model to learn diverse data representations, thereby improving the model's generalization ability and robustness on unknown data.

Current methods to assess the diversity of test cases often adopt distance measures, which overlooks semantic diversity. Here, the *Semantics* refers to the meaning of data, which is the essential attribute used to distinguish the differences among data. Instead, Chen et al. [11] demonstrate that the depth at which a model understands or learns from the data's semantics is strongly linked to its generalization capabilities. In turn, we can exploit the semantic diversity of the data for testing to reveal more model faults. Additionally, researchers [11, 18, 41, 60] found that DNNs lack robustness because they over-fitted to noisy high-frequency information and ignored semantic information during training. Suppressing high-frequency noise in the data will highlight the semantics of the data. Thus, in this paper, we utilize the frequency domain transform method to extract the semantics of the data and design the test selection method to improve the diversity of the selected data. The noise of the data is usually added in the spatial domain and can be filtered out by some frequency domain methods. For example, in the field of image processing, frequency domain based methods can help remove unimportant high frequency components that are often invisible to the human eye but contain noise [52]. Frequency domain analysis can further enhance features in certain frequency ranges, making them easier to recognize in subsequent processing. Thus, frequency domain-based methods can extract features that are semantically consistent with human intuition and avoid the impact of data noise [11]. Besides, frequency domain-based methods leverage the prior knowledge of natural images, and therefore are model-agnostic [50]. This characteristic helps circumvent the biases introduced by neural network feature extractors.

In this paper, we propose a test selection method for DNNs from a frequency domain perspective. We empirically investigated the effectiveness of four commonly used frequency domain-based saliency detection methods in extracting semantic features. The evaluation results demonstrate that these methods can effectively extract semantic features even from perturbed data. Motivated by this observation, we proposed a novel Semantic Feature-based Test Selection method (SaFeTS). Specifically, we extract the semantic features of test cases using frequency domain-based saliency detection methods. Subsequently, we leverage these semantic features to identify potentially diverse categories of samples. Then, we select samples from each category for testing and retraining purposes. Our work innovatively introduces the frequency domain analysis method into the test selection technique, focuses on the diversity of semantic information of test cases, and provides a new way of thinking for DNN testing. Extensive experimental results validate the effectiveness of the proposed SaFeTS. Compared to the baseline methods, we disclose more diverse DNN misbehavior with the proposed method. Moreover, SaFeTS can effectively improve the adversarial and OOD robustness of DNN via retraining. The main contributions of this paper are as follows.

• We explored test selection for DNNs from a frequency domain perspective. Through a comprehensive empirical study, we observed that the frequency domain-based saliency detection methods can effectively extract semantic features consistent with human intuition, even for perturbed data.

• We proposed a novel semantic feature-based test selection method SaFeTS. SaFeTS leverages saliency detection methods: Fourier Phase Transform (FPT) [39], High-pass Filter (HPF) [27], Residual Spectrum (SR) [23], and Quaternion Fourier Transform (QFT) [16], to extract semantic features from test cases and identify diverse categories of samples based on these features. Then, we select samples from each category for testing and retraining purposes.

• We conducted extensive experimental evaluations of SaFeTS. Compared to baselines, SaFeTS can detect more diverse DNN misbehavior. By selecting samples with diverse semantics for retraining, SaFeTS can effectively improve the adversarial and OOD robustness of DNNs. In particular, SaFeTS utilizes the FPT to extract semantic features, resulting in an average performance improvement of 20.79% over the best-performing baseline on adversarial samples. We make our code public at GitHub.

The rest of this paper is organized as follows. We introduce the related work in Section 2. In Section 3, we outline the empirical study to validate the efficacy of frequency domain-based methods. In Section 4, we present our proposed method, SaFeTS. Finally, we demonstrate the evaluation results and summarize this work in Section 5 and 8, respectively.

2. Related Work

This section presents an overview of the related work, encompassing test selection for DNNs, alongside the frequency domain analysis applied to image processing.

2.1. Testing for DNN

Researchers in the software engineering community have designed testing methods for DNN to mitigate security and safety threats to DL-based systems. Inspired by structural

coverage criteria in conventional software testing, Pei et al. [40] first proposed the neuron coverage criteria for DNN testing and a white-box testing framework DeepXplore. Subsequently, Ma et al. [35] extended their work by offering multi-granularity testing criteria. Based on these test criteria, various testing methods have been proposed to generate test cases to discover the misbehavior of DNNs and verify their robustness, such as differential testing [40], fuzzing testing [17], and mutation testing [36], and symbolic testing [46, 47] and metamorphic testing [49, 56].

Due to the data-driven nature of DNN, comprehensive model testing requires large-scale test cases, resulting in a vast testing overhead. Therefore, some researchers investigated test prioritization or selection techniques to reduce the cost of the testing. Feng et al. [14] proposed DeepGini, a test prioritization method, to select test cases according to the confidence level calculated by the Gini index. Wang et al. [53] introduced a robustness-oriented testing framework for DNN, which used gradient-based rules as test prioritization. Inspired by active learning, Weiss et al. [55] used the uncertainty metrics as test prioritization. In addition, some test selection methods based on mutation testing [54] and differential testing [15] were proposed.

Current test methods focus on selecting the test cases that are more prone to trigger model misbehavior, therefore to boost the quantity of the identified software errors [15]. However, Zhang et al. [61] pointed out that using the error number as an evaluation metric to measure testing effectiveness is inadequate for DNNs. It is worth taking efforts to generate or select test cases with more diversity to expose various model incorrect behavior that potentially exists in real-world scenarios. Additionally, the test cases generated or selected by current test approaches have limited effect on improving the robustness of the DL system after testing [53]. These limitations call for improved test selection techniques to increase the diversity of selected test cases and further enhance the robustness of DNNs. To address the above limitations, we propose SaFeTS for test selection, aiming at selecting more semantically diverse test cases. These test cases with semantic diversity can expose diverse DNN misbehavior during testing and improve the robustness of DNNs after retraining.

2.2. Frequency Domain Analysis

Frequency domain analysis is one of the core techniques in the field of signal processing, which mainly involves the study of the frequency components of signals [7]. There are a wide range of techniques and applications of frequency domain analysis from time sequence data to images. The Fourier transform is the basis of frequency domain analysis, which converts a signal in the time domain or spatial domain to its frequency domain representation. For images, a two-dimensional FT can be used to obtain its frequency domain representation.

Frequency domain analysis has been widely used in conventional image processing, especially for saliency detection [16, 23, 27, 33, 39]. These methods often convert spatial-domain images to frequency-domain information through the Fourier transform. They are often used to extract the semantic features of images. Although DL-based feature extractors have been well developed, frequency domain-based methods still play a vital role in improving the robustness of DNNs [11]. Recently, some work investigated the robustness generalization behaviors of DNN from a data frequency domain perspective [11, 24, 45, 51, 57, 59]. In particular, Wang et al. [51] empirically found that the invis-

ible noises in high-frequency components of input images lead DNNs to make counter-intuitive predictions. Chen et al. [11] argued that a robust DNN should focus on learning the semantic information of data rather than high-frequency noise. Hence, they trained the models on the augmented data by combining the images' phase and amplitude spectrum. Frequency domain analysis methods can separate different levels of information, such as high frequency and low frequency, phase and amplitude. The frequency domain based saliency detection method utilizes the information of different levels and can effectively extract the semantic features of the data to avoid interference from data noise.

Thus, motivated by the strong ability of the frequency domain-based saliency detection methods, we explore the effective test selection for DNNs from a frequency domain perspective in this paper. We utilize these saliency detection methods to extract data semantic features. Furthermore, these semantic features provide valuable guidance in selecting diverse and representative test cases for DNN testing and retraining.

3. Empirical Study

In this section, we conducted an empirical study to validate the effectiveness of frequency domain-based saliency detection methods involving four classic saliency detection methods. We first briefly introduce the fundamentals of saliency detection methods. Then, we applied different perturbation techniques on test images of ImageNet [12], including adversarial attack and corruption techniques, to create adversarial and OOD samples, respectively. Finally, we evaluate the saliency detection methods by comparing the semantic maps.

3.1. Frequency Domain-based Semantic Feature Extraction

In image processing, the Discrete Fourier Transform (DFT) is commonly used to convert images from the spatial domain to the frequency domain. Research has found that frequency domain information can reveal semantic features of data, which are consistent with human perception (e.g., structures, edges, and textures) [27, 39]. In the empirical study, we investigated four classic frequency domain-based saliency detection methods: FPT [39], HPF [27], SR [23] and QFT [16], which have different characteristics in image semantic feature detection.

• **FPT** [39]: When an image is transformed using the Fourier transform, it is divided into frequency components consisting of amplitude and phase. The phase spectrum is a representation of the phase information in the frequency domain of an image. Reflects the relative positional relationship between different frequency components in the image, which is often adopted to detect structural and textural changes in the image.

• **HPF** [27]: A high-pass filter aims to sharpen edges or highlight fine details by attenuating low-frequency information. It typically corresponds to the smooth regions or large-scale structures in the image, which helps to highlight the edges and contours of objects.

• **SR** [23]: The spectrum residual is based on the observation that the human visual system tends to suppress the response to frequently occurring features, while at the same time keeping sensitive to features that deviate from the norm [29]. The spectrum residual method takes the logarithmic spectrum of an input image, followed by a Gaussian blur

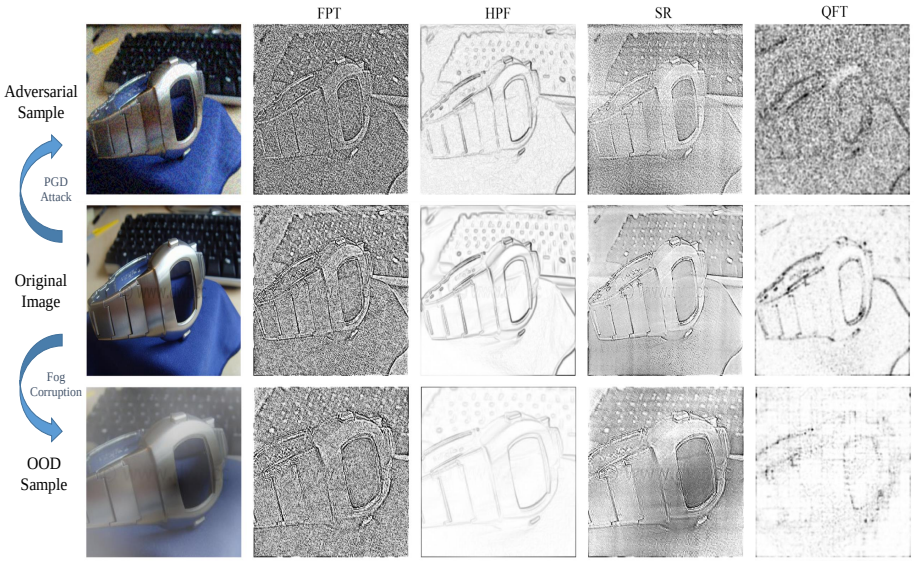


Fig. 1. Frequency domain-based semantic features of the images under different saliency detection methods. The columns from left to right are the original spatial-domain images, and the semantic maps extracted by FPT, HPF, SR, and QFT, respectively

operation. Then, the inverse Fourier transform is applied to the residual of the blurred and original log-spectrum, to obtain the saliency map.

• **QFT** [16]: QFT is an extension of the traditional Fourier transform, designed to work with quaternion-valued functions or data. QFT allows analyzing and representing color images and other multicomponent data using quaternion mathematics, simultaneously capturing an image's spatial and color information.

We applied the aforementioned methods on 1,000 images from dataset ImageNet [12]. Due to the limited space of the paper, we present one example of the images in Fig. 1. The remaining results are shown at GitHub. We observed from the second row in Fig. 1 that all four saliency detection methods preserved the semantic information of the image, such as the shape and contour of the watch. We re-verified that the extracted frequency domain-based semantic features provide a representation consistent with human perception. The experimental results confirm the feasibility of analyzing test samples from the frequency domain perspective.

3.2. Semantic Feature Extraction on Perturbed Images

Our further investigations probed whether four saliency detection methods could preserve semantic feature extraction performance on perturbed data. We utilized a pre-trained VGG16 [42] model on ImageNet [12] to generate adversarial samples by applying a gradient-based adversarial attack technique, PGD [37] and CW [10]. For crafting OOD samples, we adopted a widely used corrupted data benchmark [22], which encompasses

19 types of corruption involving weather, blur, noise, and digital changes with damage levels ranging from 1 to 5.

The first and third rows of Fig. 1 show the semantic maps on adversarial and OOD samples, respectively. Due to the space limitation, we only offer the cases of perturbations caused by PGD attacks and fog corruption. More case studies are shown at GitHub. Although both PGD attacks and fog corruption may diminish the semantic clarity of the original sample, all four methods still managed to extract semantic information from adversarial and OOD samples. In particular, FPT outperforms the other three saliency detection techniques, while QFT exhibits noticeable degradation in semantic maps.

In summary, in this section, we investigate the changes in semantic maps of adversarial and OOD samples. We found that saliency detection methods can effectively extract semantic information from perturbed samples. Inspired by these observations, we design a frequency domain-based test selection method to increase the semantic diversity of test cases for DNN model testing and robustness enhancement via retraining.

4. Approaches

In this section, we propose the SaFeTS test selection method to detect more diverse model erroneous behaviors and improve robustness via model retraining. First, we introduce the problem definition in deep learning. Then, we present the SaFeTS method in detail.

4.1. Problem Definition

The *Robustness* of deep learning model often consists of *Global Robustness* and *Empirical Robustness* [53]. Global robustness refers to the robustness of a deep learning model to disturbances or noise throughout the input space. Empirical robustness is evaluated by testing the model on specific perturbed datasets. In this paper, we study the robustness of deep learning restricted to empirical robustness, involving evaluating adversarial and OOD data. The empirical robustness is defined as follows.

Definition 1. (Empirical Robustness) Given a DNN model $\mathcal{F} : X \rightarrow Y$ and a perturbed test dataset D_t, where X and Y represent the input and output of $\mathcal{F}$, respectively. Empirical robustness is defined as $\mu : (\mathcal{F}, D_t) \rightarrow \gamma$, and $\gamma \in [0, 1]$ is the accuracy of model $\mathcal{F}$ on the test set D_t.

4.2. Semantic Feature-Based Test Selection

In this section, we introduce the proposed test selection method SaFeTS. The critical issue in test selection is how to utilize the semantics of test cases to expose the diverse misbehavior of DNN and use them to improve DNN's robustness. We design a frequency domain-based test selection method for DNNs. Fig. 2 shows the workflow of SaFeTS, and Algorithm 1 details the process of test selection based on SaFeTS. In general, SaFeTS selects test cases based on a given candidate test set through two steps: (1) semantic feature extraction and (2) clustering and sampling. Specifically, we use frequency domain-based saliency detection methods to extract the semantic features of test cases (line 4). Such methods are model-agnostic, which do not incorporate feature extractor bias. They extract semantic information while filtering out imperceptible high-frequency noise. After

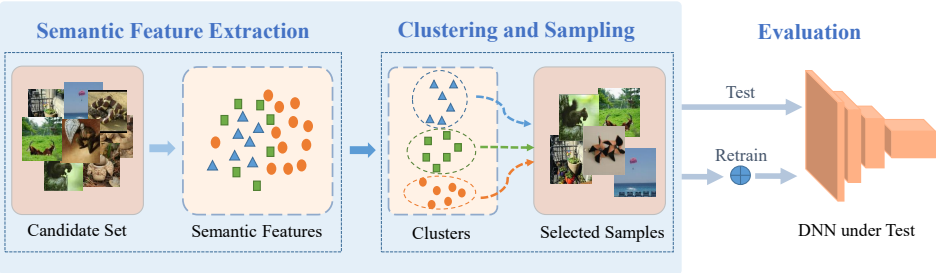


Fig. 2. The workflow of SaFeTS. SaFeTS includes two components: semantic feature extraction, clustering, and sampling. We evaluate the proposed method by testing and retraining DNN models on the selected samples

that, we adopt the clustering method to mine the semantically diverse categories of samples based on the extracted frequency-domain features (line 8). Finally, we sample from each cluster (lines 9-16), and use the sampled test cases to test and retrain DNNs.

Algorithm 1 Semantic Feature-Based Test Selection (SaFeTS)

Require: candidate set X, sampling ratio r, semantic feature set S_x, indices of sampled test cases $indicies$.
Ensure: selected set X_s.
1: $X_s \leftarrow \emptyset$, $S_x \leftarrow \emptyset$, $indices \leftarrow \emptyset$
2: **for** each $x \in X$ **do**
3: # Extract the semantic features based on frequency domain transformation methods.
4: $s \leftarrow Semantic_extraction(x)$
5: S_x.add(s)
6: **end for**
7: # Cluster semantic feature to C categories.
8: $C \leftarrow Cluster_semantic(S_x)$
9: **for** each $c \in C$ **do**
10: # Uniform sampling with ratio r from each cluster, get the indices of sampled test cases.
11: $indices \leftarrow indices \cup Uniform_Sampling(c, r)$
12: **end for**
13: **for** each $i \in indicies$ **do**
14: # Select the samples from the candidate set.
15: $X_s \leftarrow X_s \cup X[i]$
16: **end for**
17: **return** X_s

Semantic feature extraction To extract the semantic features of the test cases, we transform the input image from the spatial domain to the frequency domain using a Fourier transform. Given an image g matrix with M rows and N columns, $f(m, n)$ denotes the pixel value of the m_{th} row and n_{th} column, the Fourier transform $F(u, v)$ of $f(m, n)$ is

shown in Equation (1).

$$F(u,v) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n) \cdot e^{-j2\pi(um/M+vn/N)} \tag{1}$$

For an image g matrix with M rows and N columns, the Fourier transform theoretically requires $(M * N)^2$ operations, which is very time-consuming. Alternatively, we use the Fast Fourier Transform (FFT) to reduce the complexity from $O((M * N)^2)$ to $O(MNlog(M * N))$ by decomposing the DFT into a series of smaller DFTs. In contrast, we can transform an input image from its frequency domain to spatial domain via Inverse Fast Fourier transform (IFFT), shown in Equation (2).

$$f(m,n) = \frac{1}{\sqrt{MN}} \sum_{v=0}^{M-1} \sum_{u=0}^{N-1} F(u,v) \cdot e^{2\pi((vm/M)+(un/N))} \tag{2}$$

Based on FFT and IFFT, we employ four classic saliency detection methods mentioned in Section 3 to extract the semantic information of the test cases. The details of the calculations are as follows:

• **FPT:** The phase spectrum of an image is the phase portion of the Fourier transform result, shown in Equation (3).

$$P(u,v) = \arctan\left(\frac{Im[F(u,v)]}{Re[F(u,v)]}\right) \tag{3}$$

Where Im$[F(u,v)]$ and Re$[F(u,v)]$ are the imaginary and real parts of $F(u,v)$, respectively. Then, the IFFT of P leads to a saliency map $S_a(m,n)$.

• **HPF:** The HPF are mainly used to enhance the high-frequency components of an image so as to highlight the edges and details. In this paper, we use Sobel filter to compute HPF which is a discrete difference filter used for edge detection to highlight the edges by computing the gradient strength of the image. Sobel filter contains two $3 * 3$ matrices which is shown in Equation (4) for detecting the edges in horizontal G_x and vertical direction G_y, respectively.

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \tag{4}$$

For each pixel in the image, its gradient values in the horizontal V_x and vertical directions V_y are calculated by the convolution operation in Equation (5).

$$V_x(m,n) = f(m,n) * G_x, V_y(m,n) = f(m,n) * G_y \tag{5}$$

Then calculate the total gradient intensity, i.e., saliency map Sa, by Equation (6).

$$S_a(m,n) = \sqrt{V_x(m,n)^2 + V_y(m,n)^2} \tag{6}$$

After processing, a new image is obtained which emphasizes the edge regions in the original image and can be considered as the result of HPF process.

• **SR:** The SR method estimates significance from the residuals of the amplitude spectrum of the image. The amplitude spectrum $A(u,v) = |F(u,v)|$ and the phase spectrum $P(u,v)$ are computed after performing FFT on a given image. Then, calculate the average amplitude spectrum $\log(A(u,v))$ and its smoothed version $Sm(u,v)$. spectrum residual is computed by $R(u,v) = \log(A(u,v)) - Sm(u,v)$. The Fourier transform of the saliency map is $S_a(u,v) = e^{R(u,v)} \times e^{jP(u,v)}$. Finally, the IFFT is used to obtain the saliency map $S_a(m,n)$.

• **QFT:** QFT-based saliency detection suggests that in the human visual system, the four feature channels, motion feature M, luminance feature I, red-green neuron RG, and blue-yellow neuron BY, are almost independent. Therefore, the above four features of an image can be represented by a quaternion $q(m,n)$, defining the weighted image quaternion representation as follow:

$$q(m,n) = M(m,n) + RG(m,n) * \mu_1 + BY(m,n) * \mu_2 + I(m,n) * \mu_3 \tag{7}$$

where $\mu_i, i = 1,2,3$ satisfies $\mu_i^2 = -1$, $\mu_1 \perp \mu_2$, $\mu_2 \perp \mu_3$, $\mu_1 \perp \mu_3$, $\mu_3 = \mu_1\mu_2$. The QFT transforms an image from the spatial domain to the frequency domain as follows:

$$Q(u,v) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} q(m,n) \cdot e^{-j2\pi(um/M + vn/N)} \tag{8}$$

Here we actually extend the concept of 2D FFT to accommodate quaternionic data. Then, in the frequency domain, saliency detection is performed based on the phase information of Q (u, v), that is, $S_a(u,v) = |Q(u,v)|$. And finally, the inverse QFT of $S_a(u,v)$ leads to a saliency map $S_a(m,n)$ (extensive the concept of 2D IFFT to accommodate quaternionic data).

Clustering and sampling To group test cases into multiple categories with similar semantic features, we incorporate clustering approach in SaFeTS. In this paper, we extract the semantic feature of image samples and cluster them into various categories in the frequency domain feature space. Due to the convergence speed and cluster performance, we employ the Kmeans++ algorithm [4] as the clustering method based on semantic features to divide all samples into k clusters, where k is the number of groups. Kmeans++ algorithm is an optimized version of the Kmeans clustering algorithm [19], where the initial centroids are selected through a specific randomization process to reduce the effect of randomization. It randomly selects the first centroid and then uses weighted random sampling to choose the remaining centroids.

After that, we adopted the Uniform Sampling (US) strategy to maximize the diversity of the sampled test cases. The US strategy ensures that the samples selected from each cluster are uniformly distributed, which helps to increase the diversity and coverage of the samples. In addition, since the uniform sampling strategy does not bias the selection of samples towards any particular region, it provides a wider and diverse set of test cases, which is important for deep neural network testing. In summary, the Kmeans++ and US strategies can be easily scaled or adapted to different datasets and application scenarios, and provide an efficient, high-quality and low-bias strategy for DNN testing that ensures sample diversity and broad coverage.

Retraining objective The samples selected based on SaFeTS can be used not only to test the DNN but also to enhance the model robustness via retraining. Training samples consist of the original training set D and the sample selected from the perturbed test set D_t based on the test selection method TS. We define the optimization objective for robustness retraining as

$$\min_{\theta} \frac{1}{L} \sum_{i=1}^{L} J_{(x_i,y_i)\in D\cup TS(D_t)}(\theta, x_i, y_i), \tag{9}$$

where L is the number of training samples, x, y is the training samples and their labels, and θ are the parameters of the model $\mathcal{F}$. Retraining aims to optimize the model parameters θ that lead to minimization of the loss function J.

In summary, we introduce SaFeTS, a test selection approach for DNNs. Employing frequency domain transformation techniques, we extract test case semantics and then apply clustering to discern distinct categories within the frequency domain feature space. SaFeTS is designed to select semantically diverse samples, uncovering a broader spectrum of DNN misbehaviors and enhancing DNN robustness through retraining.

5. Experimental Setup

In this section, we describe the experimental setup for evaluating SaFeTS, including the research questions, datasets and DNN models, evaluation methods, baselines and parameter analysis.

5.1. Research Questions

Our evaluation covers various datasets and DNN models. We benchmark SaFeTS against baseline methods with the intent to address the following three research questions (RQs).

- **RQ1.** Can SaFeTS expose more diverse erroneous behavior of DNNs?
- **RQ2.** Can SaFeTS effectively improve the adversarial robustness of DNNs via retraining?
- **RQ3.** Can SaFeTS effectively enhance the OOD robustness of DNNs via retraining?

5.2. Models and Datasets

In the experiment, to evaluate the effectiveness of testing and retraining, we used two datasets and six pre-trained models: CIFAR-10 dataset [30] (ResNet18 [20], VGG16 [42], DenseNet121 [25]) and SVHN dataset [58] (ResNet50 [20], Wide-ResNet50 [20] and SqueezeNet [26]). We applied the advanced adversarial attack technique PGD [37] to generate the candidate set for test selection. Detailed configurations of datasets, models, and attack results are reported in Table 1.

CIFAR-10 [30]: The CIFAR-10 dataset is a widely used benchmark for image classification tasks. It comprises 60,000 32x32 color images across 10 different classes, with 6,000 images per class. The dataset is divided into 50,000 training and 10,000 test images.

SVHN [58]: The SVHN dataset, or Street View House Numbers dataset, is another popular dataset for digit recognition in natural images. It contains over 600,000 digit

images obtained from real-world street view images. The dataset is divided into training, validation, and test sets, and it covers a wide range of digit variations and orientations.

ResNet18, ResNet50, and Wide-ResNet50 [20]: ResNet18 is a convolutional neural network architecture known for its effectiveness in deep learning tasks. It consists of 18 layers and introduces the concept of residual blocks, which helps address the vanishing gradient problem and enables the training of deep networks. ResNet50 is an extension of the ResNet architecture with 50 layers. It offers even greater model capacity and captures intricate features from images. Wide-ResNet50 is an extended version of ResNet50 with wider layers for better feature learning and regularization.

VGG16 [42]: VGG16 is a deep convolutional neural network architecture developed by the Visual Geometry Group at the University of Oxford. The architecture consists of 16 layers, including 13 convolutional layers and 3 fully connected layers. VGG16's key feature is using small 3x3 convolutional filters in multiple layers to learn hierarchical features from images.

DenseNet121 [25]: DenseNet is a neural network architecture that introduces dense connections between layers. DenseNet121 is a variant with 121 layers, which includes dense blocks where each layer is connected to every other layer in a feed-forward manner.

SqueezeNet [26]: SqueezeNet is a compact neural network architecture designed for deep learning tasks on resource-constrained devices or environments. It achieves high accuracy with a significantly reduced number of parameters. The architecture employs various strategies such as 1x1 convolutions and 'fire' modules (combinations of squeeze and expand operations) to maintain model efficiency while retaining strong representational power.

Table 1. Detailed experiment configurations

Dataset	Model	Accuracy%	# Adversarial	#Error	Robustness%
CIFAR-10	ResNet18	95.35	10,000	8,953	10.47
	VGG16	93.88	10,000	9,358	6.42
	DenseNet121	94.78	10,000	8,990	10.10
SVHN	ResNet50	91.56	26,032	14,297	45.08
	Wide-ResNet50	91.69	26,032	14,435	44.55
	SqueezeNet	92.55	26,032	10,569	59.40

5.3. Evaluation methods

The evaluation methods we adopted in this section for testing and retraining are shown as below.

Diversity of misbehavior: To investigate the diversity of the misbehavior triggered by test cases, we visualized the confusion matrix of the prediction results. A confusion matrix is a table often used in classification task to illustrate the model performance on test data. In a confusion matrix, the rows represent the true labels, and the columns represent the predicted labels. Each entry in the matrix indicates the number of occurrences where a true label was predicted as a specific class. We removed the diagonal elements (correct

predictions) and then focused on the distribution of the non-diagonal elements (erroneous predictions). We evaluate the diversity of misbehavior by calculating the variance of the confusion matrix. The more uniformly distributed the misclassified samples, the smaller the variance of the confusion matrix, indicating higher diversity of the misclassified samples.

Accuracy on perturbed set (Robustness): The accuracy of the model on perturbed dataset. This metric is used to evaluate the effectiveness of SaFeTS in improving the robustness generalization of the model after retraining.

Accuracy on the original test set (Clean Accuracy): The model's accuracy on the original test data. This metric is used to evaluate the standard generalization of the model after retraining.

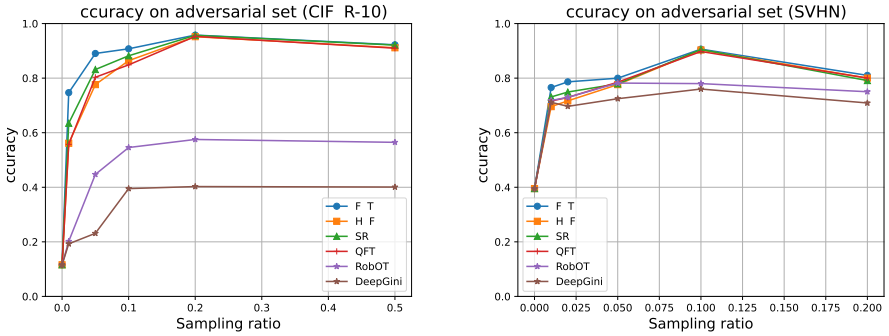
5.4. Baselines

In this paper, we use current state-of-the-art test selection techniques as baselines, including DeepGini [14] and RobOT [53]. We then evaluated these baseline methods on our proposed dataset.

DeepGini [14]: DeepGini is a test selection technique that focuses on selecting test cases based on confidence scores (computed by the Gini index) of the model predictions. It prioritizes test cases for which the model is lowly confident in its predictions. This technique explores the least confident predictions and potentially reveals the model misbehavior.

RobOT [53]: RobOT is a test selection technique based on the gradient information of the model. RobOT calculates the robustness of the test case on the basis of its gradient of the model's loss function. It prioritizes samples with large gradients, which have a higher likelihood of leading to incorrect model predictions.

5.5. Parameter Analysis



(a) CIFAR-10 **(b)** SVHN

Fig. 3. Accuracy on adversarial set with different sample ratios and test selection methods

This section analyzes the parameters of adversarial sample generation, as well as the retraining sample rate. For generating adversarial samples on CIFAR-10 and SVHN, we employ the Torchattacks library [28] to implement the advanced attack PGD [37]. For both the CIFAR-10 dataset and the SVHN dataset, we set the maximum perturbation $eps = 4/255$, and the number of steps $steps = 10$.

To choose a suitable sampling ratio for retraining, we set different sampling ratios for SVHN (ResNet50) and CIFAR-10 (ResNet18) and observed the accuracy changes on the adversarial set. Fig. 3 shows the accuracy on an adversarial set of re-trained models according to the change in the sampling ratio. We noticed that our method performs better on the CIFAR-10 and SVHN datasets than the baselines in improving robustness. However, the accuracy of the retrained model does not always increase as the sampling ratio increases. This phenomenon may be due to the excessive introduction of noisy data for training that instead destroys the performance of the model. In addition, increasing the number of training samples will increase the training overhead. Thus, we set sampling ratios of 0.2 and 0.1 for the CIFAR-10 and SVHN datasets to balance performance and time spent in the subsequent experiments.

In subsequent evaluations, to mitigate the impact of randomness in sampling, we ran the experiments five times for each configuration and then reported the mean values as the results of testing and retraining. Additionally, we carefully tuned other hyperparameters to achieve the best version of our method, such as the k-value in clustering. For other details please refer to our experimental code on GitHub.

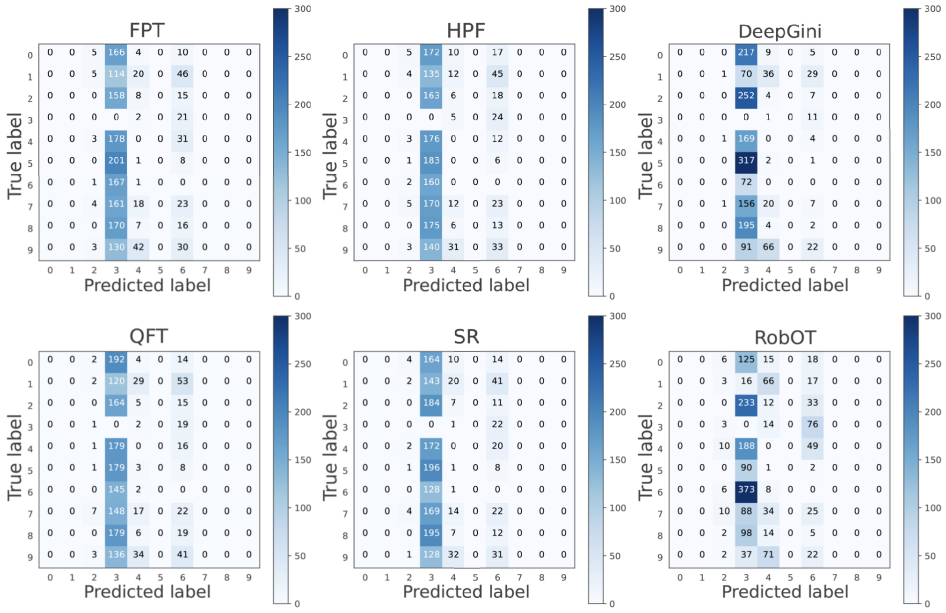


Fig. 4. The confusion matrices of test results on CIFAR-10

6. Analysis

This section reports the results of the evaluation based on three research questions mentioned in Sec. 5.1, including the evaluation on diversity of test cases, and the effect of robustness enhancement on adversarial and OOD samples.

6.1. RQ1: Evaluation of the Diversity of Selected Test Cases

We evaluated the effectiveness of SaFeTS for testing by investigating the ability of SaFeTS to expose diverse models' misbehavior. As shown in Table 1, we generated adversarial samples as the candidate set, using the PGD attack on the test set of CIFAR-10 and SVHN. Then, we adopted different test selection methods to select samples from the candidate set.

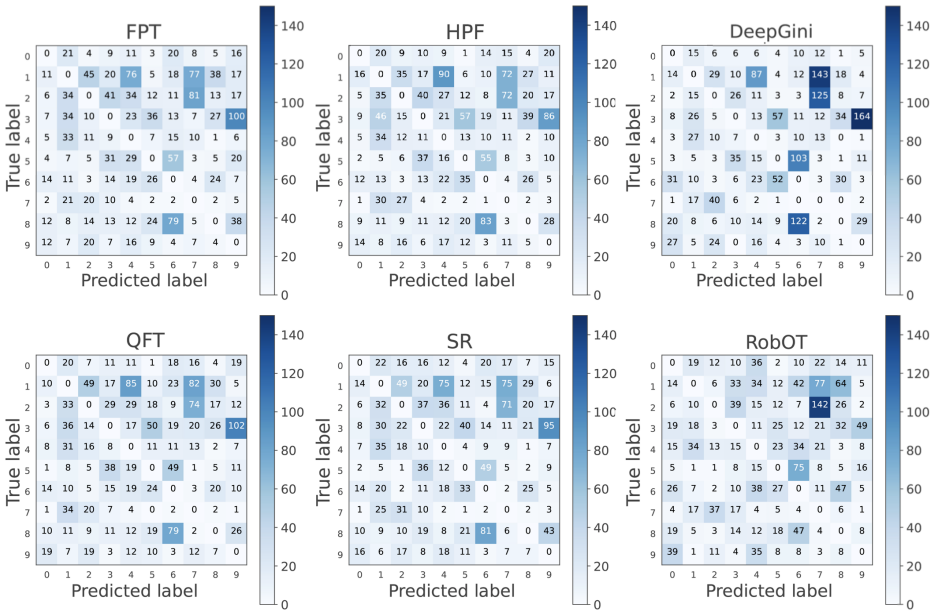
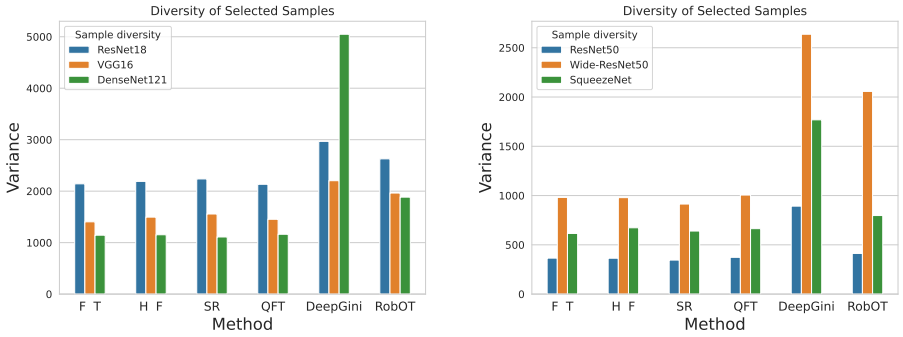


Fig. 5. The confusion matrices of test results on SVHN

Fig. 4 and Fig. 5 show the confusion matrices of the testing results on CIFAR-10 (ResNet18) and SVHN (ResNet50), respectively. According to the confusion matrices, FPT, HPF, SR, and QFT identify samples with a broader coverage of misclassifications, instead of an exclusive focus on particular errors. On the contrary, the baseline methods are heavily focused on certain predictive mistakes. In the case of CIFAR-10, DeepGini exhibits a bias towards samples where the prediction of class '5' as the class '3' occurs, while RobOT tends to emphasize instances where class '6' is misclassified as class '3'. Similarly, in SVHN, DeepGini prefers examples where class '3' is mispredicted as class



(a) CIFAR-10 **(b)** SVHN

Fig. 6. The variances of confusion matrices on different datasets

'9', while RobOT is inclined towards samples with class '2' being wrongly assigned to class '7'.

We calculated the variances associated with these confusion matrices to evaluate the model's incorrect behavior's diversity comprehensively. A lower variance in the confusion matrix indicates a more uniform distribution of misclassified samples and more diversity. Fig. 6 illustrates the variances across the confusion matrices for both CIFAR-10 (left) and SVHN (right). The variances of the samples selected by SaFeTS, namely, FPT, HPF, SR, QFT, indicate a consistently consistent trend and are smaller than those of DeepGini and RobOT. This observation underscores the enhanced diversity within SaFeTS's selection.

When we select test cases based on semantic features, we are in fact picking samples that are semantically different but could lead to classification errors. Such samples are highly diverse because they cover a wide range in sample space that could trigger error predicting. When using a frequency domain based approach, it is more easily to distinguish and capture these different semantic features. For example, HPF highlights high-frequency details, which helps to identify possible problems with the model focus in high-frequency area. Selecting samples with different semantic features through SaFeTS means that we evaluate the model's performance on multiple semantic types, rather than just focusing on a particular error pattern. Based on the experimental results, we can answer RQ1.

Answer to RQ1: Compared to baselines, SaFeTS can effectively select test cases that expose the misbehavior of more diverse models. SaFeTS mitigates the selection of samples too focused on a particular type of error and ensures the diversity and balance of the samples in subsequent retraining.

6.2. RQ2: Evaluation of the Adversarial Robustness

To evaluate SaFeTS in the adversarial robustness improvement of the model, we crafted a candidate set based on PGD attacks to generate 10 and 20 times more adversarial samples on the training sets of SVHN and CIFAR-10, respectively. Then we compared the robustness performance of baseline methods and SaFeTS by retraining on the select test cases.

We reported the robustness and accuracy of the retrained models with sampling ratios 0.2 (CIFAR-10) and 0.1 (SVHN) in Table 2. The results indicate that all four saliency detection methods outperform the baseline approach, with particular prominence observed in the performance of the FPT method. Specifically, on the CIFAR-10 dataset, the FPT, HPF, SR, and QFT methods enhance robustness by 31.21%, 30.76%, 30.96%, and 30.89%, respectively, compared to the best-performing baseline. Similarly, on the SVHN dataset, the FPT, HPF, SR, and QFT methods exhibit improvements in robustness by 10.89%, 10.66%, 10.54%, and 10.42%, respectively. According to the raw robustness reported in Table 1, after retraining, the robustness boosting on CIFAR-10 is better than that on SVHN, which is street door number data with less diversity relative to CIFAR-10. Therefore, SaFeTS performs better on more diverse datasets. In addition, the FPT works best among all the saliency detection methods, so in practice, to maximize the effectiveness of SaFeTS, more diverse data can be collected and FPT configurations can be prioritized.

In addition, SaFeTS leads to a moderate reduction in the accuracy of the retrained model, whereas retraining with DeepGini and RobOT-selected samples reduces the original accuracy of the model. For example, on CIFAR-10 and ResNet18, the FPT method reduces the model's accuracy on the original test set from 95.35% to 94.82%. The accuracy of the retrained model based on DeepGini and RobOT decreased to 70.89% and 74.49%, respectively. The experimental results illustrate that the model learns diverse semantic features by using the samples selected by SaFeTS for retraining. SaFeTS mitigates the overfitting of the perturbed data and maintains the performance of the original test set.

The experimental results demonstrate that SaFeTS can effectively distinguish and capture errors that the model may have in processing different semantic information, thus providing the model with more representative samples for retraining. Selecting samples with a high diversity is crucial for training the model, as they cover a wide range of sample space that may cause the model to make errors, which helps the model to have better generalization ability when faced with unknown adversarial samples. The frequency domain based approach selects a diverse set of samples, which reduces the risk of overfitting the model on specific types of perturbation data.

Based on the above experiments, we can answer RQ2:
Answer to RQ2: SaFeTS can effectively improve the adversarial robustness of models after retraining, especially using the FPT to extract the semantic features. FPT, HPF, SR, and QFT improved performance over the best-perform baseline by an average of 31.21%, 30.76%, 30.96%, and 30.89% on CIFAR-10, and by 10.89%, 10.66%, 10.54%, and 10.42% on SVHN, respectively.

6.3. RQ3: Evaluation of the Out-of-distribution Robustness

To verify whether SaFeTS can effectively work for improvements in OOD robustness, we investigated the performance of retraining on OOD datasets. We adopted a widely recognized corruption benchmarking [22] to generate four types of corruptions, including weather (fog), blur (Gaussian blur), noise (Gaussian noise), digital (pixelate), and five levels (level 1-5) of corrupted data on the CIFAR-10 training set for sampling. Afterward, we used the CIFAR-10-C [22] dataset to verify the robustness of generalization.

We utilized the best-perform parameters in RQ2, including using FPT to extract the semantic features of samples, setting the sampling ratio to 0.2 for each corruption type, and using K-means++ for clustering. We reported the retraining results on model ResNet18

Table 2. Accuracy on original test set (Clean Accuracy) and adversarial set (Robustness) of the retrained models

CIFAR-10	**ResNet18**		**VGG16**		**DenseNet121**	
Method	**Clean Accuracy %**	**Robustness %**	**Clean Accuracy %**	**Robustness %**	**Clean Accuracy %**	**Robustness %**
FPT	95.06	**95.78 (85.31** ↑)	93.20	**93.17 (86.75** ↑)	94.06	**95.30 (85.20** ↑)
HPF	94.82	95.19 (84.72 ↑)	92.99	92.84 (86.42 ↑)	93.80	94.88 (84.78 ↑)
SR	94.73	95.69 (85.22 ↑)	92.85	92.93 (86.51 ↑)	93.72	94.88 (84.78 ↑)
QFT	94.84	95.35 (84.88 ↑)	92.68	93.00 (86.58 ↑)	93.76	94.94 (45.31 ↑)
DeepGini	70.89	40.26 (29.79 ↑)	71.45	50.25 (43.83 ↑)	77.21	55.41 (45.31 ↑)
RobOT	74.49	57.52 (47.05 ↑)	78.94	64.33 (57.91 ↑)	78.34	68.78 (58.68 ↑)
SVHN	**ResNet50**		**Wide-ResNet50**		**SqueezeNet**	
Method	**Clean Accuracy %**	**Robustness %**	**Clean Accuracy %**	**Robustness %**	**Clean Accuracy %**	**Robustness %**
FPT	93.52	**90.63 (45.55** ↑)	93.21	**90.44 (45.89** ↑)	92.71	**89.23 (29.83** ↑)
HPF	92.83	90.32 (45.24 ↑)	92.98	90.25 (45.70 ↑)	93.04	89.05 (29.65 ↑)
SR	93.25	90.49 (45.41 ↑)	93.03	89.72 (45.17 ↑)	92.87	89.05 (29.65 ↑)
QFT	92.98	89.76 (44.68 ↑)	93.14	90.11 (45.56 ↑)	92.62	89.04 (29.64 ↑)
DeepGini	88.06	77.99 (32.91 ↑)	88.25	78.21 (33.66 ↑)	89.12	79.87 (20.47 ↑)
RobOT	91.89	76.01 (30.93 ↑)	91.58	79.64 (35.09 ↑)	92.15	80.01 (20.61 ↑)

Table 3. Accuracy on original test set (Clean Accuracy) and OOD set (Robustness) of the retrained models

Corruption	**Method**	**Clean Accuracy %**	**Robustness %**
Weather	SaFeTS(FPT)	97.35	**97.34**
	RobOT	98.49	94.32
	DeepGini	98.65	96.05
Blur	SaFeTS(FPT)	87.99	**95.44**
	RobOT	91.08	91.28
	DeepGini	92.09	94.19
Noise	SaFeTS(FPT)	96.51	**96.78**
	RobOT	96.51	95.34
	DeepGini	96.47	96.47
Digital	SaFeTS(FPT)	98.43	**95.01**
	RobOT	99.11	90.27
	DeepGini	99.07	91.47

in Table 3. The experimental results show that, for the weather, blur, noise, and digital corrupted data, SaFeTS has effectively improved the robust generalization compared to baselines, with a minor performance degradation on the original test set.

Unlike adversarial samples, OOD data are those that have not appeared in the training data, but are not intentionally designed to deceive the model. The model may be confused by specific noises or details in the data and miss the global characteristics. Choosing samples with diversity can help the model to better generalize to a wide variety of data, not just a specific subset of the training set. Experimental results show that SaFeTS can help models focus on more fundamental properties and avoid being distracted by unnecessary noise. By enhancing the diversity of the retraining samples, it can be ensured that the model has been trained in a variety of different contexts, thus improving its robustness in the face of unknown data. In summary, SaFeTS helps to enhance the robustness and generalization ability of the model on OOD data.

Based on the experimental results, we can answer RQ3: **Answer to RQ3:** SaFeTS can effectively improve OOD robustness of DNN models on various types of corruption data, the overall results indicate that our method can improve the performance over best-perform baseline.

7. Limitation

In this section, we discuss the limitations of SaFeTS. From the perspective of time overhead, SaFeTS depends on the convergence speed of the clustering algorithm. To explore the impact of clustering methods on the results, we also used a clustering algorithm different from Kmeans++, the GMM, which is a density-based clustering method. We reported the retraining results of CIFAR-10 (ResNet18) and SVHN (ResNet50) with two clustering methods in Table 4 (selecting samples from adversarial sets).

Table 4. The accuracy on an adversarial set of the retrained models (Robustness) and clustering time under different clustering methods (Kmeans++ and GMM)

Dataset	Method	Robustness %		Time (s)	
		Kmeans++	GMM	Kmeans++	GMM
CIFAR-10	PFT	**95.78**	95.42	**980.92**	4221.60
	HPF	95.19	**95.60**	**1417.65**	3008.59
	SR	**95.69**	95.41	**1480.64**	4558.23
	QFT	95.35	**95.88**	**2028.23**	4292.48
SVHN	PFT	**90.63**	90.07	**491.45**	2247.11
	HPF	**90.32**	90.20	**852.56**	3004.25
	SR	**90.49**	90.44	**845.57**	2457.21
	QFT	**89.76**	88.12	**1074.54**	2542.44

The experimental results show that the choice of clustering method impacts the performance of SaFeTS. Clustering with Kmeans++ performs better than GMM. The time

overhead of SaFeTS is more extensive compared to baseline methods. However, the training time for DNNs is much longer than the time overhead of test selection. Therefore, the time overhead of test selection can almost be ignored. In our retraining experiments, with the addition of 73,257 and 200,000 samples to SVHN and CIFAR-10, the time needed for each epoch is approximately 16 minutes and 20 minutes, respectively. And each training is run separately on a single 3090 GPU. Thus, it is worth trading more test selection time for better testing and retraining results. Nevertheless, to reduce the time overhead of sample selection, more efficient clustering algorithms or parallel computing can be attempted to speed up the process.

8. Conclusion

In summary, this research proposed SaFeTS, a novel semantic feature-based test selection approach for DNNs grounded in frequency domain analysis. The core innovation lies in utilizing saliency detection methods to extract semantic features aligned with human perception. SaFeTS then discerns diverse categories within this semantic feature space via clustering. By sampling test cases from each cluster, SaFeTS selects data with rich semantics for testing and retraining DNNs. Extensive experiments highlighted the ability of SaFeTS to uncover a more varied model misbehavior compared to existing selection techniques. Further, retraining on the semantically diverse samples enhanced adversarial and out-of-distribution robustness, with particularly strong gains of over 20% on adversarial accuracy. The proposed frequency domain viewpoint thus offers valuable new perspectives on testing and improving DNN robustness. Limitations like clustering overhead present opportunities for future work on optimizing SaFeTS. By releasing our method publicly, this research aims to provide an effective resource to advance rigorous engineering of trustworthy DNNs. In future work, we will aim to investigate the generalization of the SeFeTS approach to other domains besides image classification and computer vision, especially security-critical domains.

Acknowledgments. This work was supported by the Talent Fund of Beijing Jiaotong University, No. 2023XKRC041, and National Natural Science Foundation of China, No. 62273027, and Beijing Natural Science Foundation, No. L211021.

References

1. Abdar, M., Pourpanah, F., Hussain, S., Rezazadegan, D., Liu, L., Ghavamzadeh, M., Fieguth, P., Cao, X., Khosravi, A., Acharya, U.R., et al.: A review of uncertainty quantification in deep learning: Techniques, applications and challenges. Information Fusion 76, 243–297 (2021)
2. Aghababaeyan, Z., Abdellatif, M., Briand, L., S, R., Bagherzadeh, M.: Black-box testing of deep neural networks through test case diversity. IEEE Transactions on Software Engineering 49(5), 3182–3204 (2023)
3. Alves, E.E., Bhatt, D., Hall, B., Driscoll, K., Murugesan, A., Rushby, J.: Considerations in assuring safety of increasingly autonomous systems. Tech. rep. (2018)
4. Arthur, D., Vassilvitskii, S.: K-means++ the advantages of careful seeding. In: Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms. pp. 1027–1035 (2007)
5. Bazarbash, M.: Fintech in financial inclusion: Machine learning applications in assessing credit risk. Tech. rep., International Monetary Fund (2019), technical report

6. Biagiola, M., Stocco, A., Ricca, F., Tonella, P.: Diversity-based web test generation. In: Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering. pp. 142–153 (2019)
7. Boashash, B.: Time-frequency signal analysis and processing: a comprehensive reference (2015)
8. Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L.D., Monfort, M., Muller, U., Zhang, J., et al.: End to end learning for self-driving cars. arXiv preprint arXiv:1604.07316 (2016)
9. Borandag, E., Ozcift, A., Kilinc, D., Yucalar, F.: Majority vote feature selection algorithm in software fault prediction. Computer Science and Information Systems 16(2), 515–539 (2019)
10. Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: 2017 IEEE Symposium on Security and Privacy (sp). pp. 39–57. IEEE (2017)
11. Chen, G., Peng, P., Ma, L., Li, J., Du, L., Tian, Y.: Amplitude-phase recombination: Rethinking robustness of convolutional neural networks in frequency domain. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 458–467 (2021)
12. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. pp. 248–255. Ieee (2009)
13. Feldt, R., Poulding, S., Clark, D., Yoo, S.: Test set diameter: Quantifying the diversity of sets of test cases. In: 2016 IEEE 8th International Conference on Software Testing, verification and validation (ICST). pp. 223–233. IEEE (2016)
14. Feng, Y., Shi, Q., Gao, X., et al.: Deepgini: prioritizing massive tests to enhance the robustness of deep neural networks. In: Proceedings of the 29th ACM SIGSOFT International Symposium on Software Testing and Analysis. pp. 177–188 (2020)
15. Gao, X., Feng, Y., Yin, Y., Liu, Z., Chen, Z., Xu, B.: Adaptive test selection for deep neural networks. In: Proceedings of the 44th International Conference on Software Engineering. pp. 73–85 (2022)
16. Guo, C., Ma, Q., Zhang, L.: Spatio-temporal saliency detection using phase spectrum of quaternion fourier transform. In: 2008 IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–8. IEEE (2008)
17. Guo, J., Jiang, Y., Zhao, Y., Chen, Q., Sun, J.: Dlfuzz: Differential fuzzing testing of deep learning systems. pp. 739–743 (2018)
18. Han, S., Lin, C., Shen, C., Wang, Q., Guan, X.: Interpreting adversarial examples in deep learning: A review. Association for Computing Machinery (2023), https://doi.org/10.1145/3594869
19. Hartigan, J.A., Wong, M.A.: Algorithm as 136: A k-means clustering algorithm. Journal of the royal statistical society. series c (applied statistics) 28(1), 100–108 (1979)
20. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778 (2016)
21. Hemmati, H., Fang, Z., Mantyla, M.V.: Prioritizing manual test cases in traditional and rapid release environments. In: 2015 IEEE 8th International Conference on Software Testing, verification and validation (ICST). pp. 1–10. IEEE (2015)
22. Hendrycks D, D.T.: Benchmarking neural network robustness to common corruptions and perturbations. arXiv preprint arXiv:1903.12261 (2020)
23. Hou, X., Zhang, L.: Saliency detection: A spectral residual approach. In: 2007 IEEE Conference on computer vision and pattern recognition. pp. 1–8. IEEE (2007)
24. Huang, B., Tao, C., Lin, R., Wong, N.: Frequency regularization for improving adversarial robustness. arXiv preprint arXiv:2212.12732 (2022)
25. Huang, G., Liu, Z., Van Der Maaten, L., et al.: Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4700–4708 (2017)

26. Iandola, F.N., Han, S., Moskewicz, M.W., Ashraf, K., Dally, W.J., Keutzer, K.: Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size. arXiv preprint arXiv:1602.07360 (2016)
27. Imamoglu, N., Lin, W., Fang, Y.: A saliency detection model using low-level features based on wavelet transform. IEEE transactions on multimedia 15(1), 96–105 (2012)
28. Kim, H.: Torchattacks: A pytorch repository for adversarial attacks. arXiv preprint arXiv:2010.01950 (2020)
29. Koch, C., Poggio, T.: Predicting the visual world: Silence is golden. Nature Neuroscience pp. 2(1):9–10 (1999)
30. Krizhevsky, A., et al.: Learning multiple layers of features from tiny images (2009)
31. Langford, M.A., Cheng, B.H.: Enki: a diversity-driven approach to test and train robust learning-enabled systems. ACM Transactions on Autonomous and Adaptive Systems (TAAS) 15(2), 1–32 (2021)
32. Lazić, L.: Software testing optimization by advanced quantitative defect management. Computer Science and Information Systems 7(3), 459–487 (2010)
33. Li, J., Levine, M.D., An, X., Xu, X., He, H.: Visual saliency based on scale-space analysis in the frequency domain. IEEE Transactions on Pattern Analysis and Machine Intelligence 35(4), 996–1010 (2013)
34. Li, Z., Ma, X., Xu, C., Cao, C.: Structural coverage criteria for neural networks could be misleading. In: 2019 IEEE/ACM 41st International Conference on Software Engineering: New Ideas and Emerging Results (ICSE-NIER). pp. 89–92. IEEE (2019)
35. Ma, L., Juefei-Xu, F., Zhang, F., et al.: Deepgauge: Multi-granularity testing criteria for deep learning systems. pp. 120–131 (2018)
36. Ma, L., Zhang, F., Sun, J., et al.: Deepmutation: Mutation testing of deep learning systems. pp. 100–111. IEEE (2018)
37. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083 (2017)
38. Mehrtash, A., Wells, W.M., Tempany, C.M., Abolmaesumi, P., Kapur, T.: Confidence calibration and predictive uncertainty estimation for deep medical image segmentation. IEEE Transactions on Medical Imaging 39(12), 3868–3878 (2020)
39. Oppenheim, A., Lim, J.: The importance of phase in signals. Proceedings of the IEEE 69(5), 529–541 (1981)
40. Pei, K., Cao, Y., Yang, J., Jana, S.: Deepxplore: Automated whitebox testing of deep learning systems. pp. 1–18 (2017)
41. Shen, Z., Liu, J., He, Y., Zhang, X., Xu, R., Yu, H., Cui, P.: Towards out-of-distribution generalization: A survey. arXiv preprint arXiv:2108.13624 (2021)
42. Simonyan, K., Zisserman, A.: Very deep convolutional net-works for large-scale image recognition. In International Conference on Learning Representations (ICLR) (2015)
43. Song, H., Gao, S., Li, Y., Liu, L., Dong, H.: Train-centric communication based autonomous train control system. IEEE Transactions on Intelligent Vehicles 8(1), 721–731 (2023)
44. Song, H., Sun, Z., Wang, H., Qu, T., Zhang, Z., Dong, H.: Enhancing train position perception through ai-driven multi-source information fusion. Control Theory and Technology pp. 1–12 (2023)
45. Sun, J., Mehra, A., Kailkhura, B., et al.: Certified adversarial defenses meet out-of-distribution corruptions: Benchmarking robustness and simple baselines. arXiv preprint arXiv:2112.00659 (2021)
46. Sun, Y., Huang, X., Kroening, D., Sharp, J., Hill, M., Ashmore, R.: Deepconcolic: testing and debugging deep neural networks. In: 2019 IEEE/ACM 41st International Conference on Software Engineering: Companion Proceedings (ICSE-Companion). pp. 111–114. IEEE (2019)
47. Sun, Y., Wu, M., Ruan, W., Huang, X., Kwiatkowska, M., Kroening, D.: Concolic testing for deep neural networks. In: Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering. pp. 109–119 (2018)

48. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199 (2013)
49. Tian, Y., Pei, K., Jana, S., Ray, B.: Deeptest: Automated testing of deep-neural-network-driven autonomous cars. In: Proceedings of the 40th International Conference on Software Engineering. pp. 303–314 (2018)
50. Torralba, A., Oliva, A.: Statistics of natural image categories. Network: computation in neural systems 14(3), 391 (2003)
51. Wang, H., Wu, X., Huang, Z., Xing, E.P.: High-frequency component helps explain the generalization of convolutional neural networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8684–8694 (2020)
52. Wang, H., Wang, Y., Ren, W.: Image denoising using anisotropic second and fourth order diffusions based on gradient vector convolution. Computer Science and Information Systems 9(4), 1493–1511 (2012)
53. Wang, J., Chen, J., Sun, Y., et al.: Robot: Robustness-oriented testing for deep learning systems. In: 2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE). pp. 300–311. IEEE (2021)
54. Wang, Z., You, H., Chen, J., Zhang, Y., Dong, X., Zhang, W.: Prioritizing test inputs for deep neural networks via mutation analysis. In: 2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE). pp. 397–409. IEEE (2021)
55. Weiss, M., Tonella, P.: Simple techniques work surprisingly well for neural network test prioritization and active learning (replicability study). In: Proceedings of the 31st ACM SIGSOFT International Symposium on Software Testing and Analysis. pp. 139–150 (2022)
56. Xie, X., Ma, L., Juefei-Xu, F., et al.: Deephunter: a coverage-guided fuzz testing framework for deep neural networks. pp. 146–157 (2019)
57. Xu, Z.Q.J., Zhang, Y., Xiao, Y.: Training behavior of deep neural network in frequency domain. In: Neural Information Processing: 26th International Conference, ICONIP 2019, Sydney, NSW, Australia, December 12–15, 2019, Proceedings, Part I 26. pp. 264–274. Springer (2019)
58. Y. Netzer, T. Wang, A.C.: Reading digits in natural images with unsupervised feature learning. arXiv preprint arXiv:1412.6806 (2011)
59. Yin, D., Gontijo Lopes, R., Shlens, J., Cubuk, E.D., Gilmer, J.: A fourier perspective on model robustness in computer vision. Advances in Neural Information Processing Systems 32 (2019)
60. Zhang, C., Bengio, S., Hardt, M., Recht, B., Vinyals, O.: Understanding deep learning (still) requires rethinking generalization. Communications of the ACM 64(3), 107–115 (2021)
61. Zhang, Z., Xie, X.: On the investigation of essential diversities for deep learning testing criteria. In: 2019 IEEE 19th International Conference on Software Quality, Reliability and Security (QRS). pp. 394–405 (2019)
62. Zhao, C., Mu, Y., Chen, X., Zhao, J., Ju, X., Wang, G.: Can test input selection methods for deep neural network guarantee test diversity? a large-scale empirical study. Information and Software Technology 150, 106982 (2022)

Zhouxian Jiang received the BS degree in computer science and technology from Beijing Jiaotong University, Beijing, China. She is currently pursuing a PhD in software engineering at Beijing Jiaotong University, Beijing, China. Since 2017, she has worked in the Software Evaluation Laboratory of Beijing Jiaotong University. Her research interests include deep learning testing, software reliability testing, and deep learning uncertainty. She has been engaged in software testing for 5 years, and has participated in more than 10 software evaluation projects, research projects, as well as National Key Research and Development Projects (China). E-mail: zhouxianjiang@bjtu.edu.cn

Honghui Li received her MS degree in computer science from the Central South University, Changsha, China, in 1987. Her research interests include software testing technology and test automation. Currently, she is a Professor at the School of Computer and Information Technology, Beijing Jiaotong University, Beijing, China. She is currently the Deputy Director of the Engineering Research Center of Network Management Technology for High Speed Railway of MOE, Beijing, China. She has long been engaged in software quality assurance technology, high-reliability software, data mining analysis, and railway information technology. She has undertaken national-level provincial and ministerial research such as the National 863 Program, the Nuclear High-Level Project, and the China Railway Corporation. E-mail: hhli@bjtu.edu.cn.

Xuetao Tian received the BS and PhD degree in computer science and technology from Beijing Jiaotong University, Beijing, China. He is now a postdoctoral researcher in Faculty of Psychology, Beijing Normal University, Beijing, China. His research interests include personality prediction, recommender systems, and information extraction. He has published 12 research papers at international journals and conference proceedings. E-mail: xttian@bnu.edu.cn.

Rui Wang received the BS and MS degrees in electronic and information engineering from Beijing Jiaotong University, Beijing, China, and the PhD degree in computer science from the National Institute of Applied Science (INSA) Toulouse, France. Her doctoral research work was conducted at the Laboratory for Analysis and Architecture of Systems, French National Centre for Scientific Research (LAAS-CNRS). Since 2018, she has been an Assistant Professor at the Computer and Information Technology School in Beijing Jiaotong University. Her main fields of interest include machine learning testing, adversarial robustness, and uncertainty quantification. E-mail: rui.wang@bjtu.edu.cn.

Received: September 07, 2023; Accepted: November 18, 2023.

Computer Science and Information Systems 21(4):1523–1546 https://doi.org/10.2298/CSIS230907046S

AI Large Models Bring Great Opportunities to Reusable Design of CAD Software

Yunlei Sun, Bingyi Yan, and Zhaotong Shao

Qingdao Institute of Software, College of Computer Science and Technology,
China University of Petroleum (East China)
266580 Qingdao,China
sunyunlei@upc.edu.cn
z21070210@s.upc.edu.cn
z21070211@s.upc.edu.cn

Abstract. With the rapid development of artificial intelligence (AI), large models have achieved significant breakthroughs in general-purpose domains. However, their application in computer-aided design (CAD) software is still in its early stages. Reusable design is crucial for improving efficiency and innovation in CAD systems. This paper reviews progress in rule-based reasoning (RBR) and case-based reasoning (CBR), two prevailing techniques for reusable design. RBR represents expert knowledge as rules but lacks self-learning capabilities. CBR draws on prior cases to solve new problems but relies heavily on surface empirical knowledge. Recent advances in large AI models provide new opportunities to enhance reusable design, thanks to superior language and reasoning abilities.

However, adapting large models to effectively leverage CAD-specific design knowledge presents open challenges. To advance progress in this area, this paper analyzes the potential impacts of large models on improving knowledge acquisition, case retrieval, rule representation, and reasoning explain ability for hybrid CBR-RBR systems, and proposes a reusable design framework combing large language models, knowledge graphs, and databases to realize more intelligent and interpretable reuse. This review synthesizes key developments in RBR, CBR, and large AI models, highlighting promising directions for advancing reusable design in CAD software. The integration of reasoning techniques with large models, opening promising new directions for computer-aided engineering enhanced by artificial intelligence, as well as lays the foundation for more efficient, innovative, and sustainable engineering design.

Keywords: large model, CBR, RBR, reusable design.

1. Introduction

CAD technology since its inception through continuous updating and iteration [25], has been widely used in automotive, electronics, aviation, aerospace, chemical industry, construction, medical, civil engineering and other industries, to promote the traditional product design methods and production mode of continuous change and innovation, resulting in huge economic and social benefits. It has produced great economic and social benefits.

Every designer consults the relevant design drawings and data during the design process of new product development and tries to reuse the results of previous designs as

much as possible. Practice has shown that despite the rapid changes in customer-oriented products, the components that make up a product are relatively stable. For simple parts in general, more than half of the parts can be found in the main types and basic types that can be selected or referred to, and more than 50 per cent of the parts do not need to be re-designed. As a result, the total number of drawings can be reduced by 10 per cent and the number of new working drawings can be reduced by 30 per cent. And a series of new products inherited from the old product more results, about 80 percent [10]. If each product development and design are from scratch, in the whole development and design process can not give full play to the role of the enterprise's existing technology and resources, and there is bound to be a lot of duplication of labour, these duplication of labour mainly from the product design (conceptual design, detailed design, process design) and a variety of design work around the repetition of the product. In the process of new product development, we should try to use the existing design resources of the enterprise to reduce the duplication of labour in the development work.

The so-called design reuse ideas, refers to the product function design, principle programme design, structural programme design, overall design, construction design, process design and other design activities around the product reuse, reference or reference to existing design results, new product development and design of various views and ideas [35], design reuse technology is to integrate the design reuse idea into the broad design process of the whole life cycle of the product [36]. Every designer has the idea of design reuse in his mind, but due to the early computer hardware and software technology can not meet the requirements of design reuse, design reuse has not been paid attention to and widely used. Each designer according to their own understanding of their own design reuse behaviour, there is no standard design reuse organisation and management.

From the development of mechanical design methods, people's exploration of CAD software design reuse is reviewed. There are similarities in the elements and layout of different equipment drawings, so it is feasible to effectively reuse the existing drawing results in the generation of new drawings. In the manual drawing period, engineers reuse existing design drawings in order to improve design efficiency. When drawing new products, they put existing drawings that can be used for reference on the table, cover the non-reusable parts, and re-copy the reusable parts to achieve the reuse of drawings. However, manual drawing is prone to errors and leads to more rework, resulting in low efficiency. With the advent of CAD software, with the help of computers, engineers can save reusable drawings as new design drawings, which can be modified on the basis of the original. However, neither of these two methods can use the existing results, and both need to be modified, and the reuse efficiency is low. With the deepening of research on reuse knowledge, engineers use the modular design idea to make simple reusable graphic parts in drawings into blocks for reuse, improving the reuse efficiency of parts. However, the block reuse method lacks the guidance of scientific ideas, and the management is difficult, so the design reuse idea cannot be truly implemented. How to apply computer aided design technology to reduce drawing reuse time, shorten drawing design cycle and improve drawing utilization is always a difficult problem in the field of mechanical drawing.

CAD software in the application of design reuse technology, its impact is mainly reflected in the following aspects.

1. New product development and design capabilities and innovation will be improved. As the saying goes, "learn from the past to understand the new", when engineers carry

out new product development and design, due to the support of the design reuse system, the designers' development and design ability and innovation ability will be greatly improved, so as to enhance the enterprise's new product development and design ability and innovation ability.

2. Shortening the time to market for new products. Since most new products are improvements on the basis of old products, there is a lot of information that can be reused, so by designing a reusable system, the workload of designers and the production preparation time are reduced, which of course shortens the time to market for new products.

3. Reducing product costs. Each new part will increase the manufacturing cost of the enterprise and increase the cost of the product. However, design reuse technology will avoid the emergence of new parts to the greatest extent possible, and at the same time reduce the design cost by making full use of the reusable information. Therefore, the total cost of the product will be reduced.

4. As the parts and components reused in new products are tested in production practice, so that the quality and function of many parts and components are guaranteed, relatively speaking, the quality of the product is guaranteed.

Design reuse technology can bring so much value to enterprises, but the design reuse rate in CAD software is still low. Due to the lack of scientific standardization and standardization of design resources and achievements, engineers have too many choices when reusing existing parts and lack of selection basis. Even if the existing design results are found, because the parts are not modeled with standardized and parametric methods, the time for engineers to modify the existing results exceeds the time for redesign, resulting in engineers not choosing reuse. In addition, the traditional performance appraisal of enterprises is based on the number of engineering drawings produced by engineers to evaluate the performance of engineers, which leads engineers to think that designing a new part and creating an additional engineering drawing is the embodiment of their labor costs. All these reasons are the challenges faced by CAD software reuse design.

In order to make full use of the existing design knowledge and the constructed knowledge base to shorten the development cycle of new product design, scholars have carried out a large number of researches on the techniques related to product design knowledge reuse, such as model similarity evaluation [18], model retrieval, and knowledge navigation, etc. A lot of research has been carried out on the evaluation of similarity and retrieval of CAD assembly models considering a variety of evaluation criteria [20, 21], free-form surface retrieval, efficient retrieval of assembly models based on improved Hausdorff distance, 3D CAD model retrieval based on image recognition [50, 14], and design knowledge reuse based on knowledge graphs or knowledge templates.

Knowledge-based reasoning is a thinking process in which new knowledge is inferred from one's own knowledge through a computer system according to a certain strategy, i.e., after inputting a problem to be solved, the existing design cases, expert experience, design formulas, specifications and other knowledge in the knowledge base are called upon to simulate human thinking, so as to complete the process of solving the problem [7, 17]. Common knowledge-based reasoning methods are rule-based reasoning (RBR), case-based reasoning (CBR) and Hybrid CBR-RBR. CBR and RBR are the key technologies in the field of AI, and they are widely used in many application fields such as natural language processing, image recognition and expert system. These technologies can im-

prove our understanding and use of past experience and knowledge to better solve current problems, which is conducive to improving reusable design.

1. **RBR** is a method of making inferences based on a collection of pre-defined rules. These rules describe the causal relationships under specific conditions and the actions that should be taken when these conditions are met. For example, expert systems typically use rules to model the decision-making process of a human expert. The inference engine will infer new conclusions or recommendations based on known rules and input conditions. This approach is suitable for problem domains that can be expressed in terms of explicit rules [47, 26].

2. **CBR** is a method of learning and inferring from previously experienced cases. It uses an existing library of cases to solve new problems by comparing and matching similar problems that have been solved previously to find the best solution. This approach is similar to the process of human learning and experience building by using past cases to guide decision making. The CBR approach is suitable for situations where problems need to be solved based on prior experience with similar situations [15, 22].

3. **Hybrid CBR-RBR** Combining the characteristics of RBR and CBR, often using a combination of the two types of reasoning.

In summary, all three methods of reasoning have their applications in different contexts. RBR is suitable for domains that can be explicitly expressed as rules, while CBR is suitable for scenarios that draw on past experience to solve problems. In practice, the appropriate reasoning method is usually chosen based on the nature of the problem and the available data.

With the development of AI large model, the reuse design in CAD software has a new breakthrough point. This paper aims to propose a new framework combining AI large model to optimize the reuse design in CAD software.

The remaining papers are arranged as follows. Chapter 2 introduces the working process and characteristic analysis of artificial intelligence big model, RBR and CBR. Chapter 3 introduces the research and analysis methods. The Chapter 4 discusses the research results and propose a framework for reuse design based on AI large model.

2. Literature Review

This chapter introduces the working process and characteristic analysis of general and domain AI large models, RBR, and CBR.

2.1. Large Models of AI Opportunities of the Times

Artificial Intelligence (AI) [11] is an important driving force of the fourth industrial revolution and a core technology for digital transformation [48]. With the "explosive" growth of data volume and the rapid improvement of arithmetic power, AI technology is ushering in a new wave of innovation, of which the most striking is the large model technology.

Generalised Large Model of AI On November 30, 2022, OpenAI launched ChatGPT [9], a new conversational general AI tool. It was researched that within just a few days of its launch, the number of registered users exceeded 1 million, and the number of active

users had reached 100 million in 2 months, attracting widespread attention from all walks of life, making it the fastest-growing consumer application in history, and setting off a technological tidal wave in the field of AI.

In recent years, many epoch-making large models have emerged, such as OpenAl's GPT-4, Huawei Cloud's Pangu NLP, Baidu's Wenshin, and meta's llama2 [40]. These large models have made landmark technological breakthroughs in the field of natural language processing [4, 6], and have achieved leapfrog development in terms of model accuracy, generality, and generalisation ability, and have achieved multi-scenario applications in the fields of finance, healthcare, media, and gaming, which have improved the efficiency, lowered the cost, and created value.

Large language models can be divided into pre-trained models and fine-tuned models. Pre-trained models refer to language models that are pre-trained on large-scale text corpora, such as Transformer and BERT. A fine-tuning model is a model that is fine-tuned to a pre-trained model for a specific task, such as ERNIE Bot, GPT-3, etc.

The release of GPT-3.5 and its great success has had a strong impact on the AI industry, in which many previously unsolved problems have been found to be solved (including fact-based quizzing, text-summary fact consistency [16], etc.). However, from another perspective, we can also think of large models as a tool to assist in the development, optimisation of models, and enrichment of application scenarios, e.g..

1. Code Development: Using ChatGPT to assist in code development and improve development efficiency, including code completion [37], natural language instructions to generate code, code translation, bug fixes, etc.

2. Combination of ChatGPT and specific tasks: ChatGPT generates results that are significantly better on many tasks compared to fine-tuned miniatures, and combines the strengths of ChatGPT on the basis of fine-tuned miniatures in order to improve the inference effect of miniatures.

3. Meanwhile, based on the ability of less sample learning inspired by ChatGPT instruction fine-tuning, for tasks with only a few annotations or no annotated data as well as tasks that require out-of-distribution generalisation [13], ChatGPT can be applied directly as well as used as a tool for cold-starting to collect relevant corpus [28], enriching the application scenarios.

Reviewing the underlying technology of the large model, the breakthroughs are mainly the following:

Vaswani A proposed a network architecture Transformer [41], which introduces the self-attention mechanism. Transformer outperforms other sequence transformation models in terms of machine translation quality and efficiency, and shows strong generalisation ability, which can be applied to other natural language processing tasks. The proposal of Transformer provides new ideas and references for sequence learning and neural network Machine translation research provides new ideas and references, and the architecture has become one of the mainstream frameworks for machine translation and other sequence learning tasks.

Reinforcement learning with human feedback (RLHF) techniques have made great strides in empowering intelligences to learn from external human suggestions, with RLHF acting as an important bridge to incorporate human feedback into the learning process by constructing a human feedback dataset and training an incentive model that mimics human preferences for scoring outcomes, allowing machines to learn by abstracting human

values rather than simply mimicking human behaviour [19, 49, 32]. This is the core technology at the heart of the growing resemblance of human dialogue in large language models in the post-gpt-3 era. RLHF has recently come into the public eye through several high-profile AI large models, including OpenAI's ChatGPT, DeepMind's Sparrow, and Anthropic's Claude.

Bidirectional Encoder Representations from Transformers (BERT) [5] achieves breakthroughs in several natural language processing tasks. BERT achieves SOTA on 11 Natural Language Processing (NLP) tasks that demonstrating its powerful generalisation capabilities. BERT can be migrated to different tasks with simple fine-tuning without major changes to the model architecture, which greatly reduces the workload of developing the model. The proposal of the BERT model opens a new chapter in large-scale language understanding in NLP. Its powerful representation learning and task migration capabilities have led to its rapid application in various subfields of NLP, generating a wide range of impacts and greatly advancing the progress of NLP technology.

GPT is an approach based on generative pre-training and discriminative fine-tuning [29] to achieve transfer learning for NLP tasks, using task-oriented input transformations to achieve migration with only a few changes to the model architecture, empirically demonstrating the effectiveness of the approach on 12 tasks to achieve SOTA, and proposing a generalised framework to address the problem of data scarcity in NLP task learning. This paper is significant in the research of transfer learning and end-to-end learning in NLP. The authors' proposed approach provides an effective and practical framework for solving the data scarcity problem for different tasks.

The GPT-2 model [30] can automatically discover tasks from large-scale text data. pre-training the language model with the large-scale web dataset Web Text allows it to perform NLP tasks with zero-sample learning. proving that the capacity of the language model is crucial for migration learning and that a larger model achieves better performance. the model GPT-2 achieves SOTA on seven language modelling datasets. but still insufficient to fit the full Web Text. the authors demonstrate that pre-training of large-scale semantic models on suitable datasets can achieve zero-sample learning and adaptation for NLP tasks. This informs the solution to the problem of data scarcity and the construction of language models that can learn tasks from examples as humans do.

The T5 framework advances the development of transfer learning by systematically investigating different transfer learning methods and comparing them on multiple NLP downstream tasks, based on the Colossal Clean crawled corpus and model size, and reaching SOTA on many benchmark tests, T5 is important in the research and application of transfer learning in NLP [31].

The large-scale language model GPT-3 meets or exceeds the previous SOTA on many NLP tasks in a sample less setting, demonstrating the performance improvement that comes with scale, and GPT-3 performs at the human level on some tasks, but also faces methodological challenges. The robustness of GPT-3 foreshadows the significant advances that may be possible in deep learning in the field of NLP, but it also suggests that researchers need to confront and address the wide-ranging implications of AI [3].

Touvron [39] introduces Llama, a set of language models at different scales, where the authors use only publicly available datasets to train state-of-the-art language models that match or exceed proprietary models in terms of performance, opening up greater options for language modelling research and applications. Llama is important in the development

and application of large-scale language models and neural network models, reducing the language modelling research barriers, allowing more researchers to access and develop large-scale neural network models, and helping to advance the technology in this area.

A large-scale multimodal language model, GPT-4, has been introduced, and GPT-4 meets or exceeds the human level in AI benchmark tests [34]. The model is developed using a scalable framework that can guarantee performance at different scales and provide experience for subsequent larger-scale models. The model provides a valuable reference for the study of multimodal intelligences and the development of artificial general intelligence .The powerful capability of GPT-4 indicates that the development of AI has entered a new stage in the development of AI, which points out the direction for the future development of AI.

In addition, popular language models include:

ERNIE Bot: ERNIE Bot is a knowledge-enhanced large language model developed by Baidu that can generate high-quality text content. It is based on the Transformer architecture, has 350 million parameters, and supports both Chinese and English.

PaLM-E: PaLM-E is a large language model from the Google Brain team with 540 billion parameters. Its unique feature is that it can combine language model and visual model to realize multi-modal understanding and generation.

XLNet: XLNet is a new pre-trained language model that combines the benefits of Transformer-XL and BERT. It is capable of handling longer text paragraphs and has greater generalization ability.

RoBERTa: RoBERTa is a pre-trained language model developed by Facebook AI, based on the BERT architecture. It has been trained by a large number of corpora and has a strong ability of natural language understanding and generation.

TechGPT: TechGPT has enhanced various information extraction tasks such as relational triplet extraction with "knowledge graph construction" as the core, which means that TechGPT has a stronger ability to handle information extraction tasks and can better understand and process various types of information.

Domain Large Models in AI Domain large models are large language models that have been trained and optimised in a specific domain or industry. Compared with general-purpose larde models, vertical domain large models concentrate more on the knowledge and skills within a particular domain, possessing increased domain expertise and practical applicability.

Through "zero sample" or small sample fine-tuning, the large model can achieve better results in a variety of tasks, with a strong generalisation ability, forming a large domain model. Large model training achieves upstream and downstream division of labour, assembly line collaboration, forming a new paradigm of "pre-training + fine-tuning", which brings a new standardized AI research and development logic, realizes the scale production of AI models in a more unified and simpler way, and enhances the performance of large models in different business scenarios.

Domain macromodels are more domain-specific than general-purpose macromodels, and industry macromodels are specifically trained to better understand and process domain-specific knowledge, terminology, and context. Due to the optimisation in a specific domain, the output quality of a vertical domain grand model in that domain is usually

higher than that of a generic grand model. For domain-specific tasks, Vertical Domain Grand Models typically perform better than General Purpose Grand Models.

The classic domain large models for are enumerated in Table 1.

Table 1. Classic domain large models

Name	Domain	Characteristics
BloombergGPT	Financial Large Model	BloombergGPT builds a dataset of 363 billion labels to support various tasks within the financial industry.
FinBERT	Financial Large Model	FinBERT is pre-trained through multi-task learning on financial corpora to transfer knowledge from financial domain corpora.
LaWGPT	Chinese Legal Knowledge Model	LaWGPT constructing a dataset for dialogue and Q&A in the legal field and a dataset for China's judicial examination to conduct fine-tuning of instructions.
BenTsao	Chinese Medical Model	BenTsao integrates the medical knowledge of the Medical Knowledge Graph and fine-tuning it with knowledge-based instruction data.

The versatility and generalization of large models and new development paradigms such as "pre-training + fine-tuning" make the model customization process of AI scenario application more standardized, the effect optimization more simple, effectively reduce the ability requirements for data annotation and algorithm optimization, and make AI application research and development more convenient.

2.2. Research Progress in RBR

Definition of RBR RBR is a way to represent the empirical knowledge of experts in a certain domain in the form of rules, and the representation of these rules contains the problem and the solution to the problem, and the use of rule knowledge to simulate the reasoning process used by experts to solve a certain type of a certain type of problem [23]. RBR abstracts rule-based knowledge, encompassing expert experience, computational formulas, design specifications, etc., into symbolic, normative, and specific generative rules. These rules are then matched with the facts, ultimately leading to conclusions. Its reasoning process is shown in Fig. 1.

The advantages of RBR are: intuitive and natural, easy to understand, easy to computer reasoning; rules have the same format, easy to unify the processing; can effectively express the surface knowledge. But at the same time, RBR has certain disadvantages: the relationship between the rules is not obvious, resulting in low processing efficiency, management and maintenance is relatively difficult; cannot represent the structural knowledge; the rule base is large in size, the efficiency is not high, generally applicable to small-scale reasoning; at the same time, the rules of the acquisition of relative difficulties.

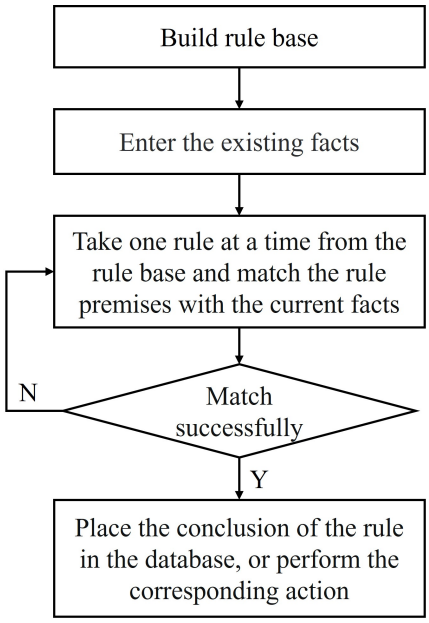


Fig. 1. RBR workflow

Different Technological Routes The implementation process of RBR has three main steps, firstly, the user input facts are matched with the antecedent part of the generative rule, if the match is successful, then this rule is selected. If more than one rule matches successfully at the same time, then it is processed by the conflict cancellation mechanism, and the appropriate rule is selected for inference; the matching operation is repeated in a cycle. The key technologies of RBR are mainly knowledge acquisition and its inference mechanism.

1. **Knowledge acquisition** is the process of delineating the scope of reasoning, the process of collecting and processing knowledge, and the basis for constructing a knowledge base, so this stage of the process is crucial. Knowledge acquisition mainly collects problem-related knowledge from industry experts or specific rules. The way of knowledge acquisition is usually manual, automatic and hybrid acquisition, the manual way is too labor-intensive although the integrity is high; on the contrary, the automatic way is the application of computer technology, compared to the former, it is more efficient and intelligent, so it is widely used in various industry sectors. Hybrid acquisition is a combination of the first two methods, with the ability to identify text, images, language, but also the ability to complete the analysis, understanding and induction, and self-learning from practice, which not only improves the completeness and accuracy of knowledge acquisition, but also ensures the efficiency of reasoning.

2. **Reasoning mechanisms** The commonly used reasoning algorithms are, forward reasoning, backward reasoning, and combined forward and backward reasoning.

Forward reasoning is the use of user-provided factual information as the basis for reasoning, and the positive use of rules for reasoning in a million ways, also known as data-driven reasoning or pre-necklace reasoning. Its basic process is based on known information as a starting point for the reasoning process. That is, the basic structure of the method is mainly divided into two parts: the premise and the conclusion. $P \Rightarrow Q$ or IF P THEN Q Forward reasoning takes the problem as a starting point, i.e., the process conditions are used as a starting point, and then the solution is reasoned out. However, forward reasoning generally has the disadvantages of low efficiency and lack of flexibility.

Reverse reasoning and forward reasoning are different, reverse reasoning and forward reasoning starting point is not the same, reverse reasoning is the conclusion of the problem as a starting point for reasoning, can also be called reverse chain reasoning or goal-oriented reasoning. The reasoning process is to take the hypothesis of the goal for the starting point of the reasoning process, according to the hypothesis of the goal to find information related to the hypothesis, if you find relevant information on the original hypothesis, the original hypothesis is established. If no information can be found to make the original hypothesis clear, then the original hypothesis is not valid, and the hypothesis target needs to be re-selected.

Forward and reverse reasoning is also known as mixed reasoning. This type of reasoning is a combination of forward and reverse reasoning, which make up for each other's deficiencies. Mixed reasoning is to use the conclusion of the result of the forward reasoning as the starting point of reasoning, and then reason out the result after the starting point has been determined. A reasoning goal is arrived at based on forward reasoning, and then the goal is confirmed based on reverse reasoning.

In the embryonic and developing stage of RBR technology, RBR related research mainly focuses on knowledge representation of rules and fast pattern matching algorithm of inference. During this period, many excellent fast pattern matching algorithms have emerged. Currently, the main pattern matching algorithms are Linear Algorithm, Treat Algorithm, Leaps Algorithm, and Rete Algorithm. Among them, Rete algorithm is the most widely accepted and used. The Rete algorithm was proposed by Charles Forgy in 1978. Although Rete algorithm improves the execution speed of RBR, its performance is still not enough to cope with massive data environment and high frequency inference applications.

In terms of the generalization of RBR technology, many open-source organizations, research institutions and manufacturers have launched their own products that integrate RBR technology into a complete system scheme or a separate system, such as Java rules engine standard JSR-94, based on Microsoft. NET platform BizTalk system business rules engine, JBoss Drools, open-source organization Apache Jena project, JESS and ILog JRules, etc. These rule engine products usually implement a variety of inference algorithms of RBR, but they still use rule-based knowledge representation, which is still limited in knowledge acquisition.

2.3. Research Progress in CBR

Definition of CBR CBR technique is a method of solving currently encountered problems based on the knowledge of past cases or experience gained from past cases, which is more suitable for solving problems that are difficult to establish theoretical models or complex problems. CBR is a kind of analogical reasoning method, which provides a

new methodology to build an expert system that approximates the human thinking model, which is consistent with the solution of natural problems by human beings. When solving problems, human beings often recall the treatment of similar situations accumulated in the past and solve new problems by appropriate modification of the treatment of similar situations in the past. Past situations and their treatment are called cases. Cases can help form solutions to new problems and can be used to prevent possible errors. In 1994, A.P [1] proposed the 4R model of CBR (Retrieve-Reuse-Revise-Retain), and since then the 4R model of CBR has been widely disseminated. In the 4R model, historical cases that have been successfully solved in the past are stored in a case base; when a new problem arises, the decision is made by extracting the attributes of the current problem, using the pre-set retrieval algorithms of the CBR system, retrieving (Retrieve) in the case base, and obtaining one or a number of historical cases that are the most similar to the new problem, with the corresponding solution as the recommended The decision maker decides whether to reuse the recommended solution as the solution to the current new problem according to the actual situation; if the retrieved cases are far away from the target problem and the decision maker is not sure about adopting them, the revision (Revise) part of the model can be executed, and the retrieved solutions are revised using a certain method to make them meet the requirements of the current problem; finally, the current target Finally, after the new problem is solved by the revised solution, the target problem and the solution to the target problem are stored as a new case in the case base (Retain). The process of CBR is shown in Fig. 2.

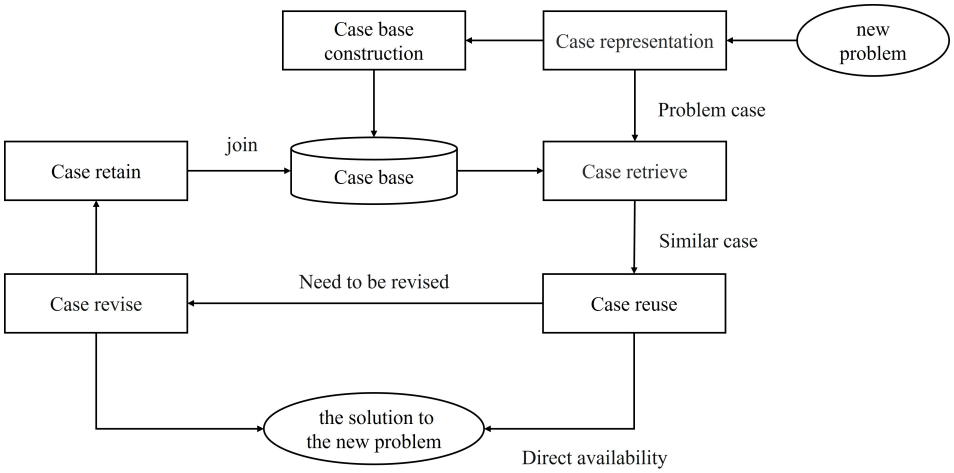


Fig. 2. CBR workflow

CBR does not require a detailed understanding of the principles of the new problem, and has the advantages of easy access to experience and knowledge, efficient problem-solving efficiency, and continuous updating of the case base to improve the accuracy of

the solution. These related theories and techniques provide references and guidance for the study of case-based intelligent reasoning process in CAD software.

For example, in the field of static equipment, to apply CBR for drawing reuse, the process is as follows: A case base is constructed according to the existing static equipment design information and drawings, the upstream conditions are received, and similar static equipment is obtained through case retrieval. The platform automatically modifies the design parameters, and the applicable static equipment drawing cases are obtained. Users can modify other data according to additional requirements and deliver the modified drawing cases to the case library for subsequent use.

Different Technical Routes According to the CBR working process, it can be seen that the system mainly includes four key technologies [43]: case representation, retrieval, modification and learning. The more mature research is the retrieval of instances, including the nearest neighbour retrieval method, the retrieval method of frames, etc., and the modification of instances is currently the focus of CBR's research, which focuses on the following: the indexing of instances, the retrieval, the amendment, and the method of acquiring the amendment rules, the maintenance technology of the case base, and the evaluation about its performance.

1. **Case Representation** The case representation is to store the case in a way that the computer can identify and process, and the case representation has three main aspects: first, the background or reason for the event and the specific content of the problem; second, the characteristics of the case, the process of solving the problem; and third, the method and effect of solving the event. Through the case representation, it is possible to quickly retrieve cases that are similar to the target case. Commonly used case representation methods are framework representation, object-oriented representation, and semantic representation.

2. **Case Retrieve** The so-called case search, in essence, is to be given in the user after a case to be examined, the retrieval system can automatically from the case database (DB), looking for cases with the user case exactly the same or part of the same case; and the output results can be in accordance with the degree of compliance with the user's requirements for sorting, in line with the question of the high degree of priority output.

The current indexing algorithms for cases are nearest neighbour strategy, inductive reasoning strategy and knowledge guide strategy. The nearest neighbour strategy is a method that calculates the similarity between the problem representation and the case to find the most similar case. The nearest neighbour strategy is the most commonly used method.

3. **Case Revise** The examples retrieved in the example database through certain search methods are not necessarily fully applicable to the new problem, so the retrieved source examples have to be processed to adapt to the new problem, which involves the modification of the examples. Example modification is to use the source example as a reference for the target example and modify the part of the source example that is not applicable to the new problem [8] to solve the new problem. Case modification does not change the data of the source instance in the instance repository, but produces a new instance. Commonly used techniques for case modification are replacement, conversion, and derived analogy [33].

4. **Case Study** As new problems are continuously solved, the case base should have some learning capability [42] to facilitate the organisation and management of the case base and to improve the representativeness of the retrieved cases. Case learning mainly consists of adding new cases and deleting existing cases.

CBR technology has been widely studied and applied. Typical application cases include: Vilhena et al. applied CBR to predict the tendency of thrombosis; Erbacher et al. applied CBR to network early warning report; Horsman et al. applied CBR to evidence location in digital forensics.

2.4. Characteristic Analysis of RBR and CBR

RBR and CBR differ greatly in the following aspects:

1. A rule describes a function or pattern, while a case is a constant.

2. The rules of RBR need to match exactly, while the cases of CBR only need to match partially.

3. RBR adopts cyclic iteration of small events and deduces step by step; The CBR estimates the entire solution by analogy at one time and then adjusts the final solution.

4. Strict form of rules, difficult to obtain; Case form freedom, access to arbitrary.

5. RBR is based on deductive reasoning, while CBR is based on analogical reasoning.

As two concrete forms of knowledge reasoning, RBR and CBR have commonalities in the following aspects:

1. Rules and cases are different conceptual models of knowledge, but can be based on the same data structure or implementation technology in a concrete computer implementation.

2. The essence of the rules and cases is a relational mapping of the domain ontology and its attributes, and both reveal a division of the attributes and states of the domain ontology.

3. The rule base and case base in the same domain correspond to the state space of the same domain ontology.

It can be found that RBR and CBR are highly complementary, mainly reflected in:

1. RBR is good at expressing qualitative relations, while CBR is good at expressing quantitative relations.

2. RBR data maintenance is difficult, while CBR data maintenance is simple.

3. RBR reasoning speed is slow, CBR reasoning speed is faster.

4. RBR has strong interpretation ability, while CBR has weak interpretation ability.

5. The solving ability of RBR edge problem is weak, while the solving ability of CBR edge problem is strong.

It can be seen from the comparison that RBR and CBR have their own advantages, so we should consider combining RBR and CBR for reasoning, so as to bring better accuracy and efficiency of reasoning system.

3. Research Methodology

In this chapter, we outline our search methodology and inclusion criteria, which form the foundation for selecting relevant studies. Following that, we present a general analysis of the research content, offering an overview of the themes and patterns emerging from the studies we have selected.

3.1. Search Methodology and Inclusion Criteria

We searched the published literature on reasoning or large model distribution up to 2023 on Google Scholar, Web of Science and Scopus, using the following keywords:

- CBR, RBR, case-based reasoning, rule-based reasoning, mixed reasoning
- Artificial intelligence generated content (AIGC), large model, big model, domain large model
- CAD software, industry large model, reusable design, case intelligent recommendation

References to the retrieved articles are also within the scope of the search, determine which articles seem relevant, find those articles, read their references, and then repeat the process until there are no new relevant articles.

The following types of literature were excluded: research reviews or progress reports, non-inference or non-large model research, studies focusing on a single model only, and research on large models used for purposes other than those specified. Additionally, literature that solely relied on previous reasoning methods without introducing new analytical approaches was not considered.

3.2. General Analysis of the Research Content

The literature analysis process proposed in this paper starts from the preliminary reading and screening of the literature, and extracts the obvious or possibly relevant parts of the literature, so as to form a general understanding of the research content of reasoning and large models. Preliminary reading shows:

(1) There is a large literature with multiple research purposes, and in some literature different authors use different phrasing to express similar concepts. Examples include big models and large models.

(2) In recent years, researches on large models of artificial intelligence have emerged in many fields and brought different impacts to the fields.

(3) In the field of reusable design, there is almost no relevant research on domain large models.

Based on the above three findings, this paper analyzes the influence of artificial intelligence large model on reusable design CBR and RBR, and proposes a recommended framework combining artificial intelligence large model in view of the areas that can be improved in reusable design..

4. Results and Discussion

This chapter analyzes the influence of AI large models on reusable design, CBR, RBR, and proposes a framework for reusable design based on AI large model.

4.1. Impact of AI Large Models on Reusable Design

Industrial Internet is centred on the comprehensive linkage of the whole industrial chain, the whole value chain and the whole elements, building a new ecology of new-generation information technology empowering the manufacturing industry, emphasizing the interconnection and interoperability of the massive production elements, the value mining of operational data and the precipitation and reuse of industrial knowledge, which provides "natural soil" for the application of the large model.

In the era of large models, computing power, network and data constitute the "iron triangle" of the underlying infrastructure. Large models with large computing power, large algorithms and large data features can be used to further optimise the Industrial Internet and promote the reuse of design knowledge for solutions. Large models can assist in analysing large datasets through machine learning algorithms to identify patterns and correlations that may not be readily visible to researchers. Secondly, AI can analyse existing scientific literature and generate hypotheses that can be tested in further research, which can help identify new reusable design methods. In addition, the emergence of large models largely solves the problem that traditional models/services do not work well for cross-modal and cross-domain applications. The MaaS layer rooted on the industrial Internet platform can directly provide high-quality large model services for user terminals in a variety of scenarios. Enterprises can process and train the data by scheduling the relevant APIs and based on the business scenarios of a specific solution, thus reducing the development and application costs of enterprises and realising the deployment, optimisation, and upgrading of enterprises' personalised application businesses. The large model can improve language comprehension and image generation capabilities, invoke model microservices in the R&D design process, help R&D personnel to accurately mine and sort out effective basic knowledge, generate application-specific basic code or carry out three-dimensional visualisation design, as well as establish an intelligent industrial knowledge base and so on.

4.2. Impact of AI Large Models on RBR and Existing Work

The whole process of RBR includes: knowledge acquisition and reasoning mechanism. Now there are many ways to implement inference for various tasks based on large models, such as:

Richardson [32] provides a survey of recent research in conversational AI with a focus on commonsense reasoning. The paper lists relevant training datasets and describes the main approaches to incorporating common sense into conversational AI. The paper also discusses benchmarks used to evaluate common sense in conversational AI problems. Finally, the authors make initial observations about the limited common-sense capabilities of two state-of-the-art open dialogue models, BlenderBot3 and Lambda, and their negative impact on natural interactions. These observations further advance the study of common-sense reasoning in conversational AI.

Large pre-trained Transformer-based language models such as BERT, GPT and T5 have demonstrated a deep understanding of contextual semantics and language syntax. Their achievements have led to notable progress in conversational AI, fostering the creation of open dialogue systems proficient in maintaining coherent and relevant conversations. These systems can address queries, engage in casual chats, and accomplish various tasks.

In terms of knowledge acquisition, the use of large models for knowledge acquisition is much better than the original automatic acquisition. In order to express the structure of the knowledge base, knowledge graphs can be constructed, knowledge graphs using graph structure can present more complex relationships, the stored information is more full, and the recommended cases are definitely more relevant . Generic large models can use a model to combine multiple knowledge acquisition rules, the rules themselves need to be refined. For example: the height of the tower and the material of the tower construction, they themselves need two sets of rules to be obtained, but based on the large model, it is possible to solve these two tasks or even more with one model.

In terms of reasoning, when the information embedded in the knowledge base is sufficiently adequate, the large model can examine the results of rule reasoning and give inference results, or even automatically fine-tune the rules to meet the current usage scenarios and directly achieve reuse. Example: Code Inspection and Code Correction.

4.3. Impact of AI Large Models on CBR and Existing Work

The retrieval of cases belongs to an important part of CBR, and from the historical experience, traditional retrieval will face the following challenges:

1. Rely on the network. Because information retrieval models do not retain knowledge or information themselves, they rely on the Internet for external knowledge, which may limit their applicability in some scenarios.

2. Lack of reasoning skills. The existing IR model mainly provides the collected knowledge/information to meet the information needs of human beings, but lacks the ability to help users understand the information. Better reasoning will lead to friendlier and more valuable outcomes for humans.

Breakthroughs in Information Retrieval after Combining Large Techniques:

As the scale of large language models continues to grow (both in terms of model size and data volume), LLMs have demonstrated significant advances in their capabilities. On the one hand, LLMs have also made unprecedented breakthroughs in language comprehension and generation, resulting in more humane and human-intended responses. On the other hand, larger LLMs are more capable of generalising and reasoning when dealing with complex tasks [45]. Notably, LLMs can effectively apply their learned knowledge and reasoning abilities to solve new tasks with only some task-specific demonstrations or appropriate instructions [27, 44]. Furthermore, advanced techniques such as contextual learning significantly enhance the generalisation performance of LLMs without the need for fine-tuning for specific downstream tasks [3]. This breakthrough is particularly valuable as it reduces the need for extensive fine-tuning while achieving superior task performance. Supported by cueing strategies such as thought chaining, LLMs can generate output through step-by-step reasoning to guide complex decision-making processes [46]. By integrating these complex language models, IR systems can provide more accurate responses to users, ultimately reshaping the landscape of information access and retrieval.

The potential of LLMs for information retrieval has been initially explored, and in terms of practical applications, New Bing aims to improve the user's experience of using a search engine by extracting information from different web pages and compressing it into concise summaries in response to user-generated queries. In the research community,

LLMs have been shown to be useful in specific modules of information retrieval (e.g., searchers), thus improving the overall performance of these systems.

Recently, dense retrieval has been extensively studied and approaches based on standard MIPS indexing and nearest neighbour search are common [12]. Given the success of Transformers as good associative memory stores or search indexes, Tay et al [38] proposed a novel architecture called micro-searchable indexing, where the indexes are stored in the model parameters.

The development of multimodal LLMs will facilitate indexing systems capable of indexing data in various modalities, including but not limited to text, images, and video, in a unified manner [2].

With the widespread use of LLMs in information retrieval, tailored evaluation strategies become essential to prove the effectiveness of LLMs. For this purpose, some properties such as robustness need to be emphasised. Many models are sensitive to distributional differences between training and test data [24]. It is eager to see the generalisation of LLM-enhanced IR models to out-of-distribution scenarios.

4.4. A Reusable Design Recommendation Framework Based on AI Large Model

Mixed CBR and RBR CBR for the requirements of the domain knowledge model is relatively broad, to facilitate the acquisition of domain knowledge, but there are certain drawbacks, stays on the surface of the empirical knowledge, the lack of deep knowledge, professional knowledge, the need for human participation, the lack of a rigorous theoretical foundation will be difficult to achieve the true meaning of AI. RBR relies too much on the experience of experts, so it does not have good self-learning ability, with the expansion of the knowledge base will easily be control saturation problem, not a good simulation of the judgmental thinking ability of the domain experts, the system will have a decline in reasoning and other problems.

Combining the characteristics of RBR and CBR, it can be seen that the advantages and disadvantages of CBR and RBR can be complementary. Previous studies have mostly used a single RBR or a single CBR for knowledge-based reasoning, but recent studies have often used a combination of the two types of reasoning, resulting in enhanced reasoning effects. Nowadays, there are four main types of reasoning fusion: RBR is mainly CBR, CBR is mainly RBR, CBR and RBR reasoning in parallel, and CBR and RBR are deeply fused.

1. RBR is dominated by CBR

This method is mostly applied in the case of small size of the case base, the first use of RBR reasoning, when the appropriate results can not be obtained, the choice of CBR to find the past cases for case retrieval.

2. CBR is the main RBR

This method is mostly applied in the case of a large case base, firstly, when using CBR case retrieval, the RBR rule is used to calculate the similarity in order to find similar cases, or for CBR retrieved cases using RBR for case modification and case evaluation. Most of the current fusion reasoning uses this approach

3. CBR and RBR Parallel Reasoning

Parallel reasoning is not considered to be a true fusion, where the two reasoning modalities operate in parallel to obtain separate results, and the two results are synthesised as the final result.

4. Deep integration of CBR and RBR

Deep fusion of CBR and RBR refers to the fusion of reasoning methods in all aspects of knowledge reasoning, which is more difficult to achieve, and thus less relevant research is conducted, generally using a combination of RBR and CBR for data analysis, RBR for generating recommendations or selecting solutions, and the same combination of RBR and CBR for case modification.

Recommended Framework After the study of two kinds of inference rules, CBR and RBR, it is found that the fusion of the two can improve the inference efficiency, so this paper proposes to apply the combined inference rules of CBR and RBR to the reuse design in CAD software. The function of RBR is to retrieve similar cases (i.e., CBR inference rules) according to a certain algorithm, and then the inference rules established by RBR are integrated into the CBR cases, and the combination of them This combination not only makes the system more efficient, but also helps to improve the use of knowledge.

The essence of large language model is a parameterised knowledge base, the main feature of parameterised knowledge base is to decompose the knowledge into a combination of parameters and rules, and generate specific knowledge instances by modifying the parameter values, which is a black-box and non-transparent process, which leads to some seemingly reasonable but absurd assertions that often appear in the natural language generation of large language model. Knowledge graph is a formal representation of knowledge, which inherently has the advantage of strong interpret ability. Therefore, the introduction of knowledge graph in case representation can help to improve the interpret ability of large language models.

Combined with the above technology we propose a set of case reuse design process, first of all, data for including text, pictures and other types of multimodal storage mode of data, and then combined with the powerful knowledge extraction capabilities of the large model to extract all kinds of services required by the value of the data; extracted to the data to build a knowledge base, which can rely on traditional relational databases, but also can be referred to the graph of the data in order to store more structure of the information, enhance the model reasoning ability in the process; knowledge retrieval intervenes in multimodal retrieval technology, aligning all kinds of data information to improve the retrieval efficiency; knowledge integration is the classification and summary of all the data in the knowledge base to organise the work, which helps to manage the knowledge base; at the same time, the knowledge base can't avoid modification, and the knowledge learning module realises the operations such as error identification, error correction, de-emphasis and updating; combining knowledge retrieval technology and grouping and aggregation Combined with knowledge retrieval technology and group aggregation tasks and recommendation algorithms, reusable programme recommendation can be carried out; the methods in knowledge learning can be empowered to the programme modification design module. The flow is shown in Fig. 3.

For further exploration, we propose a framework for collaborative LLMs, KGs and DBs, which consists of five layers, data, collaborative models, technologies, services and applications. In the data layer, textual and structural data can be stored with various types of DBs and KGs, and with the development of multimodal LLMs and KGs, the framework can be extended to handle multimodal data, such as images. In the collaborative modelling layer, LLMs, KGs and DBs can coordinate with each other to improve their respective

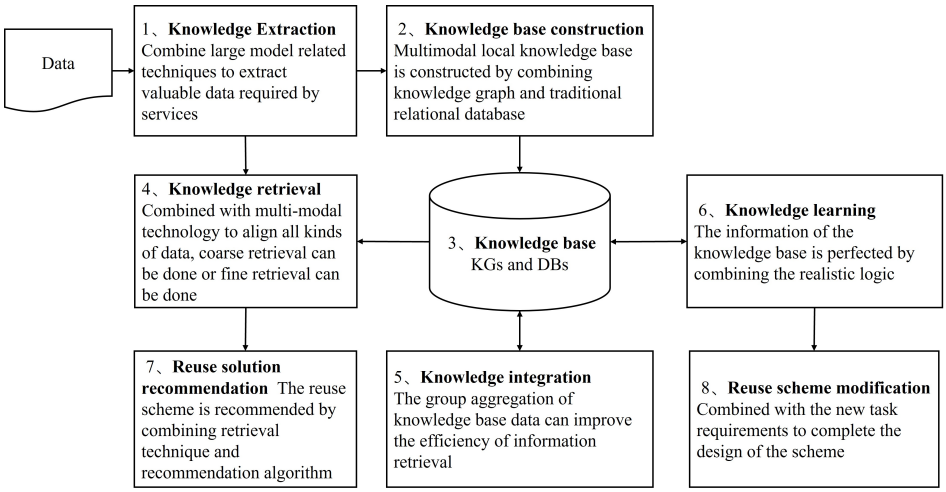


Fig. 3. Case reuse design flow

capabilities, and LLMs can fill in the missing data in KGs and DBs; in the pre-training phase, incorporating KGs and DBs into LLMs can help LLMs to learn the knowledge from the knowledge base. In the technique layer, relevant technologies used in LLMs and KGs can be incorporated into the framework, such as graph neural networks, few-shot learning and reinforcement learning, to further improve performance. At the service layer, related techniques can provide better solutions to the problems of case reuse, case modification, and rule representation in hybrid CBR-RBR. At the application layer, LLMs and KGs and DBs can be integrated to construct application products for large model-enabled reusable design. The framework is shown in Fig. 4.

The two most critical aspects of reuse design are information extraction and information alignment. We determine whether the current case is suitable for reuse through the results of alignment. There are already mature cases in information extraction. For example, TechGPT can extract domain terms, identify named entities, and extract relational triples, all of which are key technologies for knowledge graph construction. With the extracted information, we can get the case characteristics and lay the foundation for the subsequent work; GPT-4 has been able to perform multi-modal tasks, and it is also helpful for feature extraction of case images. In terms of information alignment, the large model has strong reasoning ability and generalization ability, and can clearly distinguish whether the extracted features are similar or not, and whether they are recommended cases. For example, in the reuse of drawings, the existing data include drawing design information, two-dimensional drawings and three-dimensional models. The key features of multi-modes can be extracted with the help of large models, and after obtaining similar drawings, the information can be aligned to determine whether the two equipment drawings are similar, which provides convenience for the reuse design of drawings.

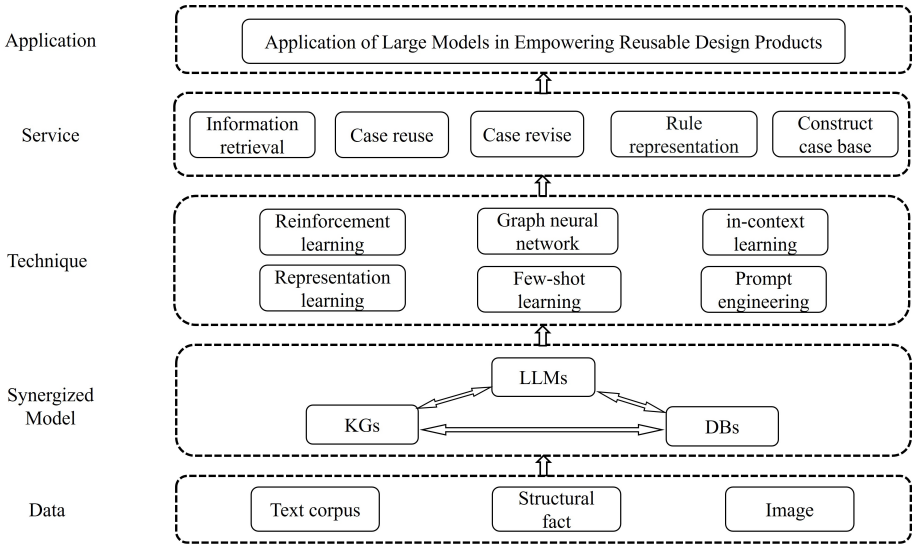


Fig. 4. Framework for co-operating with LLMs, KGs and DBs

4.5. Limitations and Future Work

This paper puts forward some new ideas and methods, but there are some limitations and need to be improved in the future. Future work needs to further explore how to apply these ideas and methods in practical scenarios to improve the efficiency and accuracy of the design.

1. Multimodal data processing. The knowledge base proposed in this paper can be extended to handle multimodal data, such as images. However, this paper does not give a specific scheme on how to deal with multimodal data. Future work could further explore how multimodal data can be processed and incorporated into existing knowledge bases.

2. The problem of large model enabling entity alignment tasks. In the future, we can make use of existing entity alignment methods and make appropriate improvements, give full play to the powerful generalization and reasoning ability of large language models, and effectively improve the accuracy and efficiency of entity alignment of commodity knowledge graph.

5. Conclusion

This review synthesizes recent progress in rule-based and case-based reasoning techniques for reusable design in CAD software systems and summarizes the advantages and disadvantages of CBR and RBR. The complementary strengths and weaknesses of RBR and CBR highlight the potential of hybrid CBR-RBR approaches. RBR effectively represents structured expert knowledge as rules but lacks learning capabilities and flexibility. CBR provides a framework for experience reuse from prior cases but relies heavily on

surface similarities without deeper reasoning. The emergence of large language models and knowledge graphs in artificial intelligence research offers new opportunities to overcome these limitations. Advanced language comprehension and reasoning abilities enable large models to better acquire domain knowledge, retrieve contextual cases, explain connections, and adapt reasoning. Integrating them with structured knowledge graphs also improves interpret ability.

The proposed reusable design framework combines RBR, CBR, large models, and knowledge graphs into a hybrid approach. This allows leveraging the strengths of both reasoning techniques while mitigating their weaknesses through large model augmentations. Such a hybrid framework paves the path toward more efficient, innovative, and sustainable CAD reuse design. Intelligent case-based and rule-based reasoning empowered by AI advances could transform engineering design by fully utilizing prior knowledge. Further research is still needed to realize the potential benefits of reasoning-based CAD systems enhanced by large models. But this review highlights promising directions and a proposed framework to guide future efforts. Overall, the synergistic integration of reasoning techniques with modern AI can fundamentally reinvent reusable design, resulting in greater automation, creativity, and intelligence in computer-aided engineering.

References

1. Aamodt, A., Plaza, E.: Case-based reasoning: Foundational issues, methodological variations, and system approaches. AI communications 7(1), 39–59 (1994)
2. Baltrušaitis, T., Ahuja, C., Morency, L.P.: Multimodal machine learning: A survey and taxonomy. IEEE transactions on pattern analysis and machine intelligence 41(2), 423–443 (2018)
3. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. Advances in neural information processing systems 33, 1877–1901 (2020)
4. Chowdhary, K., Chowdhary, K.: Natural language processing. Fundamentals of artificial intelligence pp. 603–649 (2020)
5. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
6. Eisenstein, J.: Introduction to natural language processing. MIT press (2019)
7. Fagin, R., Halpern, J.Y., Moses, Y., Vardi, M.: Reasoning about knowledge. MIT press (2004)
8. Hassanien, A.E., El-Bendary, N., Sweidan, A.H., Mohamed, A.E.k., Hegazy, O.M.: Hybrid-biomarker case-based reasoning system for water pollution assessment in abou hammad sharkia, egypt. Applied Soft Computing 46, 1043–1055 (2016)
9. Hill-Yardin, E.L., Hutchinson, M.R., Laycock, R., Spencer, S.J.: A chat (gpt) about the future of scientific publishing. Brain Behav Immun 110, 152–154 (2023)
10. Iyer, N., Jayanti, S., Lou, K., Kalyanaraman, Y., Ramani, K.: Three-dimensional shape searching: state-of-the-art review and future trends. Computer-Aided Design 37(5), 509–530 (2005)
11. Jackson, P.C.: Introduction to artificial intelligence. Courier Dover Publications (2019)
12. Karpukhin, V., Oğuz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., Yih, W.t.: Dense passage retrieval for open-domain question answering. arXiv preprint arXiv:2004.04906 (2020)
13. Keshari, R., Ghosh, S., Chhabra, S., Vatsa, M., Singh, R.: Unravelling small sample size problems in the deep learning world. In: 2020 IEEE Sixth International Conference on Multimedia Big Data (BigMM). pp. 134–143. IEEE (2020)
14. Kim, H., Yeo, C., Cha, M., Mun, D.: A method of generating depth images for view-based shape retrieval of 3d cad models from partial point clouds. Multimedia Tools and Applications 80, 10859–10880 (2021)

15. Kolodner, J.: Case-based reasoning. Morgan Kaufmann (2014)
16. Kryściński, W., McCann, B., Xiong, C., Socher, R.: Evaluating the factual consistency of abstractive text summarization. arXiv preprint arXiv:1910.12840 (2019)
17. Kumar, K.A., Singh, Y., Sanyal, S.: Hybrid approach using case-based reasoning and rule-based reasoning for domain independent clinical decision support in icu. Expert Systems with Applications 36(1), 65–71 (2009)
18. Lee, M.D., Pincombe, B., Welsh, M.: An empirical evaluation of models of text document similarity. In: Proceedings of the annual meeting of the cognitive science society. vol. 27 (2005)
19. Liu, G.K.M.: Transforming human interactions with ai via reinforcement learning with human feedback (rlhf) (2023)
20. Lupinetti, K., Giannini, F., Monti, M., Pernot, J.P.: Multi-criteria retrieval of cad assembly models. Journal of Computational Design and Engineering 5(1), 41–53 (2018)
21. Lupinetti, K., Giannini, F., Monti, M., Pernot, J.P.: Content-based multi-criteria similarity assessment of cad assembly models. Computers in Industry 112, 103111 (2019)
22. Maher, M.L., Balachandran, M.B., Zhang, D.M.: Case-based reasoning in design. Psychology Press (2014)
23. Mariani, L., Lussi, Y.C., Vandamme, J., Riveiro, A., Salcini, A.E.: The h3k4me3/2 histone demethylase rbr-2 controls axon guidance by repressing the actin-remodeling gene wsp-1. Development 143(5), 851–863 (2016)
24. Mitra, B., Craswell, N.: Neural models for information retrieval. arXiv preprint arXiv:1705.01509 (2017)
25. Miyazaki, T., Hotta, Y., Kunii, J., Kuriyama, S., Tamaki, Y.: A review of dental cad/cam: current status and future perspectives from 20 years of experience. Dental materials journal 28(1), 44–56 (2009)
26. Nalepa, G.J., Bobek, S.: Rule-based solution for context-aware reasoning on mobile devices. Computer Science and Information Systems 11(1), 171–193 (2014)
27. Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al.: Training language models to follow instructions with human feedback. Advances in Neural Information Processing Systems 35, 27730–27744 (2022)
28. Panda, D.K., Ray, S.: Approaches and algorithms to mitigate cold start problems in recommender systems: a systematic literature review. Journal of Intelligent Information Systems 59(2), 341–366 (2022)
29. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al.: Improving language understanding by generative pre-training (2018)
30. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al.: Language models are unsupervised multitask learners. OpenAI blog 1(8), 9 (2019)
31. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. The Journal of Machine Learning Research 21(1), 5485–5551 (2020)
32. Richardson, C., Heck, L.: Commonsense reasoning for conversational ai: A survey of the state of the art. arXiv preprint arXiv:2302.07926 (2023)
33. Rintala, L., Leikola, M., Sauer, C., Aromaa, J., Roth-Berghofer, T., Forsén, O., Lundström, M.: Designing gold extraction processes: Performance study of a case-based reasoning system. Minerals Engineering 109, 42–53 (2017)
34. Sanderson, K.: Gpt-4 is here: what scientists think. Nature 615(7954), 773 (2023)
35. Sivaloganathan, S., Shahin, T.: Design reuse: an overview. Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture 213(7), 641–654 (1999)
36. Stark, R., Grosser, H., Beckmann-Dobrev, B., Kind, S., Collaboration, I., et al.: Advanced technologies in life cycle engineering. Procedia CIRP 22, 3–14 (2014)
37. Svyatkovskiy, A., Zhao, Y., Fu, S., Sundaresan, N.: Pythia: Ai-assisted code completion system. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. pp. 2727–2735 (2019)

38. Tay, Y., Tran, V., Dehghani, M., Ni, J., Bahri, D., Mehta, H., Qin, Z., Hui, K., Zhao, Z., Gupta, J., et al.: Transformer memory as a differentiable search index. Advances in Neural Information Processing Systems 35, 21831–21843 (2022)
39. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al.: Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971 (2023)
40. Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al.: Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288 (2023)
41. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems 30 (2017)
42. Venugopalan, S., Ariga, P., Aggarwal, P., Viswanath, A.: Case report: Magnetically retained silicone facial prosthesis. Nigerian journal of clinical practice 17(2), 260–264 (2014)
43. Vilhena, J., Vicente, H., Martins, M.R., Grañeda, J.M., Caldeira, F., Gusmão, R., Neves, J., Neves, J.: A case-based reasoning view of thrombophilia risk. Journal of biomedical informatics 62, 265–275 (2016)
44. Wei, J., Bosma, M., Zhao, V.Y., Guu, K., Yu, A.W., Lester, B., Du, N., Dai, A.M., Le, Q.V.: Finetuned language models are zero-shot learners. arXiv preprint arXiv:2109.01652 (2021)
45. Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D., Metzler, D., et al.: Emergent abilities of large language models. arXiv preprint arXiv:2206.07682 (2022)
46. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q.V., Zhou, D., et al.: Chain-of-thought prompting elicits reasoning in large language models. Advances in Neural Information Processing Systems 35, 24824–24837 (2022)
47. Williams, D.W., Wood, M.S.: Rule-based reasoning for understanding opportunity evaluation. Academy of Management Perspectives 29(2), 218–236 (2015)
48. Xu, M., David, J.M., Kim, S.H., et al.: The fourth industrial revolution: Opportunities and challenges. International journal of financial research 9(2), 90–95 (2018)
49. Yao, Z., Aminabadi, R.Y., Ruwase, O., Rajbhandari, S., Wu, X., Awan, A.A., Rasley, J., Zhang, M., Li, C., Holmes, C., et al.: Deepspeed-chat: Easy, fast and affordable rlhf training of chatgpt-like models at all scales. arXiv preprint arXiv:2308.01320 (2023)
50. Zhu, J., Rizzo, J.R., Fang, Y.: Learning domain-invariant feature for robust depth-image-based 3d shape retrieval. Pattern recognition letters 119, 24–33 (2019)

Yunlei Sun received Ph.D. degree in engineering from Beijing University of Posts and Telecommunications, Beijing, China, in 2014. He is currently an Associate Professor with the College of Qingdao Institute of Software, College of Computer Science and Technology, China University of Petroleum (East China), Qingdao, China. His current research interests include machine learning and intelligent design.

Bingyi Yan received B.E. degree in computer science and technology from Qingdao University, Qingdao, China, in 2021. She currently is pursuing her master's degree in computer technology at the College of Qingdao Institute of Software, College of Computer Science and Technology, China University of Petroleum (East China), Qingdao, China. Her research interests include intelligent design and artificial intelligence.

Zhaotong Shao received B.E. degree in computer science and technology from Qingdao Agricultural University, Qingdao, China, in 2021. He currently is pursuing his master's

degree in computer technology at the College of Qingdao Institute of Software, College of Computer Science and Technology, China University of Petroleum (East China), Qingdao, China. His research interests include intelligent design and machine learning.

Received: September 07, 2023; Accepted: April 08, 2024.

Computer Science and Information Systems 21(4):1547–1565 https://doi.org/10.2298/CSIS230907047Y

A Hybrid GA-Powell Algorithm for Geometric Constraint Solving

Sun Yunlei[1] and Li Yucong[1]

Qingdao Institute of Software, College of Computer Science and Technology,
China University of Petroleum (East China)
Qingdao, 266580, China
sunyunlei@upc.edu.cn
s21070036@s.upc.edu.cn

Abstract. Geometric constraint solvers are crucial for computer-aided design (CAD), and their algorithms are the focus of research. Current geometric constraint solvers based on traditional numerical methods lack support for multi-solution problems, so we propose a hybrid algorithm that combines the genetic algorithm, which is good at global convergence, and Powell's method, which is good at local refinement, to address the limitations of traditional numerical methods in geometric constraint solving algorithms (sensitivity to initial values, susceptibility to falling into local optimums, and being only able to obtain a single solution) and the challenges of intelligent optimization algorithms (complex parameter tuning, slow convergence and low accuracy). Our method has a large accuracy improvement over the comparison method in basically all test cases, and its effciency can also meet the needs of real geometric constraint solving scenarios. This research provides new insights into the design of geometric constraint solving algorithms, offers a fresh perspective on improving the performance and generality of solvers, and contributes to technological advances in the CAD field.

Keywords: genetic algorithm, Powell's algorithm, geometric constraint solving, similarity calculation.

1. Introduction

Computer-aided design (CAD) is the most widely used modelling approach for engineering design. The typical starting point in these designs is 2D sketches which can later be extruded and combined to obtain complex three-dimensional assemblies [17]. The geometric constraint solver is a key component in this process, which is usually a key technology in parametric CAD design, and the geometric constraint solver algorithm is the core of the geometric constraint solver. The accuracy and efficiency of the geometric constraint solver algorithm is related to the quality of the final product and the experience of the designer, and the improvement of the geometric constraint solver algorithm has always been an important research topic.

Samy Ait-Aoudia et al [1] classified geometric constraint solving techniques into three categories: algebraic, rule-oriented and graph construction, where algebraic methods include numerical and symbolic methods. Numerical methods are widely used because they are fast and applicable to almost all systems, but numerical methods have their inherent defects, the most representative of which is the Newton-Raphson iterative method, which

is widely used in solving geometric constraints based on numerical methods [10,12,16]. In these studies, the numerical methods converge faster but rely heavily on proper initial value guessing, otherwise the algorithm will not converge or converge to a local optimal solution, and only one root can be found by such methods. Therefore, numerical algorithms can be used directly when dragging the geometry, when the solution is close to the desired result.

In recent years, many scholars have tried to introduce intelligent optimisation algorithms such as particle swarm algorithm, simulated annealing algorithm and genetic algorithm to solve the shortcomings of traditional numerical methods [5,23,6]. In these studies, the intelligent optimization algorithm is relatively more relaxed on the initial value requirements, with global convergence and can find multiple solutions to the problem, but the algorithm's parameter setting is complex, the convergence speed is slower, and the accuracy of the results obtained by the algorithm is poor, which can not meet the requirements of practical engineering use.

In this paper, a hybrid genetic-Powell algorithm (GA-Powell algorithm) combining the advantages of numerical method and intelligent optimization algorithms proposed based on the practical application scenario of chemical static equipment design. The algorithm first uses the global search ability of the genetic algorithm to give rough initial solutions, and then screens the initial solutions by similarity calculation to remove similar initial solutions that are easy to converge to the same exact solution, and then inputs the screened initial solutions into Powell algorithm for exact solution in order to obtain all the solutions that conform to the design accuracy. Compared with the traditional numerical algorithm, GA-Powell algorithm is slower, so it is not suitable to be used directly when the user drags the geometry. Generally, it is used when the geometry is generated according to commands or scripts. The proposed GA-Powell algorithm possesses the following contributions:

- Taking advantage of the genetic algorithm, it makes up for the inherent defects of Powell method that the initial value is sensitive, easy to fall into the local optimal solution, and can only obtain one solution.
- Using the advantages of the Powell method, it makes up for the inherent defects of the genetic algorithm that the parameter setting is complicated, the convergence speed is slow and the accuracy is poor.

The high efficiency and accuracy of the GA-Powell algorithm meet the needs of practical geometric constraint solving scenarios, and the design ideas of the algorithm can help guide the research of geometric constraint solving algorithms, which is conducive to improving the performance and applicability of geometric constraint solvers. In the following sections, we present our research results and analyses.

Section 2 provides an overview of existing geometric constraint solving methods, especially the research based on numerical methods with intelligent optimisation algorithms.

Section 3 presents the research methodology, including how geometric constraint solving problems can be transformed into the optimisation problems, an overview of genetic algorithms and Powell's method, and the design and implementation of a hybrid GA-Powell algorithm.

Section 4 describes our experiments, including the experimental setup, systems of equations for testing, and comparative analyses of number of solutions, solution accuracy and solving time.

In Section 5, we will give some principles of parameter setting based on the analysis of the experimental results in order to make the GA-Powell algorithm perform better in different problem scenarios.

Finally, in Section 6, we summarise our main findings, discuss the contributions and implications of the research, acknowledge the limitations and provide recommendations for future research.

2. Related Work

According to the classification proposed by Samy Ait-Aoudia et al [1], geometric constraint solving methods can be broadly classified into three categories: algebraic methods, rule-oriented methods and graph construction methods. In turn, algebraic methods include numerical and symbolic methods as shown in Fig. 1.

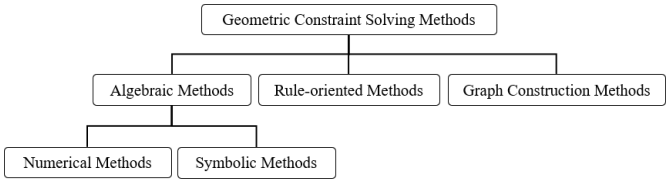


Fig. 1. Classification of geometric constraint solving methods

Numerical methods convert a geometrically constrained system into a set of nonlinear equations that are solved using numerical algorithms. This method, originally introduced by Hillyard [7] at Cambridge University and further developed and improved by Gossard [11] at MIT, is called variational geometry. Although these equations are usually multi-solution, numerical methods usually find only one solution.Borning [2], Hillyard and Braid [8] and Sutherland [21] use relaxation methods. This method interferes with the values assigned to the variables and minimises the measure of global error. Typically, convergence to a solution is slow.

The most widely used method is the Newton-Raphson iteration. It is used in the geometric constraint solving described in Refs [10,12,16]. Newton-Raphson is a local method that converges much faster than relaxation. However, this class of numerical methods requires computing the gradient of the problem to iteratively update the values, and the method is not applicable to systems of equations that are always overconstrained unless special steps are taken, such as solving a least squares problem. It works well if a good approximation of the expected solution is provided and the system is not pathological. Thus, if the starting point is taken from the user's sketch, then the sketch should approximate the expected solution. Numerical methods are generalized and are used as a last resort when other methods do not work well.

Powell's method is a direct search method, which makes full use of the objective function's sex state and constructs the conjugate direction by using the change of the objective function value, and the optimization search method is based on the conjugate direction, which makes it unnecessary to compute the gradient of the objective problem, which greatly saves the amount of computation, and therefore the method has a high convergence efficiency, and it is widely used in the field of optimization [4,9,13]. However, Powell's method still has the disadvantage that traditional numerical methods are sensitive to the initial value and easy to fall into local optimization.

The symbolic method, like the numerical method, transforms a system of geometric constraints into a set of nonlinear equations. However, it uses symbolic algebra to solve these constraints, using methods such as the Grobnacki method [3] or the Wu-Ritt characteristic column method [22]. If symbolic parameters are used in a system of nonlinear equations, the symbolic method can find generalised solutions for geometrically constrained systems, making it a very efficient method. However, the disadvantages of this method are its slowness and its high time and space complexity, and therefore the limitations on the types of geometric elements and constraints that can be used.

The rule-oriented geometric constraint solving method utilizes rules to define and carry out the construction process, hence the name "rule-construction solving method". This approach allows for clear representation of geometric knowledge, separating it from the processing stage, and makes it easy to expand the rule base. However, this method also has its drawbacks: the rules are often incomplete, the system is bulky, the solving speed is slow, and it is unable to solve cyclic constraints.

The graph construction method converts the geometric constraint system into graphs, deduces the construction process by analysing the geometric constraint graphs, and generates geometric shapes based on the construction steps. Currently, several researchers use machine learning training datasets (e.g., SketchGraphs [20]) and frameworks (e.g., SketchGen [17]) to train models to automatically generate sketches, thereby reducing design time and enabling new design workflows. The method is based on graph theory and is theoretically rigorous, fast and efficient. However, it can only solve closed-loop constraints by numerical methods. In addition, it is sensitive to the type of geometric elements and constraints used, thus requiring modification of the solution algorithm when new geometric elements or constraints are added, making this method less versatile.

The summary of geometric constraint solving methods is shown in Table 1. Numerical methods are widely used in practical geometric constraint solving due to their universal applicability, and improving their solving efficiency and stability is the focus of research in this field.

Intelligent optimisation algorithms have entered the scholarly scene in recent years. In this approach the problem is reinterpreted as an optimisation problem and solved using genetic, particle swarm or other evolutionary methods such as [5,23,6]. It is stable and less sensitive to initial value guessing, but its solution efficiency is low compared to traditional numerical methods, which cannot meet the real-time demands of design work, and its solution accuracy is relatively poor, problems that are illustrated in Ref. [24,25].

By comparing the characteristics of traditional numerical methods and intelligent optimisation algorithms, it can be found that the two can complement each other by combining their strengths - the global convergence of intelligent optimisation algorithms can make up for the shortcomings of traditional numerical methods that are sensitive to the

Table 1. Summary of geometric constraint solving methods

Methods		Advantages	Applicable scene
Algebraic	Numerical	Fast, general	Most systems
	Symbolic	Effective, generic solution	Small systems with restricted elements and constraints
Rule-oriented		Separate knowledge and processing, avoid numerical instability	Small systems
Graph Construction		Rigorous theory, fast solving	Systems without new types of elements and constraints

initial value, prone to falling into local optimal solutions and only able to obtain a single solution, whereas the accuracy and high efficiency of traditional numerical algorithms can make up for the slow convergence speed and accuracy of intelligent optimisation algorithms. optimisation algorithm's slow convergence speed and poor accuracy.

Therefore, in this paper, the classical algorithm genetic algorithm in intelligent optimization algorithm and the classical algorithm Powell algorithm in traditional numerical algorithm are fused to take advantage of the powerful global searching ability of the genetic algorithm and the fast local convergence ability of the Powell method which does not need to derive the objective function, in order to obtain a high-performance algorithm for geometric constraint solving.

3. Methodology

3.1. Nonlinear equation systems and optimization problems

Often, designers sketch graphics can be based on the principles of analytic geometry expressed in the form of a system of equations, the parameters and expressions of the three basic geometric elements of the point, the line and the circle, as shown in Table 2, and the problem of solving the system of equations can be transformed into the problem of finding the extreme points of the function.

Table 2. Parameters and expressions for the three basic geometric elements: point, line and circle

Geometric	Representation	Parameters	Equation
Point	P_0	$P_0(x_0, y_0)$	$x = x_0, y = y_0$
line	$L(P_1, P_2)$	$P_1(x_1, y_1), P_2(x_2, y_2)$	$\frac{x-x_1}{x_2-x_1} = \frac{y-y_1}{y_2-y_1}$
Circle	C_0	$C_0(x_0, y_0, r)$	$(x - x_0)^2 + (y - y_0)^2 = r^2$

A system of nonlinear equations consists of n equations involving n unknown quantities of the form

$$\begin{cases} f_1(x_1, x_2, \cdots, x_n) = 0\,, \\ f_2(x_1, x_2, \cdots, x_n) = 0\,, \\ \cdots, \\ f_m(x_1, x_2, \cdots, x_n) = 0\,, \end{cases} \tag{1}$$

where $f_i(\boldsymbol{x}) = 0 (i = 1, 2, \cdots, m)$ is a system of nonlinear equations, $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)$ is the unknown vector of the system of equations.

The use of numerical methods and intelligent optimisation algorithms in solving systems of non-linear equations requires the conversion of a system of non-linear equations problem into an optimisation problem. Converting a system of nonlinear equations into a single objective optimisation problem is expressed in the form of

$$\min F(\boldsymbol{x}) = \sum\nolimits_{i=1}^{m} |f_i(x_1, x_2, \cdots, x_n)|\,, \tag{2}$$

or

$$\min F(\boldsymbol{x}) = \sum\nolimits_{i=1}^{m} f_i^2(x_1, x_2, \cdots, x_n)\,. \tag{3}$$

Then the problem of solving the system of nonlinear equations is transformed into the problem of finding the minimum value of the fitness function $F(\boldsymbol{x})$. In order to test the ability of the GA-Powell hybrid algorithm to solve high power complex problems, in this paper, the method of Eq. 3 is chosen to transform the system of equations into an optimisation problem for solving the system of nonlinear equations.

3.2. Overview of genetic algorithm and Powell's method

Genetic Algorithm (GA) is a prominent algorithm among meta-heuristics that draws inspiration from biological evolutionary processes [15]. GA emulates the Darwinian theory of survival of the fittest in the natural world. J.H. Holland first proposed GA in 1992.

The fundamental elements of GA comprise chromosome representations, fitness selections, and biologically inspired operators. Additionally, Holland introduced a novel element called inversion, which is widely used in GA implementations [19]. Normally, chromosomes are formatted using binary strings. In a chromosome, each locus (a specific position on the chromosome) has two possible allele forms: 0 and 1. Chromosomes are regarded as points in the solution space. These chromosomes undergo processing using genetic operators, by iteratively cumulating their populations. The fitness function applies in assigning a value to every chromosome within the population [15].

Operators inspired by biology include selection, mutation, and crossover. During selection, a chromosome is selected for further processing based on its fitness value. In the crossover operator, a random locus is selected, and subsequences between chromosomes are altered to create offspring. During mutation, certain random bits of chromosomes are flipped according to the probability distribution [19,26,15]. Table 3 displays the relationship between biological evolution and genetic algorithm. And the general steps of the genetic algorithm are shown in Fig. 2.

Genetic algorithm searches for diversity in the solution space and is more capable of finding the global optimal solution compared to other intelligent optimization algorithms without easily falling into the local optimal solution, which makes it perform well

Table 3. Correspondence between biological evolution and genetic algorithm

Biological Evolution	Genetic Algorithm
Individual	Feasible solution to a problem
Population	A group of feasible solutions
Chromosome	Encoding corresponding to a feasible solution
Gene	Element of the encoding
Environment	Adaptive function
Survival of the Fittest	The more the adaptive value approximates the optimal value, the more likely a feasible solution is to be retained.
Selection	The process of passing genetic information from parent individuals to offspring individuals.
Crossover	The process of generating offspring individuals from parent individuals through crossover.
Mutation	Changing a portion of the chromosome encoding.

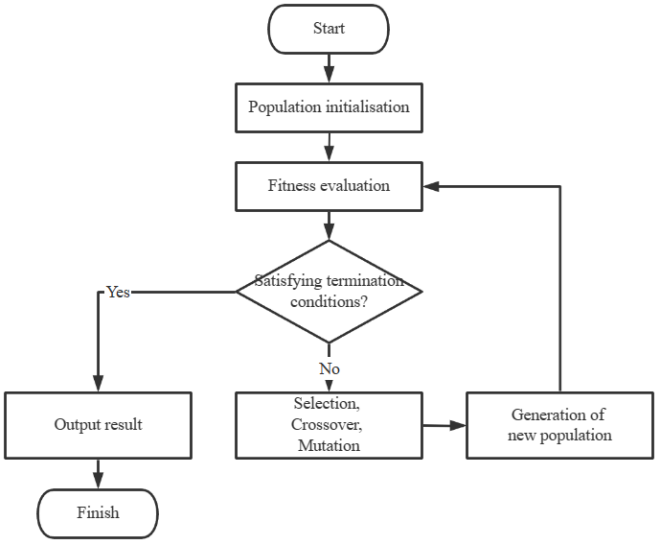


Fig. 2. Classification of geometric constraint solving methods

in complex multi-peaked functions and high-dimensional spaces. Moreover, the genetic algorithm is adaptive in the sense that it can automatically adjust to the complexity and characteristics of the problem. This makes it applicable to a wide range of problem types without the need to manually fine-tune the algorithm parameters. Meanwhile, the genetic algorithm is easy to parallelize, and multiple individuals can be evaluated and evolved at the same time, which improves the search efficiency, which facilitates the improvement of the algorithm's efficiency from the parallelization perspective in the future. Combining the above advantages, we chose the genetic algorithm as part of the hybrid algorithm.

Powell's method, also known as direction acceleration method, which was proposed by Powell [18] in 1964, is a search method formed by using the property that conjugate directions can accelerate the convergence speed. The steps of Powell's algorithm are shown in Algorithm 1. Powell's method can be used to solve general unconstrained optimization problems, and for the optimization problem of objective function with dimension $n < 20$, this method can obtain more satisfactory results. Unlike other direct methods, Powell's method has a complete theoretical system, so its computational efficiency is higher than other direct methods. The method does not require derivation of the objective function and can be applied when the derivative of the objective function is discontinuous, therefore, Powell's algorithm is a very effective direct search method, which we choose for fast convergence.

Algorithm 1: Powell's Method

Choose an initial point x_0 and set $p_i = e_i$, for $i = 1, 2, ..., n$;
Compute x_1 as the minimizer of f along the line $x_0 + \alpha p_n$;
Set $k \leftarrow 1$;
while *convergence test is not satisfied* **do**
    Set $z_1 \leftarrow x_k$;
    for $j = 1, 2, \ldots, n$ **do**
        Calculate α_j so that $f(z_j + \alpha_j p_j)$ is minimized;
        Set $z_{j+1} \leftarrow z_j + \alpha_j p_j$;
    end
    for $j = 1, 2, \ldots, n-1$ **do**
        Set $p_j \leftarrow p_{j+1}$;
    end
    Set $p_n \leftarrow z_{n+1} - z_1$;
    Calculate α_n so that $f(z_{n+1} + \alpha_n p_n)$ is minimized;
    Set $x_{k+1} \leftarrow z_{n+1} + \alpha_n p_n$;
    Set $k \leftarrow k + 1$;
end

3.3. Genetic-Powell hybrid algorithm

While the crossover and variational operators in genetic algorithms can look for solutions to equations in the complete space of variables, Powell's method does so rapidly and methodically in the area around the convergence point with a high level of local convergence.

To leverage the benefits of both approaches, this paper proposes the hybrid genetic-Powell algorithm. The algorithm's progression is explained as Algorithm 2.

Algorithm 2: GA-Powell Algorithm

Input: Optimized function $F(X)$ representing the equation system, number of genetic algorithm calls N, Euclidean distance threshold O_t
Output: Solution set of the equation system X^*

```
Initialize the initial solution set X_0;
for i from 1 to N do
    Obtain an initial solution x_i using the genetic algorithm;
    Add x_i to the set X_0;
end
for each initial solution x_i in set X_0 do
    for each other element x_j in set X_0 except x_i do
        if Euclidean distance between x_i and x_j is less than O_t then
            Remove x_j from set X_0;
        end
    end
end
Initialize the precise solution set X*;
for each initial solution x_i in set X_0 do
    Apply Powell's algorithm to x_i as the initial value and obtain the precise solution x*;
    Add x* to set X*;
end
Output Set X* containing precise solutions;
```

Genetic algorithm has strong global search ability, Powell locally and carefully searches for solutions near the convergence point and has high local convergence. The termination condition of the genetic algorithm is the set number of genetic generations. In order to play the global search ability of genetic algorithm, in this paper, when using genetic algorithm to obtain a set of initial solutions, it is not necessary to set the genetic generations to be very large in order to avoid missing some feasible solutions; at the same time, appropriately increase the number of individuals in the population and the mutation rate, in order to give full play to the global search ability of genetic algorithm.

After cyclically calling the genetic algorithm to obtain the initial solution, the algorithm calculates the similarity between the initial solutions to exclude the invalid solutions, i.e., to exclude multiple initial solutions that will converge to the same optimal solution, in order to greatly reduce the useless calculations and improve the efficiency of the algorithm. Euclidean distance is the simplest and easiest to understand method in all similarity calculations, and Euclidean distance is more suitable for data with obvious geometric structure in geometrically constrained solution scenarios, so we use Euclidean distance to determine the similarity of the solutions, and if the distance is greater than the set threshold then it is judged to be dissimilar, and the dissimilar solutions are retained.

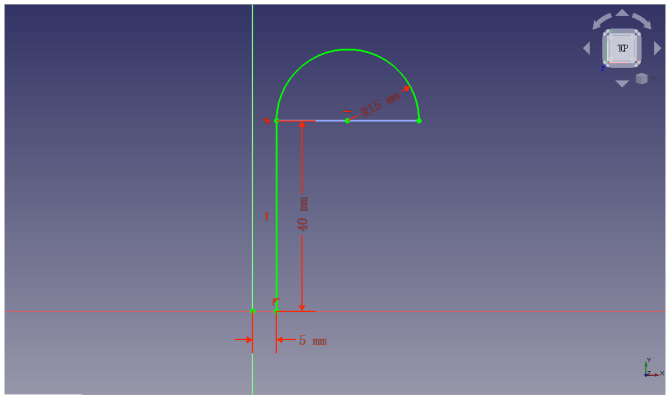


Fig. 3. The arc is tangent to the right of the line segment.

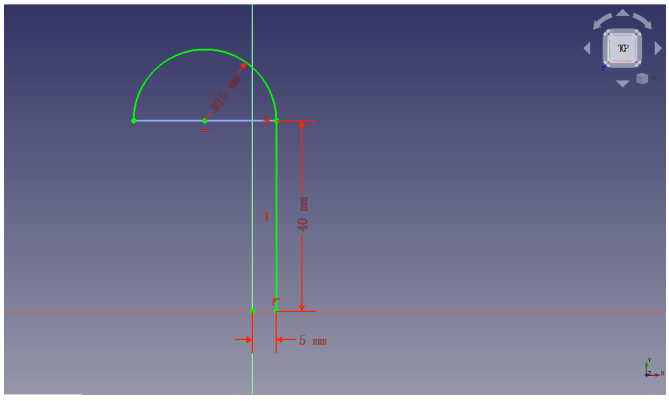


Fig. 4. The arc is tangent to the left of the line segment.

4. Experiment

In this paper, the application scenario of GA-Powell algorithm is static equipment design in chemical industry, which mainly involves geometric elements such as arcs and lines. The Fig. 3 and Fig. 4 show a simplified design example: if the head of the tower is tangent to the upper side of one barrel, there are two solutions as shown in the figure (regardless of the concave head into the barrel). Let the lower endpoint of the line segment be (x_1, y_1), the upper endpoint be (x_2, y_2), the center of the arc be (x_c, y_c), and the radius be R, then the constraint equation of this example is:

$$\begin{cases} \sqrt{(x_1-0)^2+(y_1-0)^2}=5, \\ y_1=0, \\ x_2=x_1, \\ \sqrt{(x_2-x_1)^2+(y_2-y_1)^2}=40, \\ R=15, \\ y_c=y_2, \\ \sqrt{(x_c-x_2)^2+(y_c-y_2)^2}=R. \end{cases} \tag{4}$$

Regardless of the optimization of the actual solver, the equations are input into GA-Powell algorithm and two optimal solutions are obtained. Optimal solution 1 is (5.000000000000956, 0.00000000000074145893, 5.000000000001288, 39.999999999 9704, 14.999999999998659, -9.999999999995033, 39.99999999999881). Optimal solution 2 is (4.999999999997651, -0.00000000000083986589, 4.999999999998072, 39.9 99999999999254, 15.000000000000888, 19.999999999997126, 39.99999999999913). The errors of the solutions are all within 10^{-20}, and the solution time is 0.245536 seconds.

In order to better test the performance of the GA-Powell algorithm for solving multi-root nonlinear systems of equations, for the convenience of comparison and without loss of generality, we use 9 selected systems of equations from reference [14] as the test systems of equations and compare them with the PGWO method proposed in that literature. And in order to prove that the GA-Powell algorithm is insensitive to the initial value, the initial value intervals of all the test equation sets except Eq. 8 are set to [-10, 10]. The simulation experiments were implemented in python on a PC with a CPU of AMD Ryzen 7 6800H at 3.20 GHz and 16.0 GB of RAM. Some of the key quantitative results are selected below to show the effect of the algorithm.

The experiment 1 is to solve

$$\begin{cases} f_1(x)=x_1^2+x_2^2-2=0\,, \\ f_2(x)=x_1^2+x_2^2/4-1=0\,, \end{cases} \tag{5}$$

the equation has four solutions, which are $(-0.8165,-1.1547)^T, (0.8165,-1.1547)^T, (-0.8165,1.1547)^T, (0.8165,1.1547)^T$.

and

$$\begin{cases} f_1(x)=x_1^4+4x_2^4-6=0\,, \\ f_2(x)=x_1^2x_2-0.6787=0\,, \end{cases} \tag{6}$$

the equation has four solutions, which are $(-1.56353, 0.27763)^T, (1.56353, 0.27763)^T, (-0.78971, 1.08830)^T, (0.78971, 1.08830)^T$.

The parameters are set as follows: the number of cycles of the genetic algorithm N = 10, the population is 100, the maximum number of iterations of the genetic algorithm is 50, the mutation rate is 0.03, and Powell's method has a maximum number of iterations of 500 and a convergence accuracy of 10^{-20}. The results of the experiment are shown in Table 4.

Table 4. Experiment 1 results

Equation	GA-Powell	Error
Eq. 5	-0.816496580927726 1.1547005383792515	0.0
	-0.8164965809277257 -1.154700538379252	1.9721522630525295e-31
	0.8164965809277259 1.1547005383792575	1.8858706015439813e-28
	0.8164965809277255 -1.1547005383792457	2.197470659106281e-28
Eq. 6	-1.5635325915863936 0.27762845243830503	4.078467204661264e-23
	-0.7897063597753254 1.0882948602064995	4.2762300902543955e-24
	0.7897063597753038 1.0882948602066016	2.9496002284279395e-29
	1.5635325915863927 0.2776284524446274	8.496843844873951e-23

As can be seen in Table 4, the GA-Powell algorithm is able to find all solutions and has higher accuracy compared to the PGWO method solutions. However, the accuracy of the solution of eq. 6 with higher powers decreases compared to eq. 5. The solution took 0.321683 seconds and 0.317919 seconds, respectively.

Experiment 2 solves the Eq. 7, Eq. 8 and Eq. 9. The parameters are set as follows: the number of cycles of the genetic algorithm N = 50, the population is 100, the maximum number of iterations of the genetic algorithm is 50, the mutation rate is 0.03, and Powell's method has a maximum number of iterations of 500 and a convergence accuracy of 10^{-20}. The results are shown in Table 5, Table 6 and Table 7.

$$\begin{cases} \cos(2x_1) - \cos(2x_2) - 0.4 = 0\,, \\ 2(x_2 - x_1) + \sin(2x_2) - \sin(2x_1) - 1.2 = 0\,, \end{cases} \tag{7}$$

the equation has 13 solutions, which are
$(-9.268258, -8.931402)^T, (-8.744542, -7.164787)^T, (-6.126665, -5.789809)^T,$
$(-5.602950, -4.023195)^T, (-2.985073, -2.648216)^T, (-2.461357, -0.881602)^T,$
$(0.156520, 0.493376)^T, (0.680236, 2.259991)^T, (3.298113, 3.634969)^T,$
$(3.821828, 5.401583)^T, (6.439705, 6.776562)^T, (6.963421, 8.543176)^T,$
$(9.581298, 9.918154)^T$.

Table 5. Experiment 2 results of Eq. 7

Equation	No.	GA-Powell	Error
Eq. 7	1	9.581298030452515 9.918154334992423	5.068086189398967e-25
	2	-6.12666523749644 -5.7898089329563325	2.3517915736901414e-28
	3	6.439705376862722 6.776561681402831	2.7240353133413064e-30
	4	-2.985072583906657 -2.648216279366548	5.053640174072107e-31
	5	-5.602949507250662 -4.023194565762204	2.6262773526874095e-24
	6	3.8218284535187137 5.401583395006905	8.69749223328177e-25
	7	-2.4613568536608685 -0.88160191217265	1.025081519222289e-24
	8	-9.268257891086272 -8.931401586543789	7.002171694601335e-23
	9	0.6802357999289202 2.259990741416713	6.570829235990062e-27
	10	-8.74454216084046 -7.164787219352157	1.474125009516472e-24
	11	6.963421107108508 8.543176048596612	5.089100457837923e-25
	12	3.2981127232729306 3.634969027817465	2.430470957633393e-22
	13	0.15652006968312956 0.4933763742281069	2.9416965098544255e-22

As can be seen from Table 5, the GA-Powell algorithm is still able to find all exact solutions for problems with a large number of roots. The solution took 1.758389 seconds.

$$\begin{cases} \sin({x_1}^3) - 3x_1x_2^2 - 1 = 0\,, \\ \cos(3{x_1}^2x_2) - \left|x_2^3\right| + 1 = 0\,, \end{cases} \tag{8}$$

the equation has 10 solutions, which are
$(-1.810885, -0.3490924)^T, (-1.810885, 0.3490924)^T, (-1.502221, -0.409077)^T,$
$(-1.502221, 0.409077)^T, (-1.791302, 0.301926)^T, (-1.791302, -0.301926)^T,$
$(-0.947268, 0.785020)^T, (-0.947268, -0.785020)^T, (-0.213057, 1.256845)^T,$
$(-0.213057, -1.256845)^T$.

Since Eq. 8 had too many solutions in the interval $[-10, 10]$, the interval was narrowed down to $[-2, 2]$. The GA-Powell algorithm did a better job of solving the problem, obtaining all the exact solutions. The solution took 1.923495 seconds.

Table 6. Experiment 3 results of Eq. 8

Equation	No.	GA-Powell	Error
Eq. 8	1	-0.9472681469862627 0.7850200155682888	4.930380657631324e-32
	2	-1.5022159860694961 0.40907656829415534	4.601031229701551e-28
	3	-0.2130566192382045 1.256845317434239	3.059160711793778e-22
	4	-1.5022159860694986 -0.4090765682941479	3.5745259767827097e-31
	5	-0.947268146986263 -0.7850200155682433	7.643742929444023e-26
	6	-1.810885199439986 -0.3490909919763665	3.5745259767827097e-31
	7	-1.791302084615447 -0.3019263417131378	4.930380657631324e-32
	8	-1.7913020846154477 0.3019263417147986	4.6459929979441084e-23
	9	-0.2130566192381985 -1.2568453174310898	2.7295323426479193e-24
	10	-1.810885199439986 0.34909099197636656	3.5745259767827097e-31

$$\begin{cases} -3.84{x_1}^2 + 3.84x_1 - x_2 = 0\,, \\ -3.84{x_2}^2 + 3.84x_2 - x_3 = 0\,, \\ -3.84{x_3}^2 + 3.84x_3 - x_1 = 0\,, \end{cases} \tag{9}$$

the equation has 8 solutions, which are
$(0.0000, 0.0000, 0.0000)^T$, $(0.4881, 0.9594, 0.1495)^T$, $(0.5403, 0.9538, 0.1694)^T$, $(0.9594, 0.1494, 0.4879)^T$, $(0.1494, 0.4881, 0.9594)^T$, $(0.9538, 0.1693, 0.5402)^T$, $(0.1693, 0.5399, 0.9538)^T$, $(0.7396, 0.7396, 0.7396)^T$.

Table 7. Experiment 3 results of Eq. 9

Equation	No.	GA-Powell	Error
Eq. 9	1	0.5403878416288977 0.9537362774344668 0.16943381967326443	2.465190328815662e-32
	2	0.7395833333333289 0.7395833333333411 0.739583333333685	5.465303046638133e-25
	3	0.16943381967327076 0.5403878416289855 0.9537362774344623	5.724788241092171e-27
	4	0.9537362774344795 0.16943381967316742 0.5403878416283857	9.46563760490936e-26
	5	0.14940689655344594 0.48800438713233574 0.9594474442442135	6.407722999371885e-29
	6	0.48800438713234184 0.959447444244211 0.14940689655344785	9.013968437314468e-29
	7	0.9594474442438022 0.14940689655528064 0.4880043871388201	3.51920906640077e-24
	8	-0.00000000000130681231 0.00000000000393663907 -0.00000000000165209884	3.8675468719474286e-22

As can be seen from Table 7, the GA-Powell algorithm still solves well when the number of variables increases. The solution took 1.951163 seconds.

From the above experiments, it can be seen that the GA-Powell algorithm proposed in this paper is able to find all the solutions of the above nine systems of equations with high solution accuracy, fast solution speed and insensitivity to the setting of the initial value.

5. Discussion

From the experiments in Section 4, it can be seen that the GA-Powell algorithm proposed in this paper indeed balances the advantages of the genetic algorithm and the Powell algorithm, and is able to accurately solve all the solutions of the problem with a large range of initial values, which is more effective compared to the PGWO method in reference [14]. The proposed algorithm theoretically gives an effective idea for solving multi-solution problems in the geometric constraint solving domain. This section mainly discusses the principle of parameterization of GA-Powell algorithm when targeting different types of systems of equations based on experimental results.

5.1. Systems of equations with a greater number of roots

The problems in Experiment 2 have a large number of solutions, and this kind of equations mainly test the global search ability of the genetic algorithm. When setting the parameters of the genetic algorithm, we should focus on enhancing its global search ability, so we need to increase the population size to increase the breadth of the search space, increase the variation rate to jump out of the local optimum, and also increase the number of times to call the genetic algorithm to obtain the initial value. The above operations can be used to make the genetic algorithm give initial values for all solutions for Powell's algorithm to converge to the exact solution.

5.2. Systems of equations with densely distributed roots

The roots of Eq. 9 are distributed in a small range of intervals, and the Euclidean distance between the solutions of such systems of equations is small, i.e., the similarity between the solutions is high, so when the genetic algorithm outputs the initial solutions and then screens them, it is necessary to reduce the Euclidean distance threshold for determining the similarity of the two values so as not to exclude the valid initial solutions.

5.3. Systems of equations with higher powers

Eq. 6 has higher powers of the variables, and these systems of equations are more complex and computationally intensive, requiring an increase in the number of iterations and termination accuracy of Powell's algorithm in order to obtain a highly accurate solution.

It can also be seen from the above discussion that the GA-Powell algorithm has limitations in terms of the parameter tuning problem, and the parameter adaptivity needs to be enhanced in the future; at the same time, the number of test cases used in this paper is not large enough, and there is a need to test the effectiveness of the algorithm by using more test cases and applying the algorithm to a real geometric constraint solver.

6. Conclusion

The conclusion of this paper highlights the research used to address the lack of support for solving multi-solution problems by geometric constraint solvers based on traditional numerical algorithms, which focuses on genetic algorithms and Powell algorithms, and the proposed hybrid GA-Powell algorithm combines the advantages and compensates for the shortcomings of the two. The results are compared and analysed in terms of the number of solutions, solution accuracy and solution time. The research successfully enhances the support of geometric constraint solving algorithm for solving multi-solution problems and obtains the following main research results:

- To address the shortcomings of the most prevalent geometric constraint solving algorithms based on traditional numerical algorithms investigate the selection of suitable algorithms in order to hybridize them in order to create complementary effects.
- The proposed GA-Powell hybrid algorithm has the advantages of high accuracy, high efficiency and high robustness in solving multi-solution problems.
- The experience of the GA-Powell hybrid algorithm is given to guide the parameter selection in different scenarios.

In this study, the method of mixing genetic algorithm with Powell algorithm to enhance the performance of geometric constraint solving algorithm is proposed, which provides a new idea to improve the solving of geometric constraints. The effectiveness of the GA-Powell algorithm is verified by conducting scientific analog simulation tests, which provides an important method for CAD systems to solve geometric constraint problems with multiple solutions. The results of the research are of great significance in guiding the further development and improvement of geometrically constrained solvers in the field of CAD, providing practical experience and suggestions for related work.

Future work in this research includes enhancing the parameter adaptivity of the algorithm, using more test cases, and applying the algorithm to a real geometric constraint solver through user experience in order to test the effectiveness of the algorithm.

References

1. Ait-Aoudia, S., Bahriz, M., Salhi, L.: 2d geometric constraint solving: An overview. In: 2009 Second International Conference in Visualisation. pp. 201–206. IEEE (2009)
2. Borning, A.: The programming language aspects of thinglab, a constraint-oriented simulation laboratory. ACM Transactions on Programming Languages and Systems (TOPLAS) 3(4), 353–387 (1981)
3. Bose, N., Bose, N.: Gröbner bases: An algorithmic method in polynomial ideal theory. Springer (1995)
4. Chen, R., Chu, J., Zhang, B.: A coincidental calculation method of some terminal-guided projectile inertial navigation segment ballistic coeffcients based on powell method. Fire Control & Command Control 45(8), 176–180 (2020)
5. Chunhong, C., Bin, Z., Limin, W., Wenhui, L.: The parametric design based on organizational evolutionary algorithm. In: PRICAI 2006: Trends in Artificial Intelligence: 9th Pacific Rim International Conference on Artificial Intelligence Guilin, China, August 7-11, 2006 Proceedings 9. pp. 940–944. Springer (2006)

6. Gao, X., Sun, L., Sun, D., et al.: Artificial immune-chaos hybrid algorithm for geometric constraint solving. Inf. Technology J pp. 360–365 (2009)
7. Hillyard, R., Braid, I.: Analysis of dimensions and tolerances in computer-aided mechanical design. Computer-Aided Design 10(3), 161–166 (1978)
8. Hillyard, R., Braid, I.: Characterizing non-ideal shapes in terms of dimensions and tolerances. In: Proceedings of the 5th annual conference on Computer graphics and interactive techniques. pp. 234–238 (1978)
9. Li, C., Li, G., Tan, Y., Xu, X.: Medical image registration algorithm based on powell algorithm and improved genetic algorithm. Journal of Computer Applications 33(3), 640–644 (2013)
10. Light, R., Gossard, D.: Modification of geometric models through variational geometry. Computer-Aided Design 14(4), 209–214 (1982)
11. Light, R.A., Gossard, D.C.: Variational geometry: a new method for modifying part geometry for finite element analysis. Computers & Structures 17(5-6), 903–909 (1983)
12. Lin, V.C., Gossard, D.C., Light, R.A.: Variational geometry in computer-aided design. ACM SIGGRAPH Computer Graphics 15(3), 171–177 (1981)
13. Liu, X., Yin, X., Li, C.: Improvements of powell mechanical optimization. Journal of Machine Design 36(6), 80–86 (2019)
14. Lu, M., Qu, L., He, D.: Pathfinder grey wolf algorithm for solving multiple-roots nonlinear equations. Chinese Journal of Engineering Mathematics 39(6), 957–968 (2022)
15. Michalewicz, Z., Schoenauer, M.: Evolutionary algorithms for constrained parameter optimization problems. Evolutionary computation 4(1), 1–32 (1996)
16. Nelson, G.: Juno, a constraint-based graphics system. In: Proceedings of the 12th annual conference on Computer Graphics and Interactive Techniques. pp. 235–243 (1985)
17. Para, W., Bhat, S., Guerrero, P., Kelly, T., Mitra, N., Guibas, L.J., Wonka, P.: Sketchgen: Generating constrained cad sketches. Advances in Neural Information Processing Systems 34, 5077–5088 (2021)
18. Powell, M.J.: An efficient method for finding the minimum of a function of several variables without calculating derivatives. The computer journal 7(2), 155–162 (1964)
19. Sampson, J.R.: Adaptation in natural and artificial systems (john h. holland) (1976)
20. Seff, A., Ovadia, Y., Zhou, W., Adams, R.P.: Sketchgraphs: A large-scale dataset for modeling relational geometry in computer-aided design. arXiv preprint arXiv:2007.08506 (2020)
21. Sutherland, I.E.: Sketchpad: A man-machine graphical communication system. In: Proceedings of the May 21-23, 1963, spring joint computer conference. pp. 329–346 (1963)
22. Wenjun, W.: Basic principles of mechanical theorem proving in elementary geometries. Selected Works Of Wen-Tsun Wu p. 195 (2008)
23. Yuan, H., Li, W., Yi, R., Zhao, K.: The tpso algorithm to solve geometric constraint problems. Journal of Computational Information Systems 2(4), 1311–1316 (2006)
24. Yuan, H., Chang, X.: Combining genetic algorithm with simlex method for geometric constraint solving. In: 2010 International Conference on Computer, Mechatronics, Control and Electronic Engineering. vol. 1, pp. 453–455. IEEE (2010)
25. Yuan, H., Yu, C.: Optimization genetic algorithm for geometric constraint solving. In: 2010 International Conference on Artificial Intelligence and Education (ICAIE). pp. 590–593. IEEE (2010)
26. Zbigniew, M.: Genetic algorithms+ data structures= evolution programs. Comput Stat pp. 372–373 (1996)

Sun Yunlei (Corresponding author: sunyunlei@upc.edu.cn) is a PhD, associate professor and master director, who graduated from Beijing University of Posts and Telecommunications with a doctorate degree in engineering in 2014. In recent years, he has been

engaged in the research of intelligent design, machine learning and other fields, presided over a number of special projects of basic scientific research operations of central universities, lateral projects of enterprises and institutions, and participated in the projects of National 863 and 973 projects, National Natural Science Foundation of China, and Shandong Provincial Natural Fund as an academic backbone.

Li Yucong received his B.S. degree in Computer Science and Technology from China University of Petroleum (East China) in 2021. Currently, he is a master's student at China University of Petroleum (East China). His main research interests are geometric constraint solving problems in computer-aided design.

Received: September 07, 2023; Accepted: January 05, 2024.

Computer Science and Information Systems 21(4):1567–1582 https://doi.org/10.2298/CSIS231015048L

ALFormer: Attribute Localization Transformer in Pedestrian Attribute Recognition

Yuxin Liu, Mingzhe Wang, Chao Li, and Shuoyan Liu*

Institute of Computing Technology, China Academy of Railway Sciences Corporation Limited
Beijing, China
{yxin_l,zzh355997367,lchao1234}@163.com
06112062@bjtu.edu.cn

Abstract. Pedestrian attribute recognition is an important task for intelligent video surveillance. However, existing methods struggle to accurately localize discriminative regions for each attribute. We propose Attribute Localization Transformer (ALFormer), a novel framework to improve spatial localization through two key components. First, we introduce Mask Contrast Learning (MCL) to suppress regional feature relevance, forcing the model to focus on intrinsic spatial areas for each attribute. Second, we design an Attribute Spatial Memory (ASM) module to generate reliable attention maps that capture inherent locations for each attribute. Extensive experiments on two benchmark datasets demonstrate state-of-the-art performance of ALFormer. Ablation studies and visualizations verify the effectiveness of the proposed modules in improving attribute localization. Our work provides a simple yet effective approach to exploit spatial consistency for enhanced pedestrian attribute recognition.

Keywords: spatial attention, attribute localization, contrast loss, random mask.

1. Introduction

Due to the constant demand for intelligent video surveillance and psychosocial behavior research, Pedestrian Analysis Recognition (PAR) is a long-standing research topic.Compared to low-level features, pedestrian attributes can be considered high-level semantic information and are more robust against changes in lighting conditions and viewpoint. Therefore, PAR is widely applied in the tasks of person re-identification, face verification, and human identity recognition.

With the development of deep learning technology, PAR has made significant progress in the past decade. Existing methods[1–4] mainly use complex networks such as Feature Pyramid Network (FPN) to enrich attribute representation from multilevel feature maps and combine attention mechanisms to accurately locate attribute-related regions. However, these CNN-based methods emphasized the extraction of local features, it is difficult to capture long-range dependencies.

In recent years, vision transformers such as ViT [5], DeiT[6]and Swin Transformer[7] have been widely used in various computer vision tasks, which combine the learning methods of natural language processing and computer vision. Using the multi-head self-attention module to focus on multiple areas at the same time, transformer-based models

* Corresponding author

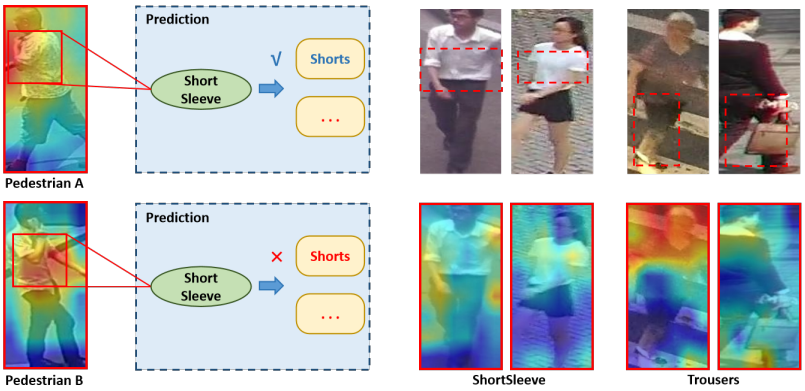


Fig. 1. Illustration of the problem of predicting by inter-attribute correlation and our main hypothesis on attribute localization.In(a), Class Activation Maps (CAM)[8] (with red boundary) of the baseline method in "short sleeve" and "trousers" attributes of the PA100K are visualized in the second row. Attribute-related regions of each attribute are plotted by the red dotted frame in the first row. In (b), it mainly expresses the process of Swin Transformer utilizes correlation between attributes for prediction, where the green attributes indicate the attributes that Swin Transformer focus on, and the yellow attributes indicate the attributes that Swin Transformer predicts based on the correlation

can capture long-range dependencies and retain more detailed information, which makes them suitable for extracting rich structural features. However, they mainly complete feature extraction by modelling the fine-grained relevance of images. In the task of PAR, it is manifested as over-reliance on attribute relevance for prediction, thereby reducing the accuracy of attribute positioning. As shown in Figure 1(a), for pedestrian A, Swin Transformer focuses on the short sleeve attribute region and can identify pedestrian A as a person with shorts based on its correlation to the short sleeve. However, for pedestrian B with trousers instead of shorts, the correlation between the short sleeve and shorts does not exist. In this case, Swin Transformer still tends to use the prior correlation to predict via focusing on the short sleeve attribute region, which results in incorrectly attribute localization and predicting the trousers as shorts attribute for pedestrian B. Furthermore, recognition errors may occur for other attributes such as 'handbag', 'female', and 'long hair'. Although there is a significant correlation between these attributes, it is not absolute. In the PA100k [1] dataset, a considerable proportion of males carry handbags. Therefore, conventional Swin Transformer-based methods struggle to achieve satisfactory recognition results for these attributes due to incorrect attribute relevance. This highlights the baseline network's over-reliance on attribute relevance. So correct semantical correlation (such as gender attributes and skirt attributes) is critical between attributes [9].

Additionally, we hypothesize that the spatial location of the same attribute mostly appears at the similar position between different pedestrian images. For example, the short sleeve attribute and the trousers attribute mostly appear in the upper and lower part of the picture, respectively, which is shown in the first row of Figure 1(b). However, we

observe that Class Activation Maps (CAM) [8] of the same attribute generated by the baseline method exhibit significant location variations. Some examples are shown in the second row of Figure 1(b). Considering two aspects, we propose Attribute Localization Transformer (ALFormer). Specifically, ALFormer uses attribute spatial memory(ASM) module that generates reliable and stable attribute-related regions to suppress position shifts caused by over-fitting or label noise of single image. Then ASM extracts inherent information based on accurate attribute positions, which assists to extract more comprehensive and represent attribute spatial features. Experimental evaluations results show that the proposed method achieves 1.79% and 2.34% improvements on widely adopted public datasets PA100k and PETA..

We make the following four contributions in this works:

- The Attribute Localization Transformer (ALFormer), a novel Transformer framework is proposed to extract comprehensive attribute spatial information for pedestrian attribute recognition.
- Mask Contrast Learning (MCL) scheme is introduced to suppress relevance to locate more precise region for each attribute.
- The Attribute Spatial Memory (ASM) module is designed to generate exact spatial attention regions and extract discriminative semantic features for each attribute.
- The experiments on two datasets (PETA and PA100K) show that our framework has excellent performance compared with several other state-of-the-art methods. In addition, we also present visualization results to verify the effectiveness of the proposed framework.

2. Related Works

Due to the constant demand for intelligent video surveillance and psychosocial behavior research, pedestrian attribute recognition is an important and challenging task.

Early methods [10–12] usually rely on hand-crafted features and learn individual classifiers for each attribute. Kumar et al. [11] extracted HOG and color histograms of important facial feature areas and then trained multiple support vector ma-chines for facial attribute recognition. Bourdev et al. [12] developed a three-layer SVM method to extract high-level semantic information. However, it cannot effectively utilize the potential relevance between attributes for separated classifiers.

Recently deep learning is well incorporated into PAR from a variety of studies. Liu et al. [13] designed a deep CNN composed of LNet and ANet to locate faces and predict facial attributes. Abdulnabi et al. [14] proposed a joint multi-task learning method to achieve information sharing between attributes for clothing attribute prediction. Hand and Chellappa [15] adopted a multi-task learning CNN method based on a grouping scheme to classify facial attributes. Han et al. [16] designed a Deep Multi-Task Learning CNN method (DMTL) that learns shared features for all attributes and specific category features for heterogeneous attributes. Li et al. [17] proposed a landmark-free facial attribute prediction method that does not rely on landmark an-notations. Li et al. [18] introduced pedestrian structural knowledge into pedestrian at-tribute recognition and proposed a posture-oriented pedestrian attribute deep prediction model.

Based on global feature similarities, the JRL network [19] uses Long Short-Term Memory [20] to treat the pedestrian attribute recognition task as a sequence prediction

problem. Attention mechanisms [1, 2, 18, 21, 22] are widely used in pedestrian attribute recognition to locate attribute-related regions and learn discriminative feature representations. Tan et al. [21] proposed a pedestrian attribute analysis method based on parsing attention, label attention, and spatial attention. The multi-directional attention module HydraPlus-Net [1] extracts pixel-level features and semantic-level features to locate fine-grained attributes. Liu et al. [2], based on CAM [8] and EdgeBox [23], proposed a Localization Guided Network method to extract attribute-related local features. The PGDM [18] framework uses a pre-trained human soft tube estimator and Spatial Transformer Networks (STNs) [24] to generate reliable attribute-related regions.

In recent years, Transformers have played an extremely important role in deep learning. It is a model proposed by Vaswani et al. [25] for natural language processing (NLP) tasks with far-reaching impact. The Vision Transformer (ViT) proposed by Dosovitskiy et al. [5] achieved breakthrough applications in the field of computer vision, combining learning methods from natural language processing and computer vision, using multi-head self-attention modules to simultaneously focus on multiple regions, capture long-range dependencies and retain more detailed information, thereby extracting rich structural features to achieve higher performance improvements. DeiT proposed by Touvron et al. [6] uses knowledge distillation to guide transformers to learn more effectively so that the network no longer needs large-scale pre-training data. To reduce computation of ViT, Liu et al. proposed the Swin Transformer [7], which introduces a shift-window-based attention mechanism to reduce computational costs. Transformer models achieve competitive performance in different computer vision tasks. However, such methods usually rely too much on predicting image region relationships, which leads to biased learning of attribute position information in PAR.

As stated, previous methods generally considered aspects such as global or local pedestrian images, attention mechanisms, and attribute correlation but these methods only considered a single image and could not fully exploit the spatial information of each pedestrian attribute. In contrast, our method makes full use of the potential attribute spatial information among different images. With proposed ASM and MCL modules, framework can better focus on attribute region and in turn obtain robust performance.

3. Attribute Localization Transformer (ALFormer)

This section first describes the overall architecture of Attribute Localization Transformer (ALFormer). After that, we present an attribute joint learning framework composed of two parts: an attribute spatial module (ASM) module and a mask contrast learning (MCL) scheme that incorporates Random Mask and contrast loss. Finally, we present the detail of the employed loss function for ALFormer.

3.1. Overall Architecture

Figure 2 shows the overall structure of the proposed ALFormer framework. Our basic network is identical to the four stages of Swin Transformer. Given a 2D image, it uniformly splits the input image into a sequence of 2D non-overlapping flattened patches. However, studies have shown that basic network over-reliance on attribute relevance for prediction, thereby reducing the accuracy of attribute positioning. This paper proposes MCL

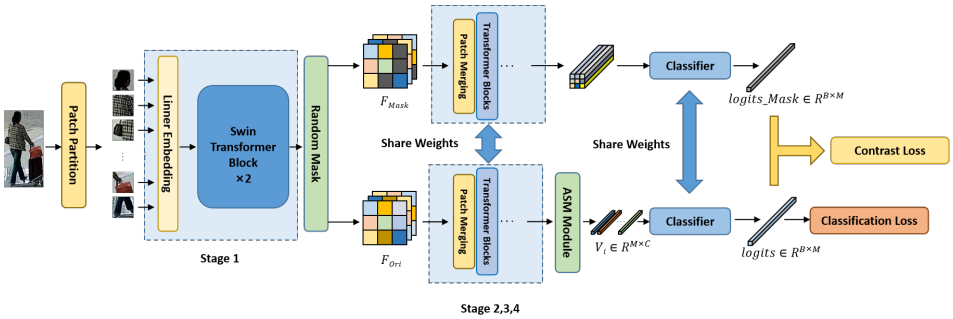


Fig. 2. Overall framework architecture

scheme. It first introduces a random mask module that obtains feature map from stage 1 and generates two results: F_{mask} and F_{Ori} .F_{mask} (masked feature map) is fed into the rest stages and classifier to obtain prediction $logits_Mask$. F_{Ori}(original input feature map) is put into stage 2,3,4 and ASM module to generate prediction $logits$. ASM module generates spatial attention map and embedded vector V_i for each attribute. ALFormer calculates contrast loss between $logits$ and $logits_Mask$. In addition, classification loss is calculated by $logits$. Finally, ALFormer outputs the predicted score on attributes.

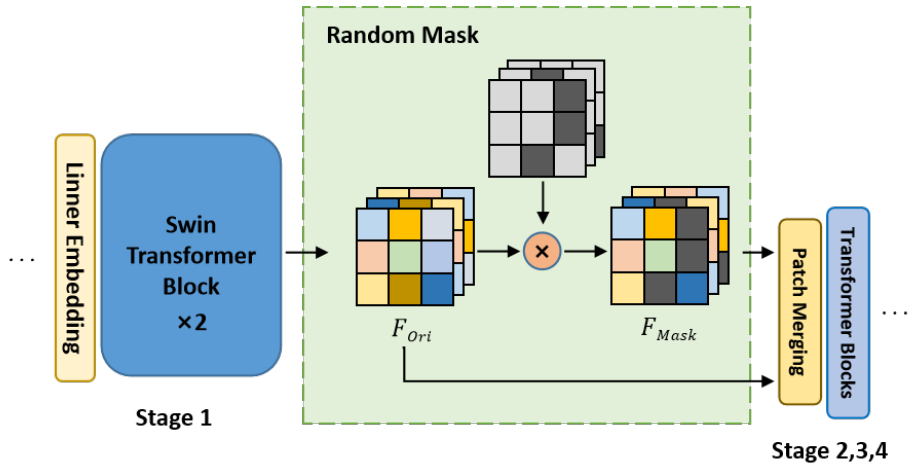


Fig. 3. MCL scheme structure. We introduce Random Mask (RM) on the feature map of backbone stage 1, and calculate the contrast loss with the original feature prediction

3.2. Mask Contrast Learning Scheme

As illustrated in Figure 2, the summation of ASM module and MCL scheme is to capture the spatial information of pedestrian attributes. Different spatial regions in input image

are associated with different attributes. It is desirable to focus on related regions when recognizing one attribute. However, we observe that Swin Transformer tends to focus on a relatively small number of local regions within which attributes may be correlated with other attributes, which leads Swin Transformer to predict attributes in those neglected regions based on such correlation. In fact, discriminative information may exist within these neglected regions, which is crucial for attribute identification. To address this problem, we propose MCL scheme which consists of Random Mask and contrast loss. It can force the framework to focus on intrinsic and discriminative attribute spatial regions for PAR tasks. The MCL scheme is illustrated in Fig.3

Different from existing random occlusion schemes, the proposed MCL scheme is performed at the feature level. The advantage is to reduce the correlation among feature regions and force the framework to focus on exact attribute spatial regions.

Specifically, random mask features are generated for each batch of input image at corresponding number. And then, Stage 1 of the framework outputs F_{Ori} in the spatial dimension of the feature map, as shown in the formula:

$$F_{mask} = RM(F_{Ori}) \tag{1}$$

A contrast loss function L_{con} is set at the classifier stage to evaluate the difference be-tween prediction results with and without Random Mask.As shown in the formula:

$$L_{con} = -pred_{Ori} \times \log(pred_{Mask}) \tag{2}$$

3.3. Attribute Spatial Memory Module

In this section, we propose the Attribute Spatial Memory(ASM) module to tackle the spatial attention region deviation problem. As shown in Figure 4, we visually demonstrate how to extract attribute spatial and semantic feature from ASM module. ASM module first takes input feature to generate attention map $A \in R^{B\times M\times H\times W}$, and then selects reliable ones and saves them in Memory module. Finally, it generates exact spatial attention regions and extract discriminative semantic features for each attribute.

In detail, ASM module takes feature map $F \in R^{B\times C\times H\times W}$ and classifier weight $w \in R^{M\times C}$ as inputs, where $F \in R^{B\times C\times H\times W}$ is the output of the backbone framework and H, W, C represent height,width and channel dimension of feature map.

Inspired by CAM[8], we introduce probability values as a criterion to construct spatial attention. As shown in the formula, ASM generates attention maps for each attribute by weighting the classifier:

$$Attention_Map\ A_{i,m} = w_m \cdot F_i(x, y) \quad m = 0, 1, 2 \ldots, M-1 \tag{3}$$

where M is the number of attributes, F_i is the feature map of the backbone framework, w_m is the classifier weight of the m-th attribute.

After getting the spatial attention regions of each attribute, we adopt the selector which takes logits and ground truth labels as input to aggregate the attention maps of qualified positive samples of$m$-th attribute. The main reason is that some attributes among images are negative or low confidence values. As shown in the formula, ASM filters out negative samples to get high-confidence predictions:

$$Attention_Map = \{A_{i,m} \mid Lable_{i,m} = 1, Logitis_{i,m} > coeff_threshold\} \tag{4}$$

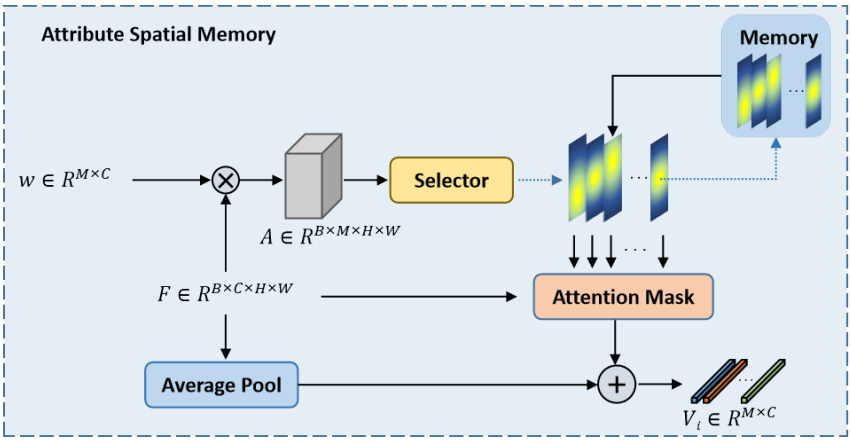


Fig. 4. Attribute Spatial Memory(ASM) module structure

Through strict selection, $A_{i,m}$can be regarded as reliable attribute spatial locations for m-th attribute. A criticism of this experimental design is that in actual training process, there are differences in attribute distribution of images in different batches. Moreover, all images in a batch may be negative on some attributes. Inspired by EMA strategy, we adopt a momentum-weighted scheme to update attention maps and save the reliable spatial location. In addition, for cases where all images in a batch are negative on some attributes, the corresponding attention map will not participate in updates. Specifically shown in the formula:

$$A_{s+1} = l \cdot A_{e+1} + (1-l) \cdot A_{e+1} \tag{5}$$

Where s is the number of training batches step. We save the generated attribute spatial attention map in memory module to assist subsequent feature pooling fusion.

The attention map $A_{i,m}$ indicates the contribution of different spatia regions to the prediction of attributes.. Therefore, ASM generates corresponding embedding vectors for different attributes separately, which weights the spatial location information. Thus improves spatial localization for each attribute. As shown in the formula, we normalize the spatial attention map and then use it as a spatial weighting coefficient to perform weighted pooling on the feature Feature:

$$Embed_Attr_m = Attention_Map(F_i) \quad m = 0, 1, 2 \ldots, M-1; i = 0, 1, 2 \ldots, B-1 \tag{6}$$

Where M is the number of attribute types and B is the number of batches. The semantic embedding vector $Embed_Attr_m$ for each attribute is obtained through weighted pooling.

3.4. Loss Function

As illustrated in Figure 2, contrast loss L_{con} and classification loss L_{cls} are obtained from the ALFormer. The classification loss is calculated according to binary cross entropy loss,

which is formulated as follow:

$$L_{cls} = \frac{1}{N}\sum_{i=1}^{N}(y_i \log p_i + (1 - y_i) \log (1 - p_i)) \tag{7}$$

The final loss function L is a weighted summation of the classification loss and contrast loss:

$$Loss = L_{cls} + \lambda \cdot L_{con} \tag{8}$$

Where $\lambda = 1$ is set as default in all experiments if not specially specified.

4. Experiments

4.1. Datasets and Evaluation Metrics

We conduct experiments on public datasets PETA [26] and PA100K [1]. Figure 5 shows some pedestrian image samples. The Pedestrian Attribute (PETA) dataset [26] is collected from 10 small-scale human datasets and consists of 19,000 human images, of which 9,500 images are used for the training set, 1,900 for the validation set, and 7,600 for the test set. Each image is labeled with 61 binary attributes and 4 multi-class attributes. We follow the commonly used experimental scheme [3, 4, 27] and only evaluate 35 attributes with a positive rate greater than 5%.

PA100K dataset is constructed from images taken by 598 real outdoor surveillance cameras and contains 100,000 pedestrian images with resolutions ranging from 50 × 100 to 758 × 454. It is the largest pedestrian attribute recognition dataset to date. The entire dataset is randomly divided into training set, validation set and test set in a ratio of 8:1:1. Each image in the dataset is labeled with 26 attributes, with labels of 0 or 1 indicating the presence or absence of the corresponding attribute.

Two metrics are used to evaluate attribute recognition performance [27], namely label-based metrics and four instance-based metrics. For label-based metrics, Deng et al. [27] use mean accuracy (mA) to evaluate attribute recognition algorithms. For each attribute, mA indicates the classification accuracy of positive and negative samples separately and then takes their average value as the recognition result of that attribute. Finally, the average value of all attributes is taken to obtain the recognition rate. Specifically, the experiment uses this indicator as part of the evaluation criteria as shown in the formula:

$$mA = \sum_{i}^{L}(\frac{TP_i}{P_i} + \frac{TN_i}{N_i}) \tag{9}$$

where L is the number of attributes. TP_i and TN_i are the number of correctly predicted positive and negative samples respectively, and P_i and N_i are the number of positive and negative samples respectively.

For instance-based metrics, we use Accuracy, Precision, Recall and F1-score.



(a) (b)

Fig. 5. Public dataset image samples. (a) Pedestrian samples from the PA100k dataset; (b) Pedestrian samples from the PETA dataset

4.2. Implement Details

Our method is implemented based on the PyTorch framework and trained in an end-to-end manner. Since our work mainly focuses on how to use local features defined by attributes, such framework can improve spatial location for each attribute. In the experiment, we adopt Swin Transformer [7] as the backbone network to extract pedestrian image features for a fair comparison. Pedestrian images are uniformly adjusted to 256×192 as input. Random horizontal mirroring, padding and random cropping are used for enhancement. AdamW is used for training with a weight decay of 0.0005. The initial learning rate is 0.0001 and the batch size is set to 64. The total epoch of the training phase is 50. The momentum coefficient is 0.9998. The label smoothing coefficient is set to 0.2.

4.3. Baseline Method

Given a dataset, $D = \{(X_i, y_i) \mid i = 1, 2, \ldots, N\}$, PAR aims to predict multiple attribute $y_i \in \{0, 1\}$ to i-th pedestrian image, where N,M denotes the number of images and attributes respectively, zero and one in the attribute vector y_i indicate the absence and presence of the corresponding attributes of a pedestrian image. Following [3, 4, 27, 28], we formulate pedestrian attribute recognition as a multi-label classification task which adopts multiple binary classifiers with sigmoid functions [27]. Binary cross-entropy loss L_{cls} is used as the optimization target:

$$L_{cls} = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{M} (y_{i,j} \log p_{i,j} + (1 - y_{i,j}) \log (1 - p_{i,j})) \tag{10}$$

where $p_{i,j}$ is the prediction probability of the classifier.

4.4. Ablation Analysis

In this section, we first study the impact of the ASM and MCL modules by conducting analysis experiments on both two datasets. The baseline methods use the original Swin Transform set only in section 4.3. As shown in Table 1, compared to the baseline method, we have conducted the following observations. First, when the ASM module is not used, the attribute positioning accuracy is worse and the recognition accuracy decreases. The results show that without correct attention regions, attribute semantic features lack discriminative ability and contain more noise. The ASM module generates reliable attribute space attention maps through attribute prediction scores, guiding attribute space feature fusion and improving attribute positioning accuracy. Second, using the ASM module can directly bring 1.18% and 2.06% performance improvements on PA100K and PETA respectively, indicating that the spatial memory module is beneficial for locating attribute-related regions. In addition, we introduced occlusion contrast loss MCL by generating random occlusions on pedestrian images and evaluating the difference between their predictions and normal input predictions. By suppressing regional relevance to improve the framework's attribute positioning ability, it brings 0.39% and 0.47% performance improvements on PA100K and PETA respectively. Finally, when the ASM module and MCL module are used together, our method's mA on PA100K and PETA is 1.79% and 2.34% higher than the baseline model.

In Figure 6, by comparing the recognition mean accuracy rates of the baseline model Swin Transform and our model in the 26 attributes of the PA100K dataset , it can be seen that the model in this paper has a recognition effect on most attributes. For some attributes that rely on spatial regions (such as "hat", "boots"), the improvement is significant due to ALFormer focus on intrinsic spatial regions. For some attributes that inferred from other related attributes (e.g. "female"), the improvement is also significant because ALFormer could generate reliable attention maps that capture inherent locations for each attribute.

Table 1. Ablation study of each component of our method on the PETA and PA100K. Performance 328 improved by Mask Contrast Learning (MCL) and Attribute Spatial Memory (ASM) validates the 329 effectiveness of our methods

Model	PA100K					PETA				
	mA	Accu	Pre	Recall	F1	mA	Accu	Pre	Recall	F1
Baseline	82.82	81.47	89.08	88.88	88.98	87.20	80.17	**86.54**	88.73	87.62
Ours(+MCL)	83.21	**81.70**	**89.18**	88.99	**89.09**	87.67	71.65	76.81	89.01	82.46
Ours(+SPM)	84.00	77.00	82.85	90.08	86.32	89.26	79.55	84.83	89.92	87.30
Ours(+SPM,MCL)	**84.61**	78.86	84.11	**91.03**	87.43	**89.54**	**80.75**	86.15	**90.04**	88.05

4.5. Comparison with State-of-the-art Methods

In Table 2, we compare the performance of the proposed methods with several existing algorithms on the PA100K and Table 3 is on the PETA. Compared with the performance reported by the paper of MsVAA [3], VAC [28], and ALM [4] methods, ALFormer achieves better performance on the PETA and PA100K with increasing few learnable parameters.

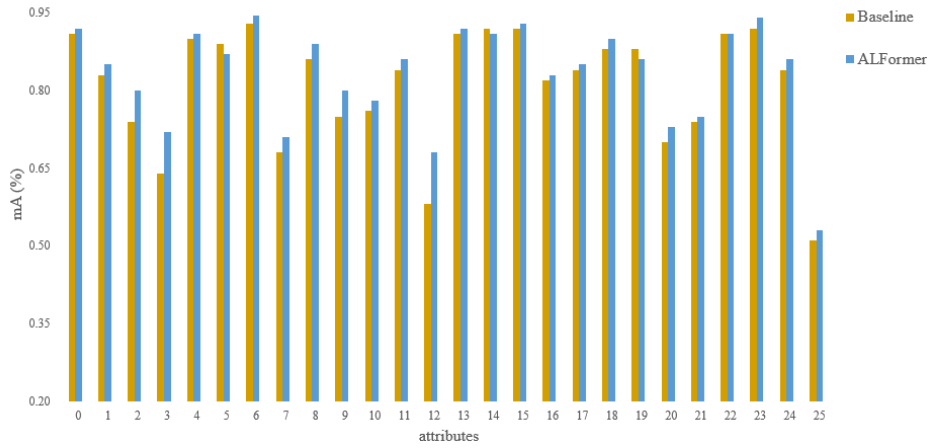


Fig. 6. The mA for each label on PA100k compared to baseline model. Legends in the figure: models are the same as in Table 1. The labels 0 to 25 are: female, ageover60, age18-60, ageless18, front, side, back, hat, glasses, handbag, shoulderbag, backpack, holdobjectsinfront, shortsleeve, longsleeve, upperstride, upperlogo, upperplaid, uppersplice, lowerstripe, LowerPattern, longcoat, trousers, shorts, skirtanddress, boots

Compared to the MsVAA model adopted ResNet101 as the backbone network, we achieve 4.95% and 1.56% performance improvements in mA and F1 on the PETA dataset. Compared with localization-based methods such as the ALM model, which utilizes the complicated combination of FPN, STN and SE modules introducing extra 17% parameters, ALFormer achieves 3.93% and 3.24% performance improvements in mA on both popular datasets. We localize attributes by ASM, which is a simple yet effective attention module to generate reliable and stable attribute-related regions and avoid heavy handcrafted design.

It can be noticed that the proposed method substantially outperforms the visual attention consistency (VAC) method. The VAC method hypothesis that global attention regions of random augmentations of the same image are consistent. The VAC method focuses on global attention regions of an individual image and cannot generate precise local attention regions for each fine-grained attribute. However, for a pair of augmentations of the same image, if attention regions of both augmentations are inaccurate, the VAC method cannot solve the attention region deviation problem, which can be addressed by our proposed method.

4.6. Effects of ALFormer

To better understand the effect of the improved model, we visualize the attribute regions most attended by the framework in Figure 7. It visualizes the attribute heat maps of Swin Transformer on the test set of the PA100K dataset. We take attribute attention maps of four pedestrian images(from P1 to P4) as examples that shows the visualization of spatial attention regions between the baseline method (green boundary) and the proposed AL-

Table 2. Performance (%) of the comparison state-of-the-art methods and the proposed method on the PA100K datasets. Five metrics, mean accuracy (mA), accuracy (Accu), precision (Prec), recall (Recall), F1 are evaluated. The best results are highlighted in red color, while the second bests are in blue.

method	mA	Accu	Pre	Recall	F1
DeepMar[29]	72.70	70.39	82.24	80.24	81.32
HP-Net[1]	74.21	72.19	82.97	82.09	82.53
PGDM[19]	74.95	73.08	84.36	82.24	83.29
JLPIS-PAA[31]	81.61	78.89	86.83	87.73	87.27
ALM[4]	80.68	77.08	84.21	88.84	86.46
JLAC[30]	82.31	79.47	87.45	87.77	87.61
CoCNN[37]	80.56	78.30	89.49	84.36	86.85
LG-Net[32]	76.96	75.55	86.99	83.17	85.04
VAC[9]	79.04	78.95	88.41	86.07	86.83
SSC[33]	81.70	78.85	85.80	88.92	86.89
ALFomer	84.61	78.86	84.11	91.03	87.43

Table 3. Performance (%) of the comparison state-of-the-art methods and the proposed method on the PETA datasets. The best results are highlighted in red color, while the second bests are in blue

method	mA	Accu	Pre	Recall	F1
CNN+SVM[13]	76.65	45.41	51.33	75.14	61.00
ACN[38]	81.15	73.66	84.06	81.26	82.64
DeepMar[29]	82.89	75.07	83.68	83.14	83.41
HP-Net[1]	81.77	76.13	84.92	83.24	84.07
PGDM[19]	82.97	78.08	86.86	84.68	85.76
DIAA[35]	84.59	78.56	86.79	86.12	86.46
JLPIS-PAA[31]	84.88	79.46	87.42	86.33	86.87
WPAL[34]	85.50	76.98	84.07	85.78	84.90
ALM[4]	86.30	79.52	85.65	88.09	86.85
JRL[20]	85.67	-	86.03	85.34	85.42
RCRA[36]	85.78	-	85.42	88.02	86.07
JLAC[30]	89.96	80.38	87.81	87.09	87.45
CoCNN[37]	86.97	79.95	87.58	87.73	87.65
MsVAA[3]	84.59	78.56	86.79	86.12	86.46
VAC[9]	83.63	78.94	87.63	85.45	86.23
SSC[33]	86.07	79.23	84.58	89.26	86.54
ALFomer	89.54	80.75	86.15	90.04	88.05

Former framework (red boundary). Compared with the baseline method, it is noted that the ALFormer helps to locate attribute-related regions for each attribute. In pedestrian image P2 and P3 shown in Figure 7, only head region is considered when recognizing the attribute "Glasses". Our method can effectively improve spatial location for each attribute by generating stable spatial attention and relevance suppression modules.

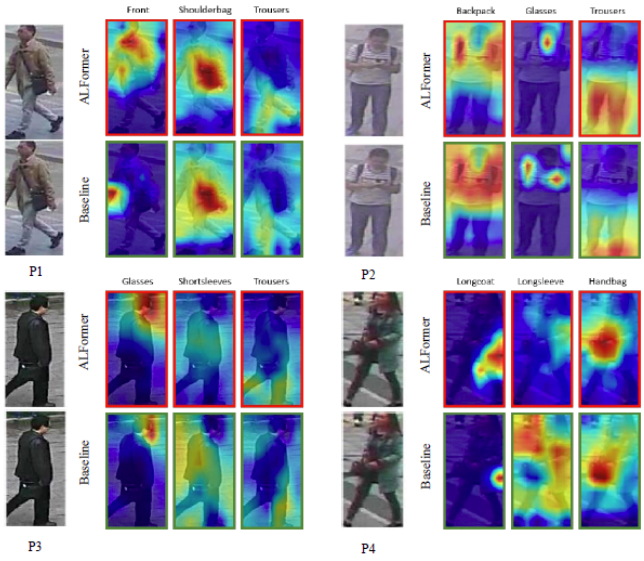


Fig. 7. Illustration of the effect of ALFormer. We take the attributes of four pedestrian images as examples to show the visualization of spatial attention regions between the baseline method (green boundary) and the proposed method (red boundary)

5. Conclusions

In this work, we identified and addressed the limitation of existing methods in accurately localizing discriminative regions for pedestrian attributes. We proposed ALFormer, a transformer-based framework that introduces two novel components Mask Contrast Learning and Attribute Spatial Memory. The former suppresses misleading regional correlations to force focus on intrinsic areas. The latter captures inherent spatial locations for each attribute. Experiments showed state-of-the-art results on two datasets, with ablation studies confirming the efficacy of our modules. The improved localization was further demonstrated through visualizations. Our model provides an effective way to exploit spatial consistency for pedestrian attribute recognition. Future directions include extending the approach to other fine-grained recognition tasks and exploring optimal fusion with global context.

Acknowledgments. This research work was supported by the China State Railway Group CO., Ltd (N2023X005).

References

1. Liu, X., Zhao, H., Tian, M., Sheng, L., Shao, J., Yi, S., ... and Wang, X. (2017). Hydraplus-net: Attentive deep features for pe-destrian analysis. In Proceedings of the IEEE international conference on computer vision (pp. 350-359).
2. Liu, P., Liu, X., Yan, J., and Shao, J. (2018). Localization guided learning for pedestrian attribute recognition. arXiv preprint arXiv:1808.09102.
3. Sarafianos, N., Xu, X., and Kakadiaris, I. A. (2018). Deep imbalanced attribute classification using visual attention aggregation. In Proceedings of the European Conference on Computer Vision (ECCV) (pp. 680-697).
4. Tang, C., Sheng, L., Zhang, Z., and Hu, X. (2019). Improving pedestrian attribute recognition with weakly-supervised multi-scale attribute-specific localization. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 4997-5006).
5. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... and Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929.
6. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., and Jégou, H. (2021, July). Training data-efficient image trans-formers and distillation through attention. In International conference on machine learning (pp. 10347-10357). PMLR.
7. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., ... and Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF international conference on computer vision (pp. 10012-10022).
8. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., and Torralba, A. (2016). Learning deep features for discriminative localization. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2921-2929).
9. Fan, Z., Guan, Y.: Pedestrian attribute recognition based on dual self-attention Mechanism. Computer Science and Information Systems, Vol. 20, No. 2, 793–812. (2023), https://doi.org/10.2298/CSIS220815016F).
10. Kumar, N., Berg, A., Belhumeur, P. N., and Nayar, S. (2011). Describable visual attributes for face verification and image search. IEEE Transactions on Pattern Analysis and Machine Intelligence, 33(10), 1962-1977.
11. Kumar, N., Belhumeur, P., and Nayar, S. (2008). Facetracer: A search engine for large collections of images with faces. In Computer Vision–ECCV 2008: 10th European Conference on Computer Vision, Marseille, France, October 12-18, 2008, Pro-ceedings, Part IV 10 (pp. 340-353). Springer Berlin Heidelberg.
12. Bourdev, L., Maji, S., and Malik, J. (2011, November). Describing people: A poselet-based approach to attribute classification. In 2011 International Conference on Computer Vision (pp. 1543-1550). IEEE.
13. Liu, Z., Luo, P., Wang, X., and Tang, X. (2015). Deep learning face attributes in the wild. In Proceedings of the IEEE international conference on computer vision (pp. 3730-3738).
14. Abdulnabi, A. H., Wang, G., Lu, J., and Jia, K. (2015). Multi-task CNN model for attribute prediction. IEEE Transactions on Multimedia, 17(11), 1949-1959.
15. Hand, E., and Chellappa, R. (2017, February). Attributes for improved attributes: A multi-task network utilizing implicit and explicit relationships for facial attribute classification. In Proceedings of the AAAI conference on artificial intelligence (Vol. 31, No. 1).
16. Han, H., Jain, A. K., Wang, F., Shan, S., and Chen, X. (2017). Heterogeneous face attribute estimation: A deep multi-task learning approach. IEEE transactions on pattern analysis and machine intelligence, 40(11), 2597-2609.
17. Li, J., Zhao, F., Feng, J., Roy, S., Yan, S., and Sim, T. (2018). Landmark free face attribute prediction. IEEE Transactions on Image Processing, 27(9), 4651-4662.

18. Li, D., Chen, X., Zhang, Z., and Huang, K. (2018, July). Pose guided deep model for pedestrian attribute recognition in surveil-lance scenarios. In 2018 IEEE international conference on multimedia and expo (ICME) (pp. 1-6). IEEE.
19. Wang, J., Zhu, X., Gong, S., and Li, W. (2017). Attribute recognition by joint recurrent learning of context and correlation. In Proceedings of the IEEE International Conference on Computer Vision (pp. 531-540).
20. Graves, A., and Graves, A. (2012). Long short-term memory. Supervised sequence labelling with recurrent neural networks, 37-45.
21. Yang, W., Huang, H., Zhang, Z., Chen, X., Huang, K., and Zhang, S. (2019). Towards rich feature discovery with class activation maps augmentation for person re-identification. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 1389-1398).
22. Hou, R., Chang, H., Ma, B., Shan, S., and Chen, X. (2020). Temporal complementary learning for video person re-identification. In Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV 16 (pp. 388-405). Springer International Publishing.
23. Zitnick, C. L., and Dollár, P. (2014). Edge boxes: Locating object proposals from edges. In Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13 (pp. 391-405). Springer International Publishing.
24. Jaderberg, M., Simonyan, K., and Zisserman, A. (2015). Spatial transformer networks. Advances in neural information processing systems, 28.
25. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... and Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 30.
26. Deng, Y., Luo, P., Loy, C. C., and Tang, X. (2014, November). Pedestrian attribute recognition at far distance. In Proceedings of the 22nd ACM international conference on Multimedia (pp. 789-792).
27. Li, D., Zhang, Z., Chen, X., and Huang, K. (2018). A richly annotated pedestrian dataset for person retrieval in real surveillance scenarios. IEEE transactions on image processing, 28(4), 1575-1590.
28. Guo, H., Zheng, K., Fan, X., Yu, H., and Wang, S. (2019). Visual attention consistency under image transforms for multi-label image classification. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 729-739).
29. D. Li, X. Chen, and K. Huang, "Multi-attribute learning for pedestrian attribute recognition in surveillance scenarios," in Proc. IEEE Asia Conf. Pattern Recognit., 2015, pp. 111–115.
30. Z. Tan, Y. Yang, J. Wan, G. Guo, and S. Z. Li, "Relation-aware pedestrian attribute recognition with graph convolutional networks," in Proc. AAAI Conf. Artif. Intell., vol. 34, no. 07, 2020, pp. 12 055–12 062.
31. Z. Tan, Y. Yang, J. Wan, H. Hang, G. Guo, and S. Z. Li, "Attentionbased pedestrian attribute analysis," IEEE Trans. Image Process., vol. 28, no. 12, pp. 6126–6140, 2019.
32. Pengze Liu, Xihui Liu, Junjie Yan, and Jing Shao. Localization guided learning for pedestrian attribute recognition. arXiv preprint arXiv:1808.09102, 2018.
33. Jia J, Chen X, Huang K. Spatial and semantic consistency regularizations for pedestrian attribute recognition[C]//Proceedings of the IEEE/CVF international conference on computer vision. 2021: 962-971.
34. Kai Yu, Biao Leng, Zhang Zhang, Dangwei Li, and Kaiqi Huang. Weakly-supervised learning of mid-level features for pedestrian attribute recognition and localization. arXiv preprint arXiv:1611.05603, 2016.
35. Sarafianos, N., Xu, X., Kakadiaris, I.A., 2018. Deep imbalanced attribute classification using visual attention aggregation, in: ECCV, pp. 680–697.
36. Zhao, X., Sang, L., Ding, G., Han, J., Di, N., Yan, C., 2019. Recurrent attention model for pedestrian attribute recognition, in: AAAI, pp. 9275–9282.

37. Han, K., Wang, Y., Shu, H., Liu, C., Xu, C., Xu, C., 2019. Attribute aware pooling for pedestrian attribute recognition. arXiv preprint arXiv:1907.11837.
38. Tang C, Sheng L, Zhang Z and Hu X (2019). Improving pedestrian attribute recognition with weakly-supervised multi-scale attribute-specific localization. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 4997-5006).

Yuxin Liu received his M.Sc. degree in computer science from Beijing Jiaotong University, Beijing, China, in 2018. He is currently a Research Associate in China Academy of Railway Sciences Corporation Limited. He is also a researcher of the Computer Vision Laboratory of Institute of Computing Technology (ICT-CVLab). His current research interests include image processing, object detection, computing architecture, and deployment.

Mingzhe Wang received the B.S. degrees in mechanical electronics engineering from the Lanzhou Jiaotong University, Lanzhou, China, in 2002. He is currently a Professor with Institute of Computing Technology, China Academy of Railway Sciences Corporation Limited. His research interests include railway informatization.

Chao Li received the M.S. degrees in transportation information engineering and control from the China Academy of Railway Sciences, Beijing, China, in 2008. He is currently a Professor with Institute of Computing Technology, China Academy of Railway Sciences Corporation Limited. His research interests include informatization and intelligence of railway passenger stations and sections.

Shuoyan Liu received his Ph.D. degree in computer science from the Beijing Jiaotong University, Beijing, China, in 2011. She is currently a Associate Professor in China Academy of Railway Sciences Corporation Limited. She is also a researcher of the Computer Vision Laboratory of Institute of Computing Technology (ICT-CVLab). Her research interests include computer vision, object detection, image analysis, and visualization.

Received: October 15, 2023; Accepted: March 13, 2024.

Computer Science and Information Systems 21(4):1583–1605 https://doi.org/10.2298/CSIS240314049L

GCN-LSTM: Multi-label educational emotion prediction based on graph Convolutional network and long and short term memory network fusion label correlation in online social networks

Zhiguang Liu[1,⋆], Fengshuai Li[2], Guoyin Hao[3], Xiaoqing He[1], and Yuanheng Zhang[1]

[1] School of Electronics and Electrical Engineering, Zhengzhou University of Science and Technology
Zhengzhou, 450064 China
snowycry@qq.com

[2] College of Civil and Architectural Engineering, Zhengzhou University of Science and Technology
Zhengzhou, 450064 China
lifsfsfs@163.com

[3] School of Music and Dance, Zhengzhou University of Science and Technology
Zhengzhou, 450064 China
hexiaoqingvip@163.com

Abstract. Although there are a lot of methods for multi-label classification in the past research, there are still many problems. For example, in the real world, labels are not necessarily independent of each other, and there may be some connection between labels. Therefore, exploring and utilizing the interdependence between labels is a key issue in current research. For example, in the photo category, a picture that contains blue sky often also contains white clouds, and in the text category, a political story is less likely to be entertainment news. Therefore, the key to improve the accuracy of multi-label classification is to effectively learn the possible correlations between each label. Therefore, we propose a novel multi-label educational emotion prediction based on graph convolutional network and long and short term memory network fusion label correlation in online social networks. This model uses Word2Vec method to train word vectors and combines graph convolutional neural network (GCN) with long and short term memory network (LSTM). The GCN is used to dig deeper word features of text, the LSTM layer is used to learn the long-term dependence relationship between words, and the multi-granularity attention mechanism is used to assign higher weight to the affective word features. At the same time, label correlation matrix is used to complete the label feature vector and text features as the input of the classifier, and the correlation between labels is investigated. The experimental results on the open data set show that the proposed model has a good classification effect compared with other advanced methods. The research results promote the combination of deep learning and affective computing, and can promote the research of network user behavior analysis and prediction, which can be used in personalized recommendation, targeted advertising and other fields, and has wide academic significance and application prospects.

Keywords: multi-label educational emotion prediction, GCN, LSTM, multi-granularity attention mechanism.

⋆ Corresponding Author

1. Introduction

With the rapid development of Internet technology, data such as pictures and text in education have also shown rapid growth, and the development of data has provided a cornerstone for machine learning. Therefore, multi-label classification learning has become a hot spot in the field of machine learning research, which is mainly reflected in emotion classification, image and video semantic annotation [1,2], gene function analysis, etc. The traditional single-label classification problem, in which an object belongs to only one specific category, however, in most classification tasks, an instance may contain multiple categories at the same time. For example, in a daily text classification task, it may belong to both the entertainment and economic categories; In the image classification task, an image may contain cats, dogs, people, and the sun at the same time. For this kind of multi-label data, the classification task changes from the traditional prediction of one category to the prediction of multiple categories [3,4].

In traditional image classification and recognition, the information contained in the image (such as edge detection, contour detection, local binary mode (LPB), directional gradient histogram (HOG), Haar feature detection) is usually extracted manually to achieve image classification [5,6]. However, these extraction methods not only cost a lot of manpower, but also have many serious problems. For example, the image is affected by external environmental factors, and the change of spatial information such as illumination, rotation and translation of objects leads to the change of object characteristics, resulting in the decline of classification accuracy.

The task of affective prediction is to determine an attitude towards a goal or subject, which can be either a polarity (positive or negative) or an emotional state such as joy, anger, or sadness [7]. Most previous studies have judged the positive or negative attitudes contained in social media content, such as whether customers are satisfied with a product in product reviews, or whether viewers view a movie as exciting or boring in movie reviews. Most of the work on judging emotional states only focus on single-label classification, dividing emotional texts into one emotional category in multiple categories, while ignoring the situation that multiple emotional categories may coexist in a text instance, which is obviously inconsistent with reality [8]. For example, an old man who has three sons but no one to take care of him in his old age may cause the reader to feel sad, angry, or other different emotions.

In recent years, the aspect-level emotion classification model based on deep learning [9] performs well. These models are mainly based on Long Short-Term memory network (LSTM) and Bi-directional LSTM (Bi-LSTM), and introduce attention mechanism and position weight to improve the classification performance. Multi-label classification has been widely used in many fields, such as text classification, image annotation, bioinformatics [10] and so on. At present, the research on multi-label sentiment classification of microblog texts is still in its infancy. The Second Conference on Natural Language Processing and Chinese Computing presented a fine-grained sentiment prediction task for Chinese microblogs for the first time, requiring multiple sentiment tags to be selected from a set of associated tags for each instance of a microblog. In 2019, Parwez et al. [11] used a convolution neural network (CNN) model to synthesize word vectors in microblog sentences into sentence vectors. These sentence vectors were used as features to train a multi-label classifier to complete the multi-label sentiment classification of microblogging on the NLP&CC2013 dataset. This method only extracted the local features of the

text without considering the global semantic information of the text. In 2020, Wang et al. [12] proposed a long short term memory networks (LSTM) model based on hierarchical attention. It used an attention mechanism to represent the text, and then used LSTM for multi-label classification, but this method ignored the local features of the text. In 2020, Acheampong et al. [13] used the FastText model to make emotion prediction on the crawled news text dataset. The experimental results showed that the accuracy rate of the FastText model was higher than that of traditional machine learning methods such as support vector machine and logistic regression, and the speed advantage was more obvious than that of the neural network model. In 20220, Reza et al. [14] uses Transformer multi-head attention mechanism to simultaneously acquire global features and partial information related to specific aspects of the text, which to a certain extent solved the problems that it was difficult for CNN to obtain global semantic information, the RNN training speed was too slow, and the dependence degree between words decreased with the increase of distance. Fine-grained sentiment analysis was performed on the Amazon food review dataset and Booking Hotel review dataset, but the method failed to take into account the relevance of labels.

In terms of LSTM basic network, Kumar et al. [15] adopted two LSTM networks to encode statements from the left and right sides of aspect words respectively. Fadel et al. [16] used multi-layer Bi-LSTM network to model the concatenation of statements and aspect words. Li et al. [17] proposed Target-Specific Transformation Networks-Adaptive Scaling (TNET-AS). The context conversion unit was built based on Bi-LSTM, and the features were extracted from the output by Convolutional Neural Networks (CNNs). However, because LSTM was time-dependent, it could not be able to accurately identify the potential association between distant descriptive words and aspect words when dealing with complex long sentences. Moreover, when multiple aspect words with different emotional polarity appear at the same time, LSTM could mistakenly identify some adjectives or phrases with obvious emotional color as the modifiers of aspect words, resulting in semantic confusion.

Graph Convolutional Networks (GCN) [18] can be seen as an improvement over traditional CNN-encoded unstructured data. Although CNN can handle semantically rich text structures and are easy to parallelize, convolution operation will treat the features of multiple words as continuous words and cannot accurately identify the emotions described by multiple discontinuous words. GCN can solve the problems of LSTM and CNN in processing text data to a certain extent. By convoluting the input statements and their syntactic dependence graphs, GCN can gather the distant descriptive words and aspect words into a smaller range, better combine the syntax information of the sentences, and find the complex dependence relationship between long-term multi-words. Therefore, it has certain advantages in the problem of aspect level emotion classification. Zhu et al. [19] used GCN to build aspect-specific GCN (ASGCN), used Bi-LSTM to capture context information, and multi-layer graph convolution to extract aspect word features, and fed aspect word features back to the hidden layer of Bi-LSTM.

In order to make the model focus on the semantic information related to aspect words, attention mechanism has been introduced in most models in recent years. Jang et al. [20] used word vectors and Bi-LSTM hidden vectors respectively to compute attention iteratively. Kamyab et al. [21] used both word vector and hidden vector to calculate attention in segmented decoding. Li et al. [22] used multi-head attention mechanism to improve

the memory network and alleviate the selective preference of the model. The introduction of attention mechanism can further improve the classification performance. However, the above attention mechanism does not pay attention to the bidirectional influence relationship between a particular aspect and its context, that is, the semantics of a particular aspect are based on the context, and the emotion expression of the context is also based on a specific aspect. In a statement, some words can express semantics and emotions independently (fine-grained), while others need to form phrases with neighboring words to have full semantics (coarse-grained). Based on this situation, Liu et al. [23] proposed Interactive Attention Networks (IAN) to calculate the interaction between words and sentences from a coarse-grained perspective. Puthige et al. [24] proposed an Attention-over-Attention (AOA) neural network to calculate the interaction between words and sentences from a fine-grained perspective. Sun et al. [25] proposed that Multi-grained Attention Network (MGAN). The multi-granularity attention mechanism used two sets of symmetric structures to train the attention weight for each word. It could capture the interaction between aspect words and context at the word level and phrase level, respectively, and used aspect alignment loss to describe the influence relationship between aspect words with the same context.

Bidirectional Encoder Representation from Transformers (BERT) [26] is a pre-trained language model based on the coding part of multi-layer bidirectional Transformer, which has obvious advantages in the semantic expression of text. Liu et al. [27] proposed an attention encoder network based on pre-trained BERT model. Mewada et al. [28] used BERT model to encode statements, and used the pooled aspect word vector to classify the sentiment of statements. The fusion of aspect information made BERT pre-training model more suitable for the fine-grained sentiment classification task of text.

In the task of multi-label emotion prediction and classification, not only the local features and global semantic information of text should be considered, but also the correlation between labels should be fully modeled. In 2023, Ahanin et al. [29] proposed a multi-label sentiment classification method that combined context features and text features. In this method, each sentence in the text was initially classified by emotion, and then the emotion label correlation between each sentence was examined by the emotion transfer relationship between the sentences. Finally, the experiment was carried out on the NLP&CC2013 dataset. But this method neglected the local features of the sentence. Huang et al. [30] proposed to learn the multi-label classification of label features by calculating the cosine similarity of labels, and to examine the correlation of labels by judging whether different labels can share the features between labels. In 2000, Yang Kwon et al. [31] performed K-nearest neighbor retrieval on the word vector space of all labels, and took the first k labels with the nearest cosine distance to the network output vector as the multi-label prediction, verifying the feasibility of the proposed method in label semantic expansibility. Therefore, using the correlation between different labels could effectively improve the performance of multi-label learning.

Some existing multi-label sentiment classification methods fail to extract both local features and global semantic information, and some fail to fully consider the correlation between labels. To solve the two problems, a novel multi-label educational emotion prediction based on graph convolutional network and long and short term memory network fusion label correlation in online social networks model is proposed in this paper. GCN is used to extract the local features of words in text and combine them with word vec-

tors as the input of LSTM. As a special RNN model, LSTM can better solve the long dependency problem [32]. After extracting the long-term dependence between word local features and word vectors, the attention mechanism generates text feature vectors at different time steps according to the contribution of each text fragment and combines them into a complete text feature expression. At the same time, in order to solve the correlation between labels, the label correlation matrix is used to complete the label matrix and concatenate the text feature expression. Finally, the text emotion classification is carried out in the output layer, and the extracted text features and the corresponding label matrix are used as the input of the classifier to enhance the accuracy of the classification.

This paper is structured as follows. In section 2, we detailed introduce the related works. Multi-label educational emotion prediction model is proposed in section 3. Experiments are conducted in section 4. There is a conclusion in section 5.

2. Related Works

2.1. Affective Computing

The research process of emotion computing can be summarized into three aspects: how to describe emotion, namely emotion quantification; How to describe the research object, that is, the characteristics; How to build emotional models. About the quantification of emotion, the current research ideas are generally divided into ”category view” and ”dimension view”. Category view divides, for example, the positive, negative and neutral of text classification and Ekman’s six basic emotions. Dimensional view is divided as PAD model [33].

The feature and emotion computing models are determined according to the object of study. The content of social network mainly includes text and image, and text emotion classification is one of the important research contents in natural language processing. In the early image emotion calculation, a single feature of the image, such as color or texture, was used to establish its mapping relationship with emotion. Then the comprehensive features of the image are used to establish an emotional model. With the development of social networks, the emotional model of social networks has become a research hotspot. The following characteristics of social networks should be considered when constructing the user emotion model of social networks: (1) Dynamic, the network structure and content change dynamically with time; (2) Heterogeneity, including text, images, network structure and other data sources.

2.2. Social Network User Emotion

Studies have been conducted to analyze users’ emotional states based on their behaviors in social networks, such as microblogs, geographical locations, and phone records. Predict user personality based on social network structure and user behavior. A study based on data from Facebook shows that the emotions of users on the social network are closely related to their social activities and interconnection. Sociological research shows that Instinctive Empathy makes emotions collective, that is, how people feel depends on what they are exposed to. Some researchers have established influence models and influence propagation models based on user annotations, microblogs and published articles on the Internet [35].

The emotion based on probability graph model has too many assumptions (artificial empirical formula) about the change of emotion, if the assumption is biased on the data set, it is difficult to achieve good results, and the modeling process often only deals with the binary emotion problem each time, which will be worse in the multi-classification problem. At the same time, these studies do not provide a comprehensive and systematic quantitative model of their own psychology, emotions, external influences and behaviors.

2.3. Multi-label Problem Definition

Suppose the training set has m samples, which is denoted as $X = x_1, x_2, \cdots, x_m$. q labels are denoted as $L = l_1, l_2, \cdots, l_q$. $Y = y1, y_2, \cdots, y_m$ denotes label space. Then multi-label classification is the function $f : X \to Y$ learned by training set $(x_i, y_i)|1 \leq i \leq m$. Where $x_i \in X$ represents a sample, $y_i \in X$ represents the class label to which sample x_i belongs, and y_i is a subset of the label L. By inputting the predicted data into the trained model, the label classification results that are close to the reality can be obtained.

The traditional method is to train a classifier independently for each label, and train each classifier separately with all the data. However, this independent method does not take into account the reciprocity of labels, and the classification effect is often difficult to be satisfactory. The convolutional neural network multi-label classification algorithm based on label correlation proposed in this paper solves this problem by adding the correlation between different labels.

2.4. GCN Method

The co-occurrence relationship between labels is usually represented in the form of a graph. The formula $G =< V, E >$ is used to represent the dependency co-occurrence relationship of the class label node. V indicates all label nodes, and E indicates the dependency between two label nodes. As an effective feature extractor based on graph structure, graph convolutional neural network (GCN) has been widely used in node classification, graph classification, link prediction and so on. The input to GCN contains the node characteristics and the correlation matrix between these nodes. GCN updates the characteristics of a node by aggregating the characteristics of its neighbors and itself into the node. The process of graph propagation can be expressed as follows:

$$X^{l+1} = \alpha^l(\bar{A}X^lW^l). \tag{1}$$

Where, X^l, W^l, and α^l respectively represent the input features, weight parameters, and nonlinear activation functions of the convolution layer of the $l-th$ graph. $\bar{A}$ represents the normalized form of the correlation matrix:

$$\bar{A} = \bar{D}^{-0.5}\bar{A}\bar{D}^{-0.5}. \tag{2}$$

Where $\bar{A} = A + I$. I is the identity matrix. $\bar{D}$ is a diagonal matrix and $\bar{D} = \sum_j \bar{A_{ij}}$.

GCN represents the word vector to be trained as the correlation matrix constructed by the features of the label nodes in the graph and the statistics of the sample labels as the input of the graph convolutional neural network, learns semantic label embedding through the two-layer graph convolutional neural network, and uses it as an interdependent object

classifier in the prediction stage. At the same time, the feature maps extracted by convolutional neural network of sample images are globally maximized, and the resulting image features and label co-occurrence features extracted by convolutional network are fused in the form of matrix multiplication to obtain the final classification result.

3. Proposed Multi-label Educational Emotion Prediction Model

GCN-LSTM is mainly composed of four parts: statement feature vector generation module, aspect feature vector generation module, multi-granularity attention computing module, and emotion classification module integrating aspect level features. The specific block diagram is shown in Figure 1. The statement feature vector generation module in Figure 1 uses the LSTM network to obtain the statement feature vector containing word order information. Aspect feature vector generation module uses the GCN layer constructed in this paper to obtain the aspect feature vector integrating context information by word masking. The multi-granularity attention computing module uses the output of the above two modules as input, and cross-calculates the fine-grained attention and the coarse-grained attention based on double branches by vector product and pooling methods. The multi-granularity attention weight is obtained by concatenating and normalizing the calculation results of the two kinds of granularity attention. The sentence emotion classification module uses the feature vector and multi-granularity attention weight to calculate the final emotion expression of the sentence, and uses the full connection layer and softmax classifier to get the aspect level emotion classification results.

3.1. Semantic Information Expression Based on GCN

In this paper, two word vector generation methods are adopted: static word vector generated based on Stanford University pre-trained Global Vectors for Word Representation (GloVe) and dynamic word vector generated based on Google pre-trained BERT language model. In this paper, we sum the last four hidden layer vectors of the BERT model to get the sub-word vector, and use the embedding vector of the average sub-word to generate an approximate vector for the original word.

We use $x = [x_1, \cdots, x_{\tau+1}, \cdots, x_{\tau+M}, \cdots, x_N]$ denotes a statement of length N mapped word by word to a fixed real-valued vector in a low-dimensional space. The subsequence $[x_{\tau+1}, \cdots, x_{\tau+M}]$ of x represents the aspect word of length M. $x_i \in R^d$ represents the word vector of the word i, and d represents the dimension of the word vector.

Graph convolution describes dependencies between words using adjacencies between nodes in a graph structure. A node in the undirected graph G represents a word, the edge between the nodes represents the dependency between the two words, and the adjacency matrix A of G is the syntactic dependency graph. If the input statement is "World peace leads to economic development.", the dependency diagram for the statement is shown in Figure 2. The values on the subdiagonal of A are all 1. If there is a direct dependence between the two words described by node i and node j, indicating that the two nodes have edges in the graph G, then $A_{ij} = A_{ji} = 1$; Otherwise, $A_{ij} = A_{ji} = 0$.

GCN simply uses aspect word features that extract and fuse contextual information. However, because the mean and variance of the feature vector containing the semantic

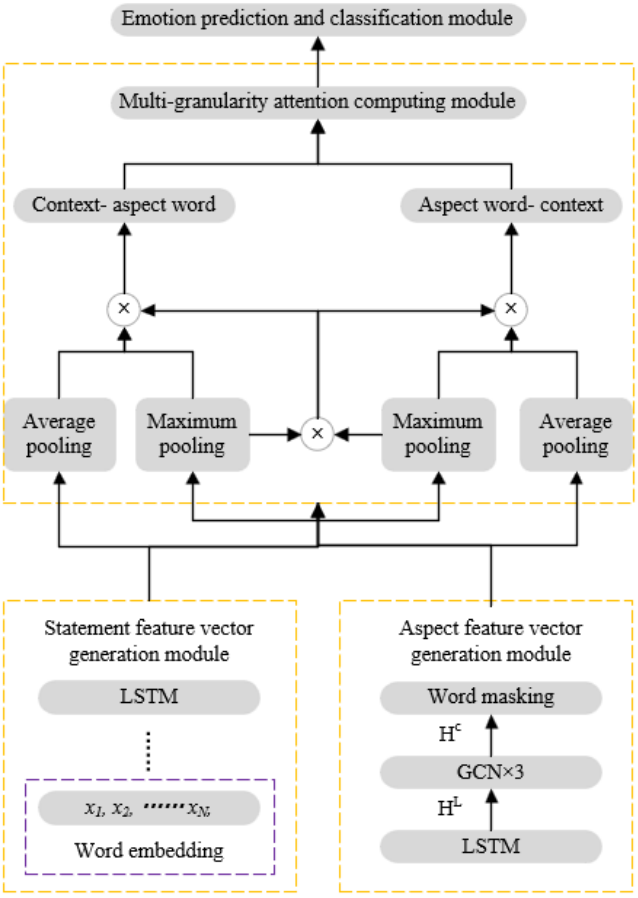


Fig. 1. Proposed model block diagram

.	0	0	0	0	0	0	1
development	0	0	1	0	0	1	0
economic	0	0	1	0	1	0	0
to	0	0	1	1	0	0	0
leads	1	1	1	1	1	1	0
peace	0	1	1	0	0	0	0
World	1	0	1	0	0	0	0
	World	peace	leads	to	economic	development	.

Fig. 2. Dependency graph of the sentence sample

and syntactic information of the statement will change after the nonlinear transformation of the graph convolution, the context information is lost during the forward propagation of GCN. In order to further reduce the lost input information in the output of each layer of graph convolution, this paper uses adaptive scaling strategy [35] to add PD unit to GCN in the aspect feature vector generation module and build CPGCN layer, so as to carry out integrated training on the output of GCN of the local layer and CPGCN of the previous layer. The internal structure of CPGCN layer is shown in Figure 3.

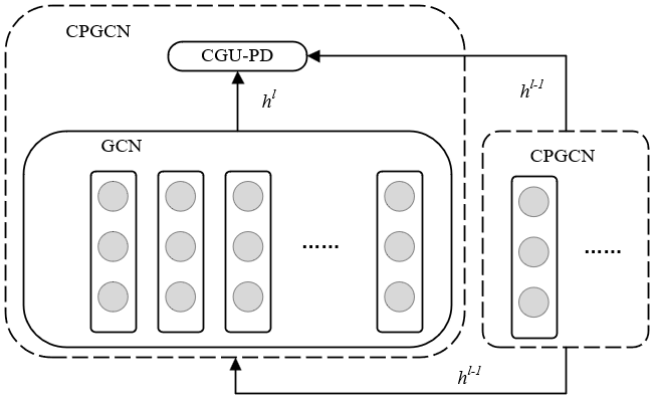


Fig. 3. Internal structure of improved GCN

In Figure 3, h^l is the output of GCN in this layer. If this layer CPGCN is the first layer, h^{l-l} is the output of LSTM hidden layer; Otherwise, h^{l-l} is the output of the previous CPGCN layer.

The CPGCN layer based on CGU-PD and GCN can dynamically adjust the weight of the input vector and the vector transformed by GCN in the output of this layer. The output vector of $l-th$ layer CPGCN is:

$$\eta_i^l = \sigma(W_{gate}^l h_i^l + b_{gate}^l). \tag{3}$$

$$h_i^l = \eta_i^l \odot h_i^l + (1 - \eta_i^l) \odot h_i^{l-1}. \tag{4}$$

Where h_i^l is the output of node i in $l-th$ layer CPGCN. h_i^{l-1} is the output of node i in CPGCN at layer $l-1$. η_i^l is the gating weight of node i in $l-1$ layer CPGCN, which is obtained by h_i^l training. σ is the sigmoid activation function. W_{gate}^l and b_{gate}^l are trainable weight matrices and deviations in the calculation of gated weights, respectively. The $\odot$ dot product represents the multiplication of vectors by their corresponding bits. After CGU-PD transformation, the output of the last CPGCN layer is:

$$H^C = [h_1^C, \cdots, h_{\tau+1}^C, \cdots, h_{\tau+M}^C, \cdots, h_N^C]. \tag{5}$$

Where $h_i^C \in R^{2d_L}$ is the dimension of the hidden state vector of the unidirectional LSTM output. $[h_{\tau+1}^C, h_{\tau+2}^C, \cdots, h_{\tau+M}^C]$ corresponds to the aspect word in the statement.

CPGCN integrates the output of GCN and its upper layer CPGCN to effectively reduce information loss during network propagation and retain the context information of statements more completely. Compared with the common LSTM foundation network, in the final output vector of CPGCN, each word can be fused into the comprehensive semantic, syntactic and interdependent information of a larger range of nodes. Finally, by shielding the non-aspect words in the final output of CPGCN, the aspect feature vector of fusion context information is obtained:

$$H^C_{mask} = [0, \cdots, 0, (h^C_{\tau+1})_{mask}, \cdots, (h^C_{\tau+M})_{mask}, 0, \cdots, 0]. \tag{6}$$

3.2. Context-dependent Learning Based on Deep LSTM Model

Core modules such as user emotion change and influence change relationship are the basis of the entire temporal network. Combined with the advantages of LSTM correlation and easy training, as well as the characteristics of deep network strong expression ability, the deep LSTM module is designed, as shown in Figure 4. The transfer relationship of variables is as follows:

$$z_{t+1} = f^1_{deep}(H_t, X_{t+1}, R_{t+1}). \tag{7}$$

$$r_{t+1} = f^2_{deep}(H_t, X_{t+1}, R_{t+1}). \tag{8}$$

$$\tilde{H_{t+1}} = f^3_{deep}(H_t \cdot r_{t+1}, X_{t+1}, R_{t+1}). \tag{9}$$

$$H_{t+1} = (1 - z_{t+1}) \cdot H_t + z_t \cdot \tilde{H_{t+1}}. \tag{10}$$

Where f^1_{deep}, f^2_{deep}, f^3_{deep} are deep neural networks, which are designed in the form of short-circuit modules. Compared with the classical LSTM, the linear part is replaced by the deep residual neural network, and the state is more compact. The input, R_{t+1} (processed observation data), and the previous state, through f^1_{deep}, f^2_{deep} becomes two activation quantities z_{t+1}, r_{t+1}, which are used to modulate the influence of the state on the new intermediate state $\tilde{H_{t+1}}$ (generated by the deep network f^3_{deep}), and the contribution of the new intermediate state $\tilde{H_{t+1}}$ and state H_t to the final new state, respectively.

X, I represent the observation data and the data after processing. H, A, R represent state variables. f represents all kinds of mapping functions. θ is the model parameter. $X_{u_i,d_m,t}$ represents class d_m observation data of user i at time t, such as uploaded pictures, texts, videos, etc. $X_{u_i,t}$ represents the summary vector of user i observation results at time t, output by f_{AT}. $I_{u_i,u_i}t$ is the interaction between user j and i at time t. For example, j leaves a message to i.

In the aspect of obtaining label correlation, the same way as GCN is used to construct the feature module of extracting label relation. The module uses two layers of GCN learning to embed the label dependencies of the training set. The natural language processing algorithm GLOVE4 is used to generate a 300-dimensional word vector for each class label (there are C class labels in total) to generate the label feature matrix $X \in R^{C\times 300}$. In addition, the correlation matrix $A \in R^{C\times C}$ is constructed based on the statistics of

category labels, that is, the number T_i of the occurrence of the $i-th$ category label o_i in the data set and the number T_{ij} of the co-occurrence of the $i-th$ category label $j-th$ and the j category label o_j are counted. Therefore, conditional probabilities can be used to generate dependencies for class labels:

$$P_{ij} = P(o_i|o_j) = \frac{T_{ij}}{T_J}. \tag{11}$$

Where P_{ij} represents the probability that category label i appears when category label j appears. Therefore, category label correlation matrix A can be defined as:

$$A_{ij} = P_{ij}. \tag{12}$$

GCN learns the relationship between different category labels based on category label feature matrix X and category label correlation matrix A to extract the features of label nodes. Using ReLu as the activation function of the first layer, the working principle of two-layer GCN can be expressed as follows:

$$W = f(X, A) = \bar{A}LeakyRelu(\bar{A}XW^0)W^1. \tag{13}$$

Where, X is the label feature matrix. A is the label correlation matrix. $\bar{A}$ is the normalized form of the label correlation matrix. W^0 and W^1 are the learning parameters of the first and second layer graph convolutional neural networks respectively. W is the interdependent object classifier learned through two layers of GCN.

After label feature matrix X and label correlation matrix A pass through two layers of GCN, the learned interdependent object classifier corresponds to the label co-occurrence embedded classifier $W \in R^{C\times D}$ in the image feature and label relationship feature fusion module. The dimension of the final residual attention feature $\bar{f}$ obtained from the multi-head class specific residual attention module is adjusted to R^D, and then the learned classifier W is applied to the residual attention feature to obtain the prediction score of the multi-label image:

$$\bar{y} = W\bar{f}. \tag{14}$$

Assuming the true label $y \in R^C$ of an image, where $y^i = 0, 1$ indicates whether the label appears in the image, the entire network is trained using traditional multi-label classification losses:

$$L = \sum_{c=1}^{C} y^c \log(\sigma(\bar{y_c})) + (1 - y^c) \log(1 - \sigma(\bar{y_c}). \tag{15}$$

Where $\sigma(\cdot)$ is the sigmoid function.

3.3. A Coarse-grained Attention Computation Based on Two Branches

The two-branch-based coarse-grained attention computation method (CGAC) mainly focuses on the interaction between specific aspect and context at the phrase level, and consists of two sets of symmetrical aspect words (CGAC-avg/CGAC-max)-context and

context-aspect words. In CGAC, the average pooling method is used to extract the comprehensive information of specific aspects and complete sentences, and the maximum pooling method is used to extract the representative information of both. Finally, the vector product is used to allocate attention to words in complete statements or specific aspects. The use of maximum pooling can further reduce the influence of redundant information and noise in feature vectors.

Aspect words-context focuses on describing the degree of attention of different context words to specific aspects, and is used to calculate the coarse-grained attention of each word to a certain aspect. Since a particular aspect may cover multiple words, the average semantic vector is generated by means of average pooling for the aspect feature vector that integrates context information, and the features of all words are integrated. At the same time, the main feature vector is generated by means of maximum pooling. The vector product is used to calculate the average semantic vector, the main feature vector and the semantic correlation between the words in the sentence, so as to allocate attention to all the words. Aspect word-context coarse-grained attention is calculated as follows:

$$s_i^{cA2C-avg} = (H_{mask}^C)_{avg}(h_i^L)^T. \tag{16}$$

$$(H_{mask}^C)_{avg} = \frac{1}{M}\sum_{j=\tau+1}^{\tau+M} h_j^C. \tag{17}$$

$$s_i^{cA2C-max} = (H_{mask}^C)_{max}(h_i^L)^T. \tag{18}$$

$$H_{mask}^C)_{max} = max_{d=0}^{2d_L}(H_{mask}^C)_d. \tag{19}$$

Where, $H_{mask}^C \in R^{2d_L \times N}$ represents an aspect feature vector that fuses context information. $(H_{mask}^C)_{avg}$ represents the average semantic vector of a particular aspect. $H_{mask}^C)_{max}$ represents the main feature vector of a particular aspect. h_i^L represents the eigenvector of the word i that contains the statement word order information. $max_{d=0}^{2d_L}$ indicates the maximum value in each dimension. $s_i^{cA2C-avg}$, $s_i^{cA2C-max}$ represent aspect word-context coarse-grained attention assigned by aspect mean semantic vector and aspect main feature vector for word i, respectively.

Context-aspect words focus on describing the influence of context on the semantic expression of each word in a specific aspect, and are used to calculate the coarse-grained attention of each word in a specific context. In this paper, the average semantic vector is generated by means of average pooling to integrate the semantic information of the whole sentence. At the same time, the maximum pooling method is used to generate the main feature vector, and the vector product is used to calculate the average semantic vector, the main feature vector and the semantic correlation between the words in the specific aspect, so as to allocate attention to the words in the specific aspect. Context-aspect coarse-grained attention is calculated as follows:

$$s_i^{cC2A-avg} = (H^L)_{avg}((h_i^C)_{mask})^T. \tag{20}$$

$$(H^L)_{avg} = \frac{1}{N}\sum_{j=1}^{N} h_j^L. \tag{21}$$

$$s_i^{cC2A-max} = (H^L)_{max}((h_i^C)_{mask})^T. \tag{22}$$

$$(H^L)_{max} = max_{d=0}^{2d_L}(H^L)_d. \tag{23}$$

Where H^L represents the statement feature vector that contains word order information. $(H^L)_{avg}$ represents the average semantic vector of the statement. $H^L)_{max}$ represents the main feature vector of the statement. $(h_i^C)_{mask})$ represents the feature vector of the word i that incorporates contextual information in a particular aspect. $s_i^{cC2A-avg}$ and $s_i^{cC2A-max}$ represent the context-aspect coarse-grained attention assigned by the sentence average semantic vector and the statement main feature vector to the word i in a particular aspect, respectively.

Fine-Grained Attention Computing Method (FGAC) [36] focuses on the interplay of specific aspects and context at the word level. When calculating coarse-grained attention, the pooling method is used to calculate the average semantic vector and the main feature vector, which leads to the loss of some information. In this paper, the aspect-based fine-grained attention computing method uses the product of the eigenvector of aspect and statement to calculate the semantic correlation between the words in the statement and the statement, so as to allocate attention to all words and make up for the information loss caused by coarse-grained attention computing. The aspect-based fine-grained attention is calculated as follows:

$$s_i^{fg} = \sum_{j=1}^{N}(h_j^C)_{mask}(h_i^L)^T = \sum_{j=\tau+1}^{\tau+M}(h_j^C)_{mask}(h_i^L)^T. \tag{24}$$

Where, h_j^C represents the feature vector of the word j that incorporates contextual information in a particular aspect. h_i^L represents the eigenvector of the word i that contains word order information. s_i^{fg} represents the fine-grained attention of the word i.

In this paper, the multi-grained attention of the emotion classification model is obtained by splicing and averaging the coarse-grained attention and fine-grained attention based on double branches. Finally, through the normalization operation of multi-granularity attention, the multi-granularity attention weight is obtained:

$$s_i^{multi} = [s_i^{fg}; [s_i^{cA2C-avg}; s_i^{cA2C-avg}]_{avg}; [s_i^{cA2C-maxg}; s_i^{cA2C-max}]_{avg};]_{avg}. \tag{25}$$

$$\alpha_i = \frac{s_i^{multi}}{\sum_{k=1}^{N} exp(s_i^{multi})}. \tag{26}$$

Where s_i^{multi} represents the multi-grained attention of the word i. $[\cdot]_{avg}$ indicates attention splicing and averaging. α_i represents the multi-granularity attention weight of the word i.

The emotion classification module of GCN-LSTM screens the emotion features related to specific aspects from the LSTM hidden state vector H^L of the statement according to the multi-granularity attention weight α_i. The aspect level emotion expression of the statement is:

$$e = \sum_{i=1}^{N} \alpha_i h_i^L. \tag{27}$$

3.4. Network Training Process

The GCN network training process can be divided into two stages: forward propagation and back propagation. Forward propagation is used to transfer the feature information of the image, and back propagation is used to transfer the error information in reverse to update the network weight.

1. Forward propagation. First, the samples in the data set are output to the network, and the weights, biases and error thresholds of the samples are initialized by using the initiation-V3 model. The output results of forward propagation are calculated by layer by layer transformation through the neural network:

$$x^i = f(W^i x^{i-1} + b^i). \tag{28}$$

   Where x^i represents the output of the current layer, f represents the activation function through relu, W represents the weight, b represents the bias, and x^{i-1} represents the input of the current layer.
2. Back propagation. The core of backpropagation is mainly divided into two steps: the error is transmitted to the previous layer in reverse, and the difference that needs to be updated is calculated according to the learning parameter expression corresponding to the current error. The error function expression of backpropagation is as follows:

$$E^n = 0.5 \sum_{k=1}^{c} (o_k^n - y_k^n)^2 = 0.5 ||o^n - y^n||_2^2. \tag{29}$$

   Where, n is the data sample. c represents the number of output nodes. o represents the expected output value of the training sample. y represents the actual output value of network training. Let $X = (x_i, y_i)|i = 1, 2, \cdots, m$ means that there are m samples. $x_i = [x_{i1}, x_{i2}, \cdots, x_{id}]$ indicates that the $i - th$ sample has d eigenvectors. $y_i = [y_{i1}, y_{i2}, \cdots, y_{iq}]$ means that the $i - th$ sample has q label vectors. If $y_{ij} = 0$, there is no label l_j. In single-label object classification, a picture has a one-to-one relationship with the object label, and a Softmax is used after convolution features. The multi-label classification problem is a one-to-many relationship. Suppose there are c classes of objects, each labeled with probabilities $p_i = (P_{i1}, P_{i2}, \cdots, P_{ic})$. Using the softmax_cross_entropy_with_logits method in tensorflow, we select the largest labels in the probability distribution. According to the calculated co-occurrence relationship between labels, a certain threshold is set. The probability of labels that are larger than the threshold is retained, and the probability of labels that are smaller than the threshold is deleted. Finally, multiple labels are output. The specific flow of Softmax in GCN-LSTM is shown in Figure 4.

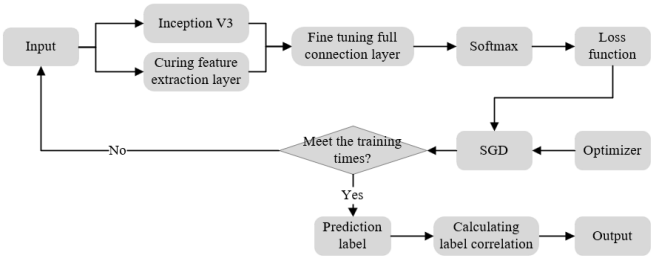


Fig. 4. Specific flow of Softmax in GCN-LSTM

4. Experimental Results and Analysis

4.1. Experimental Environment and Data Set

The experimental operating system is Ubuntul6.04.3LTS,16G memory, Core i5-7200U CPU, and 4G graphics card. The programming language is Python 3, the development tool is Pycharm, and the deep learning framework is Tensorflow-1.10 [37,38].

Since there are few well-labeled multi-label sentiment analysis datasets, the NLP&CC2013 Weibo sentiment analysis dataset is selected in this paper. The data set divides emotions into eight categories: none, dislike, liking, happiness, fear, sadness, anger, and surprise. There are 14000 marked Weibo corpora, each with one or two emotional tags. The training set has 4000 data and the test set has 10000 data. However, in the labeled test set, the distribution of emotional labels is uneven, and there are more than 5000 corpus without emotional labels. Therefore, the data set is re-labeled.

A. Data set annotation rule

Because everyone's criteria for emotion discrimination are inconsistent, it is easy to appear completely different labeling results. Therefore, in the process of annotation, this paper adopts the method of three-person independent annotation and collaborative cross-validation, that is, each person first annotates part of the corpus independently, and then randomly extracts the same number of text corpus from the annotated corpus of the three annotators for cross-evaluation. If the annotation accuracy is above 90%, the annotation results of the extracted text corpus are valid.

B. Data set label distribution

The label distribution of the re-labeled NLP&CC2013 dataset is shown in Table 1. The data in the table is the percentage of the two emotional labels in the total number of labels. This paper does not distinguish the primary and secondary relationship of emotions in the text, but only explores which emotional labels are included in the text. As can be seen from Table 1, about one-third of the text contains only one label, for which a label completion operation is required. Emotional labels of different polarity co-existed less, such as "dislike" and "like", "happy" and "angry", etc., while emotional labels of the same polarity co-existed more, such as "like" and "happy", "disgust" and "angry". Finally, the re-labeled NLP&CC2013 data set is divided and tested. The training set consists of 12000 corpus, the verification set of 1000 corpus, and the test set of 1000 corpus.

Table 1. Emotion label distribution statistics/%

	none	disgust	like	happy	fear	sad	angry	surprise
none	2.4	4.3	4.2	5.7	3.7	5.4	4.5	4.2
disgust			0.3	0.2	4.8	4.6	5.9	3.3
like				9.8	0	0.3	0.1	3.7
happy					0.1	0.1	0	3.5
fear						5.5	6.2	4.1
sad							3.5	4.7
angry								4.9
surprise								

4.2. Experimental Setup

A. Education text preprocessing

It mainly includes text standard processing and word segmentation. Standard processing includes denoising by regular, converting traditional characters into simplified ones and so on. Word segmentation is performed on the re-labeled NLP&cc2013 data set by stuttering word segmentation package. Finally, a dictionary is built to implement text preprocessing. In the process of establishing the dictionary, words that appear less than 5 times are filtered out, and words that do not appear in the dictionary are replaced by $< unk >$.

B. Word vector training

Word embedding model Word2Vec converts words into a low-dimensional real vector by using the relationship between words and context, which can effectively distinguish between a word with multiple meanings or multiple words with one meaning. The Skip-gram model in Word2Vec is used in this article. The default size of the text word vector is 256, the size of the label word vector is 128, and the default value of the maximum distance window of the word vector context is 8. This value can be dynamically adjusted according to the size of the corpus during training. The default value of the maximum number of iterations in stochastic gradient descent is 5, which can be increased for large data.

C. Setting of network model parameters

Table 2. Parameters in GCN

Parameter	value
Word vector dimension	200
Sliding window size	3,4,5
Number of sliding Windows	128
Activation function	ReLU
Epoch	20
Optimizer	Adam

Table 3. Parameters in LSTM

Parameter	value
Word vector dimension	200
Learning rate	0.001
Hidden layer size	256
Loss function	Cross entropy
Epoch	20
Optimizer	Adam

4.3. Evaluation Index

In order to evaluate the performance of GCN-LSTM, four widely used performance evaluation indexes of multi-label learning, including Hamming loss (HL), one-error rate (OE), ranking loss (RL) and average precision (AP), are adopted.

Hamming loss (HL). It is used to evaluate the proportion of inconsistency between the prediction label and the relevant real label. The smaller HL denotes the more accurate the prediction result. The calculation is shown in Equation (30).

$$HL = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{Q}|h(x_i \Delta y_i|. \tag{30}$$

Where Δ represents the symmetric difference between the set of predicted labels and the set of real labels. N is the number of test set samples. $h(x_i$ represents the prediction of sample x_i. Q is the dimension of the label space. y_i indicates the real label.

One-error rate (OE). It is used to evaluate the probability that the first label in the predicted result is not in the set of real labels, calculated as equation (31).

$$OE = \frac{1}{N}\sum_{i=1}^{N}[[arg_{y_i} minY_i \in Z_i]]. \tag{31}$$

Where, $[[\cdot]]$ means that the result returns 0 when the prediction is correct and 1 when the prediction is wrong. Y_i indicates the predicted label result. Z_i is the true label result. $arg_{y_i} minY_i$ indicates the label at the front of the sample. N is the number of test set samples.

Ranking loss (RL). It is used to estimate the average number of times an incorrect label appears before a correct label in a sort sequence in a predicted result label set. The calculation is shown in equation (32).

$$RL = \frac{1}{N} \times \sum_{i=1}^{N}\frac{|(l_k, l_j)|f_k(x_i)| \leq f_j(x_i)|}{|y_i||\bar{y_i}|}. \tag{32}$$

Where $\bar{y_i}$ is the complement of the set y_i in the label space. $f()$ is a real-valued function corresponding to a multi-label classifier. l_k is the prediction label. l_j indicates the true label. N indicates the number of test set samples.

Average precision (AP). It is used to measure the average number of correct sorting in the prediction result label. The calculation is shown in equation (33).

$$AP = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{|y_i|}\sum_{l_k}\frac{A = |l_j|rank(x_i, l_j) \leq rank(x_i, l_k)|}{rank(x_i, l_k)}. \tag{33}$$

Where y_i is the label space set. l_k is the prediction label. l_j indicates the real label. x_i is the prediction label probability. N indicates the number of test set samples. The larger the average accuracy value, the more accurate the prediction result.

4.4. Experiment Results Comparison

In order to evaluate the performance of the model combined with GCN and LSTM, the new model is compared with a single CNN and a single LSTM. During the training process, the condition of early stopping of the model is set, and the model is saved when the validation set $val - acc$ is no longer increased for three rounds. The comparison results of the three models are shown in Figure 5. The new model performs well on the test set.

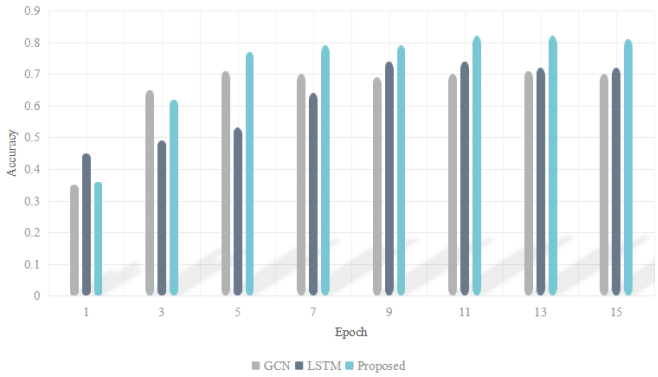


Fig. 5. Accuracy comparison of GCNN, LSTM and proposed.

In order to evaluate the performance of GCN-LSTM model, it is compared with four multi-label classification algorithms including: Unigram+Context [39], FastText [40], Transformer model [41] and PPS [42].

1. Multi-label classification method combining Context and text features (Unigram+Context). This method uses a base classifier to analyze the microblog text and get the initial emotion classification result of each sentence. Then the transfer probability of adjacent sentences is used to modify the sentence emotion category.
2. FastText model. Compared with the traditional word embedding method, this model adds the embedding of character rgram, it can contain the local features of words, and has a very fast training speed. The input to the model is a sequence of words, using hierarchical softmax functions to calculate probabilities on predefined classes, and using cross entropy to calculate losses.

3. Transformer model. The model consists of encoder and decoder. Different from the traditional CNN and RNN, self-attention can effectively solve the problem of long distance dependent features in NLP tasks.
4. PPS method. Compared with the proposed method in this paper, the complete process of label matrix is less, and the correlation between labels is not considered.

The comparison results of the five methods are shown in Figure 6 and Table 4. It can be seen that, on the test set, the method that uses word unary grammar features as the main features and combines context to predict microblog emotion is less effective than the neural network method, and the accuracy rate is about 70%. The PPS method has an accuracy rate of 76%. FastText convergence speed is faster, the accuracy is lower than GCN-LSTM method, up to 72%. The convergence rate of Transformer method is slower than other methods, and the final accuracy rate is 73%. However, the method in this paper, GCN-LSTM, can slightly improve the accuracy of the whole process of adding label related information, and the final accuracy rate is 79%.

Table 4. Comparison with different methods

Method	1	3	5	7	9	11	13	15
PPS	0.19	0.42	0.69	0.75	0.74	0.72	0.73	0.74
FastText	0.24	0.61	0.70	0.69	0.72	0.69	0.68	0.70
Transformer	0.10	0.29	0.45	0.55	0.65	0.66	0.75	0.71
GCN-LSTM	0.12	0.43	0.71	0.75	0.77	0.79	0.76	0.75

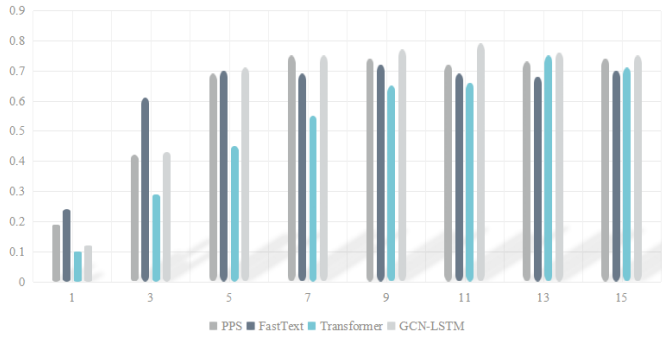


Fig. 6. Comparison with different methods.

In Table 5, the smaller value of "↓" denotes the better classification effect, and the larger value of "↑" denotes the better classification effect. As can be seen from Table 5, the GCN-LSTM model proposed in this paper performs relatively well in terms of average accuracy (AP) and Hamming loss (HL). Compared with the method of using neural network, the method of using unary feature and combining context has some shortcomings

in each index. The FastText method can obtain local word order information using N-gram. The Transformer method addresses sentence length dependency. The deep learning method PPS not only extracts local features, but also extracts global semantic information, and can complete multi-label sentiment classification (MLC) more accurately. These four methods do not consider the correlation between labels, but the GCN-LSTM model proposed in this paper can combine the local features of text with the global semantic information, and also consider the correlation between labels. The experimental results show that GCN-LSTM has a good multi-label classification effect.

Table 5. Experimental comparison results

Model	HL↓	OE↓	RL↓	AP↑
Unigram+Context	0.264	0.283	0.238	0.693
FastText	0.243	0.328	0.265	0.714
Transformer	0.239	**0.213**	0.206	0.725
PPS	0.241	0.297	**0.192**	0.750
GCN-LSTM	**0.224**	0.269	0.276	**0.784**

5. Conclusion

In this paper, we propose a multi-label classification method called GCN-LSTM, which combines GCN and LSTM neural networks to extract local features and global semantic information effectively. The Attention mechanism is used to assign higher weight to educational sentiment words. At the same time, the label correlation matrix corresponding to each text instance is completed by using the label correlation matrix, and the correlation between labels is investigated. That is, if only one emotion label is marked in a text, whether the emotion label with greater correlation is suitable for the text. Label completion of a single emotion label is beneficial to improve the accuracy of multi-label classification. Experiments show that the model presented in this paper performs well on the re-labeled NLP&CC2013 dataset. Future works will focus on more advanced method and apply them in real engineering applications.

Acknowledgments. This work was supported by the Key Soft Science Research Project of Henan Provincial Science and Technology Department in 2023 (232400411183).

References

1. Liu X, Li N, Xia Y. Affective image classification by jointly using interpretable art features and semantic annotations[J]. Journal of Visual Communication and Image Representation, 2019, 58: 576-588.
2. Bayrakdar S, Yucedag I, Simsek M, et al. Semantic analysis on social networks: A survey[J]. International Journal of Communication Systems, 2020, 33(11): e4424.
3. Tamil Priya D, Divya Udayan J. Transfer learning techniques for emotion classification on visual features of images in the deep learning network[J]. International Journal of Speech Technology, 2020, 23: 361-372.

4. Dias L L, Barrére E, de Souza J F. The impact of semantic annotation techniques on content-based video lecture recommendation[J]. Journal of Information Science, 2021, 47(6): 740-752.
5. Yin S. Object Detection Based on Deep Learning: A Brief Review[J]. IJLAI Transactions on Science and Engineering, 2023, 1(02): 1-6.
6. Jiang, Y., Yin, S.: Heterogenous-view Occluded Expression Data Recognition Based on Cycle-Consistent Adversarial Network and K-SVD Dictionary Learning Under Intelligent Cooperative Robot Environment. Computer Science and Information Systems, vol. 20, no. 4, 2023. https://doi.org/10.2298/CSIS221228034J
7. Zhang H, Xu M. Weakly supervised emotion intensity prediction for recognition of emotions in images[J]. IEEE Transactions on Multimedia, 2020, 23: 2033-2044.
8. Liu T, Wan J, Dai X, et al. Sentiment recognition for short annotated GIFs using visual-textual fusion[J]. IEEE Transactions on Multimedia, 2019, 22(4): 1098-1110.
9. L. Teng et al., "FLPK-BiSeNet: Federated Learning Based on Priori Knowledge and Bilateral Segmentation Network for Image Edge Extraction," in IEEE Transactions on Network and Service Management, vol. 20, no. 2, pp. 1529-1542, June 2023, doi: 10.1109/TNSM.2023.3273991.
10. Zhang H, Xu D, Luo G, et al. Learning multi-level representations for affective image recognition[J]. Neural Computing and Applications, 2022, 34(16): 14107-14120.
11. Parwez M A, Abulaish M. Multi-label classification of microblogging texts using convolution neural network[J]. IEEE Access, 2019, 7: 68678-68691.
12. Wang Q, Hao Y. ALSTM: An attention-based long short-term memory framework for knowledge base reasoning[J]. Neurocomputing, 2020, 399: 342-351.ntreal, Canada[J]. International Journal of Transportation Science and Technology, 2022, 11(2): 298-309.
13. Acheampong F A, Wenyu C, Nunoo-Mensah H. Text-based emotion detection: Advances, challenges, and opportunities[J]. Engineering Reports, 2020, 2(7): e12189.
14. Reza S, Ferreira M C, Machado J J M, et al. A multi-head attention-based transformer model for traffic flow forecasting with a comparative analysis to recurrent neural networks[J]. Expert Systems with Applications, 2022, 202: 117275.
15. Kumar A, Verma S, Sharan A. ATE-SPD: simultaneous extraction of aspect-term and aspect sentiment polarity using Bi-LSTM-CRF neural network[J]. Journal of Experimental & Theoretical Artificial Intelligence, 2021, 33(3): 487-508.
16. Fadel A S, Saleh M E, Abulnaja O A. Arabic aspect extraction based on stacked contextualized embedding with deep learning[J]. IEEE Access, 2022, 10: 30526-30535.
17. Li X, Bing L, Lam W et al. Transformation networks for target-oriented sentiment classification. In: Proceedings of the 56th annual meeting of the association for computational linguistics (ACL), Melbourne, VIC, Australia, 15-20 July 2018, pp. 946-956. Stroudsburg, PA: ACL.
18. Zhang S, Tong H, Xu J, et al. Graph convolutional networks: a comprehensive review[J]. Computational Social Networks, 2019, 6(1): 1-23.
19. Zhu X, Zhu L, Guo J, et al. GL-GCN: Global and local dependency guided graph convolutional networks for aspect-based sentiment classification[J]. Expert Systems with Applications, 2021, 186: 115712.
20. Jang B, Kim M, Harerimana G, et al. Bi-LSTM model to increase accuracy in text classification: Combining Word2vec CNN and attention mechanism[J]. Applied Sciences, 2020, 10(17): 5841.
21. Kamyab M, Liu G, Adjeisah M. Attention-based CNN and Bi-LSTM model based on TF-IDF and glove word embedding for sentiment analysis[J]. Applied Sciences, 2021, 11(23): 11255.
22. Li W, Qi F, Tang M, et al. Bidirectional LSTM with self-attention mechanism and multi-channel features for sentiment classification[J]. Neurocomputing, 2020, 387: 63-77.
23. Liu M, Zhou F Y, He J K, et al. Self-attention networks and adaptive support vector machine for aspect-level sentiment classification[J]. Soft Computing, 2022, 26(18): 9621-9634.
24. Puthige I, Hussain T, Gupta S, et al. Attention Over Attention: An Enhanced Supervised Video Summarization Approach[J]. Procedia Computer Science, 2023, 218: 2359-2368.

25. Sun J, Han P, Cheng Z, et al. Transformer based multi-grained attention network for aspect-based sentiment analysis[J]. IEEE Access, 2020, 8: 211152-211163.
26. Ji Y, Zhou Z, Liu H, et al. DNABERT: pre-trained Bidirectional Encoder Representations from Transformers model for DNA-language in genome[J]. Bioinformatics, 2021, 37(15): 2112-2120.
27. Liu H, Liu Y, Wong L P, et al. A hybrid neural network bert-cap based on pre-trained language model and capsule network for user intent classification[J]. Complexity, 2020, 2020: 1-11.
28. Mewada A, Dewang R K. SA-ASBA: A hybrid model for aspect-based sentiment analysis using synthetic attention in pre-trained language BERT model with extreme gradient boosting[J]. The Journal of Supercomputing, 2023, 79(5): 5516-5551.
29. Ahanin Z, Ismail M A, Singh N S S, et al. Hybrid feature extraction for multi-label emotion classification in English text messages[J]. Sustainability, 2023, 15(16): 12539.
30. Huang J, Li G, Huang Q, et al. Learning label-specific features and class-dependent labels for multi-label classification[J]. IEEE transactions on knowledge and data engineering, 2016, 28(12): 3309-3323.
31. Kwon O W, Lee J H. Web page classification based on k-nearest neighbor approach[C]//Proceedings of the fifth international workshop on on Information retrieval with Asian languages. 2000: 9-15.
32. Han K, Wang Y, Tian Q, et al. Ghostnet: More features from cheap operations[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 1580-1589.
33. Yoshimura T, Ojima M, Arai Y, et al. Three-dimensional self-organized microoptoelectronic systems for board-level reconfigurable optical interconnects-performance modeling and simulation[J]. IEEE journal of selected topics in quantum electronics, 2003, 9(2): 492-511.
34. Jisi A and Shoulin Yin. A New Feature Fusion Network for Student Behavior Recognition in Education [J]. Journal of Applied Science and Engineering. vol. 24, no. 2, pp.133-140, 2021.
35. Yang Q, Li Y, Gao X D, et al. An adaptive covariance scaling estimation of distribution algorithm[J]. Mathematics, 2021, 9(24): 3207.
36. Lv Z, Qiao L, Singh A K, et al. Fine-grained visual computing based on deep learning[J]. ACM Transactions on Multimidia Computing Communications and Applications, 2021, 17(1s): 1-19.
37. Liu Z, Jiang D, Zhang C, et al. A novel fireworks algorithm for the protein-ligand docking on the autodock[J]. Mobile Networks and Applications, 2021, 26: 657-668.
38. Liu Z, Zhang C, Zhao Q, et al. Comparative study of evolutionary algorithms for protein-ligand docking problem on the AutoDock[C]//Simulation Tools and Techniques: 11th International Conference, SIMUtools 2019, Chengdu, China, July 8-10, 2019, Proceedings 11. Springer International Publishing, 2019: 598-607.
39. Suciati A, Budi I. Aspect-based sentiment analysis and emotion detection for code-mixed review[J]. International Journal of Advanced Computer Science and Applications, 2020, 11(9).
40. Riza M A, Charibaldi N. Emotion Detection in Twitter Social Media Using Long Short-Term Memory (LSTM) and Fast Text[J]. International Journal of Artificial Intelligence & Robotics (IJAIR), 2021, 3(1): 15-26.
41. Lian Z, Liu B, Tao J. CTNet: Conversational transformer network for emotion recognition[J]. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2021, 29: 985-1000.
42. Zitouni M S, Park C Y, Lee U, et al. LSTM-Modeling of Emotion Recognition Using Peripheral Physiological Signals in Naturalistic Conversations[J]. IEEE Journal of Biomedical and Health Informatics, 2022, 27(2): 912-923.

Zhiguang Liu is with the School of Electronics and Electrical Engineering, Zhengzhou University of Science and Technology. Research direction:image processing, education data analysis, artificial intelligence.

Fengshuai Li is with the College of Civil and Architectural Engineering, Zhengzhou University of Science and Technology. Research direction: big data processing, pattern recognition.

Guoyin Hao is with the School of Music and Dance, Zhengzhou University of Science and Technology. Research direction: Image processing, data analysis, artificial intelligence.

Xiaoqing He is with the School of Electronics and Electrical Engineering, Zhengzhou University of Science and Technology. Research direction:image processing, education data analysis, artificial intelligence.

Yuanheng Zhang is with the School of Electronics and Electrical Engineering, Zhengzhou University of Science and Technology. Research direction:image processing, education data analysis, data analysis, artificial intelligence.

Received: March 14, 2024; Accepted: June 13, 2024.

Computer Science and Information Systems 21(4):1607–1627 https://doi.org/10.2298/CSIS240314050W

A multi-feature Fusion Model Based on Long and Short Term Memory Network and Improved Artificial Bee Colony Algorithm for English Text Classification

Tianying Wen

Department of Education, Liaoning National Normal College
No. 45, Chongdong Road, Huanggu District, Shenyang, 110032, China
sarkozyteague@foxmail.com

Abstract. The traditional methods of English text classification have two disadvantages. One is that they cannot fully represent the semantic information of the text. The other is that they cannot fully extract and integrate the global and local information of the text. Therefore, we propose a multi-feature fusion model based on long and short term memory network and improved artificial bee colony algorithm for English text classification. In this method, the character-level vector and word-level vector representations of English text are calculated using a pre-training model to obtain a more comprehensive text feature vector representation. Then the multi-head attention mechanism is used to capture the dependencies in the text sequence to improve the semantic understanding of the text. Through feature fusion, the channel features are optimized and the spatial features and time series features are combined to improve the classification performance of the hybrid model. In the stage of network training, the weighted linear combination of maximum Shannon entropy and minimum cross entropy is used as the return degree evaluation function of the bee colony algorithm, and the scale factor is introduced to adjust the solution search strategy of leading bees and following bees, and the improved artificial bee colony algorithm is combined with the classification network to realize the automatic optimization and adjustment of network parameters. Experiments are carried out on public data set. Compared with traditional convolutional neural networks, the classification accuracy of the new model increases by 2% on average, and the accuracy of data set increases by 2.4% at the highest.

Keywords: English text classification, multi-feature fusion, artificial bee algorithm, long and short term memory network, multi-head attention mechanism.

1. Introduction

Text classification is a multi-domain technology that spans information retrieval, machine learning and natural language processing, and is an important research direction of information processing and data mining. The main goal is to automatically assign the classified text to the category according to the content features or attribute features of the text when the category is defined in advance [1]. According to the length of the text, text classification is divided into short text classification and long text classification, and the number of short text characters usually does not exceed 200 [2].

Text classification methods based on deep learning can be divided into word level model, character level model and word mixed level model. Word-level model focuses on

using word-level features to accurately express sentence meaning [3,4]. Because English text has a space as a separator between words, the word-level model shows excellent results in English text classification. For Chinese text, because Chinese text does not contain similar separators, the word-level model must have excellent word segmentation ability, otherwise it will lead to wrong classification results. Based on the character-level model, characters are the main features to avoid the word segmentation problem [5,6], and at the same time, the adverse effects caused by excessive word segmentation results can be solved. However, because Chinese characters have many meanings, the character-level model may not be able to accurately express the meaning of sentences.

Recurrent neural network (RNN) and convolutional neural network (CNN) are the main neural network models commonly used in text classification [7,8]. Recurrent neural networks are a kind of network for modeling sequence data. Due to problems such as gradient disappearance and gradient explosion, a variant of long short-term memory (LSTM) is often used [9]. Since LSTM can only learn the global temporal features of the text and cannot learn the local spatial features of the text, CNN is generally used to learn the local features first, and then LSTM is combined to learn the temporal features. However, the spatial dimension of the single-channel word embeddings used by the existing hybrid models is low, and the feature representation is single. As a result, the one-dimensional convolutional neural network cannot give full play to the spatial feature learning ability, which affects the classification performance of the model.

In recent years, the mixed-level model based on words and words has been proved to be an effective way and has been widely used in natural language processing, such as Chinese question answering [10], text classification, etc. Tao et al. [11] proposed a Radicalaware Attention-based Four-Granularity (RAFG) model, which combined characters, words, character-level roots and word-level roots to implement text classification. In addition, RAFG employed serialized bidirectional LSTM and attention mechanism to capture and integrate features. Hao et al. [12] proposed a Mutual-Attention Convolutional Neural Network (MACNN), which generated two alignment information matrices with two-level features through word and character features, and used convolutional neural networks to generate integrated features to improve the classification performance of English texts. However, due to the existence of one-word polysemy in English characters, these mixed-level models did not solve the problem of inaccurate character-level feature representation. Therefore, this paper proposes a multi-feature fusion model based on LSTM and improved artificial bee colony (ABC) algorithm for English text classification. Our main contributions are as follows.

1. In this method, the character-level vector and word-level vector representations of English text are calculated using a pre-training model to obtain a more comprehensive text feature vector representation.
2. Then the multi-head attention mechanism is used to capture the dependencies in the text sequence to improve the semantic understanding of the text. Through feature fusion, the channel features are optimized and the spatial features and time series features are combined to improve the classification performance of the hybrid model.
3. In the stage of network training, the weighted linear combination of maximum Shannon entropy and minimum cross entropy is used as the return degree evaluation function of the bee colony algorithm, and the scale factor is introduced to adjust the solution search strategy of leading bees and following bees, and the improved artificial

bee colony algorithm is combined with the classification network to realize the automatic optimization and adjustment of network parameters.
4. A large number of experimental results on public data sets show that the proposed method achieves good classification results.

This paper is structured as follows. In section 2, we detailed introduce the related works. Multi-label educational emotion prediction model is proposed in section 3. Experiments are conducted in section 4. There is a conclusion in section 5.

2. Related Works

Traditional machine learning algorithms [13] often need feature selection in text classification, while deep learning algorithms are widely used because they can automatically carry out feature learning. Common structures include CNN and RNN, which are suitable for processing time series data and are widely used in text classification. Jin et al. [14] proposed three models based on LSTM to deal with text classification under multi-task learning. Senthil et al. [15] used bidirectional LSTM combined with feed-forward neural network for emotion analysis. Since LSTM can only output the features of the last moment and cannot make full use of the features of each moment, some scholars try to use attention mechanism to optimize the feature representation of LSTM. Lin et al. [16] used the attention mechanism to weight the features of each moment of LSTM, achieving good results in the emotion classification task. Hu et al. [17] introduced Multi-head Attention into bi-directional LSTM for emotion classification, and achieved better results than bi-directional LSTM. Because RNNs cannot learn spatial features and the training time is long, CNNs are used in the text field. Zhoum et al.[18] used CNN for text classification, adopted multi-path convolution to extract spatial features and global maximum pooling to retain the most important features, and verified the practicability of CNN in the field of text classification through experiments. Since global maximum pooling is prone to cause a large number of feature loss, Chen et al. [19] proposed a dynamic pooling idea, which adopted different K values in different pooling layers to preserve the first K maximum features, effectively solving the problem of serious feature loss in global maximum pooling. Jia et al. [20] applied capsule neural network to text classification, and achieved better results than classical CNN on some data sets. Bang et al. [21] tried to combine the attention mechanism in CNN, which effectively improved the ability of CNN to learn local features.

Because CNN and RNN have different emphasis, many scholars put forward a hybrid model combining the advantages of both. Huang et al. [22] proposed a recurrent convolutional neural network (RCNN), which used a bidirectional cycle structure to model the context of features and realized the core idea of convolution. Moradzadeh et al. [23] put forward the hybrid model C-LSTM, which provided the mode of combining CNN and RNN. Reference [24] used multiple convolutions to learn spatial features, and learned temporal features through LSTM after fusion. Reference [25] extracted more abstract spatial features through stacked convolution pooling layers for each route, and then combined LSTM for time series feature learning after fusion. The LSTM-CNN [26] explored the feature learning mode of time sequence before space. On this basis, the BRCAN model proposed by Zhou et al. [27] used bidirectional LSTM to learn context information, and

then combined CNN and attention mechanism to weight key features, achieving good classification effect on multiple data sets. Hasib et al. [28] explored a variety of attention mechanisms and conducted a comprehensive comparative evaluation. The C-HAN model proposed by Long et al. [29] divided text representation into two stages: word-sentence and sentence-document, and compared the effects of word vector and word vector on model performance. The TSOHHAN model proposed by Li et al. [30] combined the role of title in topic classification and achieved a classification accuracy rate superior to that of traditional hierarchical attention networks. Different from the small-scale shallow neural network model proposed by the above scholars, the Google team proposed a pre-trained language model BERT [31], which had achieved excellent results in a number of NLP tasks.

In terms of feature extraction, deep learning methods employ nonlinear activated neural networks to process large amounts of input data, avoiding heavy and time-consuming feature engineering. Recurrent neural networks and convolutional neural networks are widely used in the field of text classification. Kim et al. [32] applied CNN to text classification tasks for the first time and proposed a Text Convolutional Neural Network (TextCNN). TextCNN used convolution kernel of different sizes to extract local static features of text, which could effectively extract local text information. Based on TextCNN architecture, Xu et al. [33] proposed a new CNN model. First, dynamic maximum pooling was used to capture richer information in text. Secondly, in order to further improve the classification accuracy, a hidden bottleneck layer was added after the pooling layer. Ferjani et al. [34] proposed a word-level Deep Pyramid Convolutional Neural network, which enabled the model to capture long-distance dependencies in text by stacking convolutional layers and maximum pooling layers. However, the convolution operation could not take into account the text position information of the sequence. Although RNN can extract text context information and consider text position information, there is gradient explosion problem in the training process, which will seriously affect the experimental results.

In recent years, attention mechanism has been widely used in the field of text classification. Ding et al. [35] proposed a directed self-attention network, which achieved good experimental results by using only the attention mechanism without using CNN and RNN structures. Meng et al. [36] combined CNN and the attention mechanism, used the attention mechanism to calculate the weight of words with higher relevance to text semantics, and further extracted semantic features through CNN. Zhang et al. [37] proposed a Coordinated CNN-LSTM-Attention (CCLA) model. The model first learned the text sequence representation using CCLA, and then obtained the emotional tendencies in the text through the classifier.

In terms of text representation, Gong et al. [38] proposed a static word vector representation model, word2vec. Different from the previous high-dimensional sparse one-hot vector, this model could map text sequences to low-dimensional dense vectors, while also taking into account the contextual semantic information of the text in which the sequences were located.

Pre-trained models have also achieved good results in text representation. Lin et al. [39] proposed a pre-training model Bert based on Transformer. The model training task was divided into two steps. The first step was unsupervised pre-training using large-scale corpus. The second step was to fine-tune the model to accommodate different NLP

tasks. Li et al [40] proposed Enhanced Language Representation with Informative Entities (ERNIE) on the basis of Bert. ERNIE improved the mask semantic model by using global information to predict parts of the mask, and by using the information entities in the knowledge graph, so that the model could better learn the complete semantic representation. Peng et al. [41] proposed ERNIE2.0 for sustainable learning. The model used a large number of corpus models for unsupervised training tasks, and then used pre-training tasks to continuously update the model to further improve the language representation ability of the model.

Although scholars have proposed a variety of hybrid models, the existing hybrid models still have the following problems.

1. Single-channel word embedding is commonly used, the spatial dimension is low, the feature representation of text is single, and only one-dimensional convolution algorithm can be used on a single channel, which cannot give full play to the spatial feature learning ability of convolution;
2. When the existing CNN-RNN hybrid model fuses multi-channel convolution features, the feature timing after fusion is often damaged, which affects the subsequent LSTM layer's learning process of timing features.

Therefore, in order to make full use of dual-channel features, the model proposed in this paper first learns spatial features independently in the two channels, and then uses point-by-point convolution fusion channel features to enhance the spatial feature learning ability of the convolution layer. When fusing multi-path convolutional features, bidirectional LSTM combined with attention mechanism is used in each path for timing feature learning, and the features of each path are concatenated to represent text, effectively avoiding the problem that the multi-path convolutional feature fusion process will cause damage to the timing feature after fusion before entering LSTM. At the same time, we use an improved ABC to optimize the network parameters and further optimize the effect. Experiments show that the proposed text classification model achieves good classification performance on multiple data sets.

3. Proposed English Text Classification Model

Although the traditional word mixing model has a good ability to express features, the character-level features fail to take into account the polysemy of English characters. Therefore, this paper proposes a English text classification method with multi-feature fusion model based on long and short term memory network and improved artificial bee colony algorithm to achieve the task of English text classification, as shown in Figure 1.

3.1. Input Layer

The input layer mainly solves the problem of text sequence partitioning. Consider a English sentence S, which is divided into two distinct sequences: word-level $S_{word} = w_1, w_2, \cdots, w_n$ and character-level $S_{char} = c_1, c_2, \cdots, c_t$. Where n and t are the sequence lengths obtained by dividing sentences according to word and word respectively. In order to better illustrate the problem of sequence partitioning, taking "Today we see the film and eat watermelon" as an example, Figure 2 shows the results of word-level sequence and character-level sequence partitioning in English text.

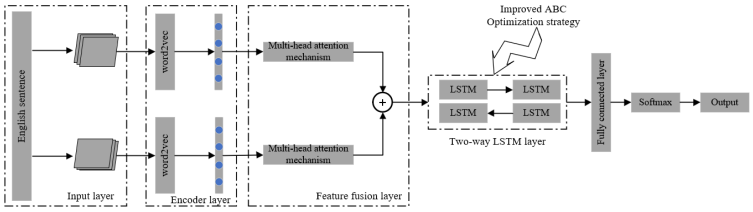


Fig. 1. Proposed English text classification model

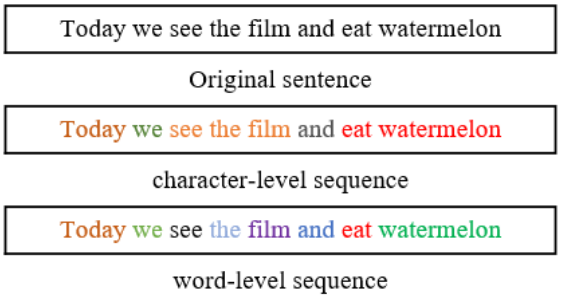


Fig. 2. Text sequence division

3.2. Encoder Layer

The main function of the encoding layer is to represent the sequence of text from the input layer in a continuous space. It receives two levels of features, the word vector representation and the word vector representation, (i.e. S_{char} and S_{word}) and outputs two embedded matrices through a pre-trained model.

Traditional language representation methods are static and can not solve the problem of polysemy in English text. Therefore, this paper uses ERNIE to complete the character-level vector representation of text, and optimizes the traditional static representation method to dynamic representation method. ERNIE improves the mask strategy on the basis of BERT, and introduces the enhanced language representation model of knowledge graph, which can complete the vectoring representation of Chinese characters more accurately. The ERNIE model adopts multi-layer Transformer module and uses bidirectional Transformer structure for text vectorization. The text sequence S_{char} is first entered into multiple self-attention layers to obtain contextual information. Then, the context information is input into the Normalize layer for residual connection and normalization operations, and then the linear change is processed through the feed-forward neural network. After repeating the previous step, the dynamic word vector representation is obtained by combining the knowledge external entity information and the prior semantic information of the masking strategy. Through the vector representation obtained by training, the coding layer converts the character-level text sequence S_{char} into a character-level vector $e_{char} = [e^1_{char}, e^2_{char}, \cdots, e^t_{char}]$.

In this paper, the common word2vec model is used to calculate the word-level vector of text. word2vec is a model for training static word vectors that can map high-

dimensional, sparse one-hot vectors to low-dimensional, dense word vectors. word2vec has two training modes [42], namely CBOW and Skip-gram. Because Skip-gram is better than CBOW in learning rare words, Skip-gram is more accurate. Therefore, this paper uses Skip-gram mode to train word vectors. By training the word vector, the input text sequence S_{word} is transformed into the word level vector $e_{word} = [e^1_{word}, e^2_{word}, \cdots, e^n_{word}]$.

3.3. Feature Fusion Layer

The feature layer aims to generate a comprehensive feature representation of the input text S by combining contextual word-level features with character-level features. Since the word-level and character-level embedding matrices are located in two independent Spaces, direct fusion of features along time series will damage text representation information, so this paper adopts two independent feature layers to extract character-level and character-level text features respectively. The feature layer consists of the attention layer and the convolution layer, and the details are as follows.

The attention layer. In order to integrate features, highlight the features with high relevance to the text semantics, and enhance their weight. This paper uses Transformer [43] multi-head attention encoder to capture full text context information. Multiple attention is calculated using multiple attention, and then the results of each attention are spliced to obtain different levels of semantic information. The calculation is shown in formula (1):

$$h_i = softmax(\frac{QK^T}{\sqrt{d_k}})V. \tag{1}$$

Where, Q, K, V are the weighted matrix, d is the dimension of the vector, softmax function normalizes the output result row.

Next, combine each self-attention output to obtain multi-head attention output, which is calculated as shown in formula (2):

$$Multihead = Concat(h_1, h_2, \cdots, h_y)W^T. \tag{2}$$

Where y is the total amount of self-attention and W^T is the matrix for transforming the dimensions of the concatenation results.

The convolution layer. In order to prevent semantic information loss in the attention layer, the feature extracted from the character level vector and word level vector and the text embedding matrix in the coding layer are folded into two two-dimensional tensors respectively. Similar to a picture with three channels, these stacked feature tensors can be viewed as an image with two channels. After stacking features into two-dimensional tensors, the convolution layer can extract feature information of different dimensions from multi-channel feature vectors.

The convolutional layer is composed of CNN, Relu function and one-dimensional maximum pooling, and its structure is shown in Figure 3. It has a strong local feature extraction ability, which can capture important phrase features. Moreover, this convolution layer uses multiple convolution kernels for feature mapping, which can extract more feature information than a single convolution kernels. In a convolutional neural network, a sub-feature c_i is calculated by a filter w and a feature window $z_{i:i+k-1}$, as shown in the formula (3):

$$c_i = f(w * z_{i:i+k-1} + b). \tag{3}$$

Where b is the biased term and f is the Relu function. The filter is applied to the feature representation $z = z_1, z_2, \cdots, z_{l-k+1}$, resulting in the feature vector:

$$C = [c_1, c_2, \cdots, c_{l-k+1}]. \tag{4}$$

Where, k is the step dimension length and l is the text sequence length. In a two-channel architecture, the filter acts on each channel. And after calculating c_i, it is entered into the pooling layer. The role of the pooling layer is to reduce the model parameters and reduce the risk of over-fitting the model while preserving the main features.

In addition, j types of convolution kernel are used in this paper, and the number of each convolution kernel is m. The output result of each convolution kernel is $H_i (1 \leq i \leq j)$, and its calculation formula is shown in equation (5):

$$H_i = concat(max(c_1), max(c_2), \cdots, max(c_m)). \tag{5}$$

max is a one-dimensional pooling operation and $concat$ is a concatenation operation.

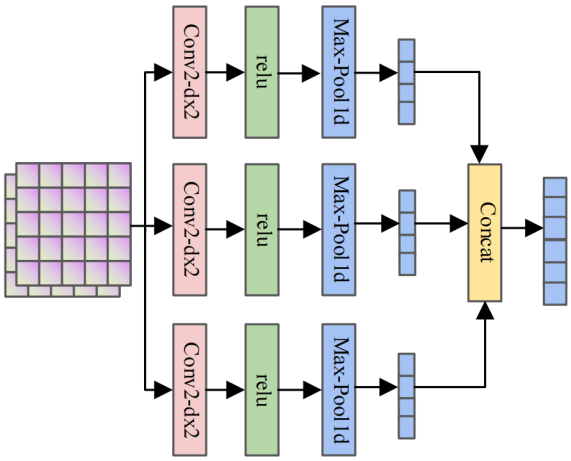


Fig. 3. Convolution layer structure

3.4. Two-way LSTM Layer

For long texts, the context information of words is sufficient and there are often long distance semantic associations. Compared with short texts with few features and insufficient time sequence information, long texts have higher requirements for the time sequence of features. Before the feature is input to LSTM, the multi-path convolution is fused first, which cannot guarantee the timing of the features after fusion, and greatly affects the

LSTM learning process of long text timing features. Let M_1 and M_2 respectively represent the feature graph matrix formed by the convolution of different paths. If M_1 and M_2 are joined horizontally, due to the difference in the convolution kernel size, M_1 and M_2 are different in row dimension, and can only be filled with 0, so that the convolution feature map size remains unchanged, which will cause the timing features of M_1 and M_2 cannot be completely aligned, resulting in the overall quality of the timing feature decline. If M_1 and M_2 are spliced vertically, the global order of the overall features cannot be guaranteed after splicing.

In order to avoid the shortcomings of the above fusion methods, the model in this paper uses bidirectional LSTM to learn the timing features of each route, and combines the bidirectional timing features of each route to represent the final text, avoiding the problem of declining the quality of timing features caused by the fusion of various features before entering the LSTM. Since the traditional forward LSTM can only learn the above information of features, ignoring the below information of features, this paper uses two-way LSTM to learn the context information of features at the same time, which greatly improves the learning ability of temporal features of the model. In order to make full use of the output features of all LSTM moments, the model uses the attention mechanism to carry out weighted summation of the features of each LSTM moment to improve the output quality of LSTM. The bidirectional LSTM layer in this paper is shown in Figure 4.

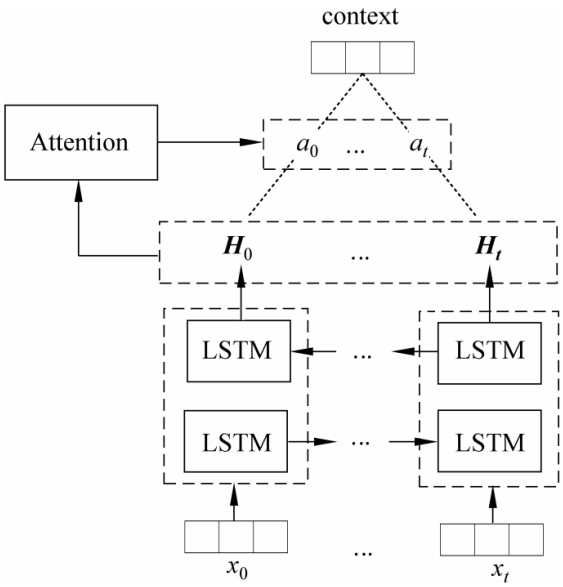


Fig. 4. Bidirectional LSTM layer

Let i represent the $i-th$ time and $i \in [0,t]$. x_i represents the input vector at time i. M_k represents the feature graph matrix formed by one convolution, then M_k can be expressed as the concatenation of multiple row vectors, as shown in equation (6).

$$M_k = x_0 \oplus x_1 \oplus \cdots \oplus x_t. \tag{6}$$

LSTM receives x_i as an input vector in chronological order. c_t indicates the LSTM unit status. h_t indicates the final output of the LSTM unit. f_t, i_t, o_t indicate the memory gate, input gate, and output gate respectively. σ indicates the Sigmoid activation function. W_f, W_i, W_o, W_c, b_f, b_i, b_o, b_c indicate the parameters that the network needs to learn. The final output calculation of LSTM is shown in equations (7-12). Since the model uses the bidirectional LSTM learning timing feature, the final output of bidirectional LSTM is obtained by combining the forward LSTM output with the reverse LSTM output.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f). \tag{7}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i). \tag{8}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o). \tag{9}$$

$$q_t = \sigma(W_c \cdot [h_{t-1}, x_t] + b_c). \tag{10}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot q_t. \tag{11}$$

$$h_t = o_t \tanh(c_t). \tag{12}$$

Since LSTM can only learn the output vector of the last moment and cannot make full use of the output of each moment, this paper uses the attention mechanism to complete the weighted fusion of the output features of each moment. Let H_i represent the output vector of the bidirectional LSTM layer at time i. e_i indicates how important H_i is to the overall semantic representation of the text. a_i represents the weight of H_i's contribution to the overall semantic representation of the text. According to the above definition, the attention weight of the bidirectional LSTM layer is calculated as shown in equations (13) and (14).

$$e_i = u^T \cdot \tanh j(W_a \cdot H_i + b_a). \tag{13}$$

$$a_i = \frac{exp(e_i)}{\sum_{j=0}^{t} exp(e_i)}. \tag{14}$$

Among them, u^T, W_a and b_a are the parameters that the network needs to learn. $\tanh$ is a nonlinear activation function. After the attention weights of each moment of the bidirectional LSTM layer are obtained, the output vectors of all moments of the bidirectional LSTM layer are weighted and summed by equation (15). The resulting vector v is the characteristic vector of the final output of the entire bidirectional LSTM layer.

$$v = \sum_{i=0}^{t} a_i H_i. \tag{15}$$

Let v_i represent the document representation vector learned after the $i-th$ way convolutional features pass through the bidirectional LSTM layer, then the document representation vector v_d finally formed by the model can be represented as the concatenation of n-way convolutional document representation vectors, as shown in equation (16).

$$v_d = v_1 \oplus v_2 \oplus \cdots v_n. \tag{16}$$

After the final representation vector v_d of the text is obtained, the final category output of v_d is carried out through the full connection layer and the Softmax layer. Let c represent a class. n indicates the number of categories. d represents the output vector of the document vector v_d after passing through the fully connected layer. d_c represents the value of the component of the vector d belonging to class c. p_c represents the probability that the text is classified as c. W_c and b_c are parameters that need to be learned in the fully connected layer network. f is a nonlinear activation function, then p_c is calculated as shown in equations (17) and (18).

$$d = f(W_c \cdot v_d + b_c). \tag{17}$$

$$p_c = \frac{exp(d_c)}{\sum_{k=1}^{n} exp(d_k)}. \tag{18}$$

3.5. Network Parameter Optimization Based on Improved ABC

Artificial bee colony algorithm (ABC) is an optimization method to imitate the honey collecting behavior of bees [44]. The minimum search model for generating swarm intelligence consists of three basic components: honey source, hired bees (also known as lead bees), and unhired bees (including scout bees and follower bees). Two basic behavioral models include recruiting bees for a honey source and giving up a honey source. The honey source is used to represent the solution of the problem, and the value of the honey source is expressed by the income degree function, which is the optimization object function. The guide bees correspond to the honey source one by one, which is used to store the relevant information of the corresponding honey source and share it with other bees with a certain probability. The main task of scout bees and follower bees is to find and extract honey. The process of bees gathering honey is the process of finding the optimal solution of the objective function. The basic swarm algorithm flow is shown in Figure 5.

However, like all other evolutionary algorithms, ABC also has the inherent problem of too slow or too fast convergence. Therefore, this paper adopts anti-learning algorithm, and proposes an improved artificial bee colony algorithm to improve the strategies of leading and following bees to search for honey sources by introducing scale factors, so as to realize the adaptive adjustment of the artificial bee colony algorithm. The specific policy modification expression is as follows.

$$v_{ij} = x_{ij} + \phi(x_{ij} - x_{kj}). \tag{19}$$

Where, $i = 1, 2, \cdots, N$, $j = 1, 2, \cdots, D$. Where v is used to indicate the location of pre-search for new nectar sources. x is used to indicate the specific location of the current nectar source. $\phi \in (-1, 1)$ is a scale regulating factor, which determines the scale

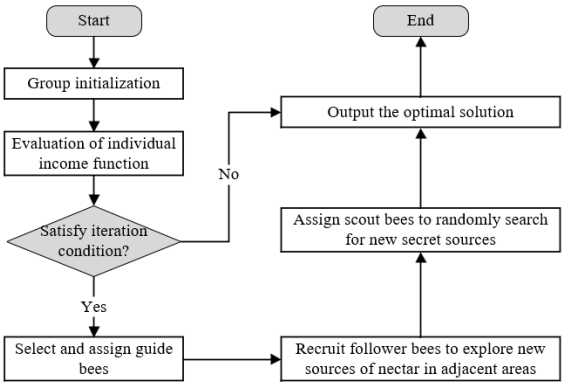


Fig. 5. ABC flow chart

of movement from the current solution to the next feasible solution. N is the number of nectar sources. D is the dimension of the solution. The expression for ϕ is as follows.

$$\phi(G+1) = \begin{cases} \phi_l + rand_1\sqrt{rand_2^2 + rand_3^2} & if P_i < rand_4\phi_u \quad otherwise \end{cases} \tag{20}$$

Where, $rand_i \in [0,1]$, $i = 1,2,3,4$. P_i is the selection probability value of the $i-th$ leader bee, that is, $P_i = \frac{Fitval_i}{\sum_{k=1}^{N} Fitval_k}$. According to the above adjustment principle, the calculation expression of the $G+1-th$ generation scaling factor ϕ_{G+1} is as follows.

$$\phi(G+1) = \{2\phi_l - \phi(G+1) \quad if \phi(G+1) < \phi_l 2\phi_u - \phi(G+1) \quad if \phi(G+1) > \phi_u \tag{21}$$

Among them, $Fitval_i$ is the return degree evaluation function value of the $i-th$ honey source. $rand_i$ is a random number between [0,1]. ϕ_u and ϕ_l are used to set the boundary of the range value of $\phi(G+1)$ under manual intervention.

The modified artificial bee colony algorithm is more close to the natural process of bees to find honey sources, and can ensure the diversity and balance of solutions in the algorithm.

In this paper, the linear weighting function of the maximum Shannon entropy and the minimum cross entropy is used as the objective evaluation function of the network classification result (i.e. the return degree function of the bee colony algorithm). By adjusting the scale factor ϕ of the solution adjustment strategy of the leading bees and the following bees, an improved artificial bee colony algorithm is proposed to optimize the parameters of the classification network. The step of the algorithm is as follows.

1. Group initialization. At the initial time, N feasible solutions are randomly generated $X = (\beta, V_\theta, a_\theta)$. The resulting formula is as follows.

$$X_i^j = X_{min}^j + rand(0,1)(X_{max}^j - X_{min}^j). \tag{22}$$

Where, $i = 1,2,3$. $\beta \in [0.001, 200]$, $V_\theta \in [0.001, 200]$, $Va_\theta \in [0.001, 100]$.

2. Automatically adjust the solution search strategy of leading and following bees.
3. Set the income degree function. In this paper, the linear weighted sum of maximum Shannon entropy and minimum cross entropy is used as the evaluation standard for parameter optimization adjustment of each generation.

$$fit = \rho H_1(p) - (1-\rho)H_2(p). \tag{23}$$

$$H_1(p) = p_1 \log_2 p_1 + p_0 \log_2 p_0. \tag{24}$$

$$H_2(p) = p_1 \log_2 \frac{p_0}{p_1} + p_0 \log_2 \frac{p_1}{p_0}. \tag{25}$$

Where $H_1(p)$ and $H_2(p)$ represent the maximum Shannon entropy and the minimum cross entropy in the fitness function in sequence. p_1 and p_0 represent the probability that the output value of the network is 1 and 0. ρ is the weighting coefficient of the fitness function, and $\rho \in [0,1]$. Where, if ρ takes 1 or 0 as the evaluation function, the two special cases are the cases where the maximum aroma entropy and the lowest cross entropy are adopted. Then the return degree evaluation function of the improved artificial bee colony algorithm is:

$$Fitval(\beta, V_\theta, a_\theta) = \begin{cases} \dfrac{1}{1+fit} & iffit \geq 01 + |fit| \quad iffit < 0 \end{cases} \tag{26}$$

4. Algorithm termination condition. To improve the termination conditions of the artificial bee colony algorithm, the local maximum iteration times (threshold $limit$ and global maximum iteration times) are set to ensure the termination of the algorithm.

3.6. Output Layer

Through the feature layer, the feature vector V_w based on the word level and the feature vector V_c based on the character level are obtained respectively, and the two feature vectors are spliced to obtain the fused feature vector V, which is the final representation of the input text. Then, input V into the fully connected neural network to obtain the output vector $O \in R^K$(K is the number of categories), as shown in formula (27):

$$O = sigmoid(V * W_f). \tag{27}$$

Where, W_f is the weight matrix of a fully connected network. $sigmoid$ is a nonlinear activation function. Finally, the values in O are mapped to conditional probabilities S through softmax, and the category with the highest probability of conditional probabilities is the prediction category. The calculation of S is shown in formula (28):

$$S = argmax(softmax(O)). \tag{28}$$

4. Experimental Results and Analysis

This section will first introduce the data set used in the experiment, the experimental environment and parameter settings, and the baseline method. Then, the experimental evaluation of the proposed method and baseline method on three data sets is given. Finally, the ablation experiment and comparison with the advanced method is given.

4.1. Dataset

The classification task is performed on three English text data sets: Reuters, the English Journal Articles Dataset (EJA), and the Agency French Press (AFP). Summary statistics for the data set are shown in Table 1.

Table 1. Dataset

Dataset	Training	Verification	Testing	Category number	Maximum length
Reuters	40000	5000	5000	14	32
EJA	10000	1000	1000	10	32
AFP	200000	20000	20000	15	36

EJA. This data set is an English journal paper dataset containing article titles. 10000 samples are randomly selected for training, 1000 samples for validation and 1000 samples for testing. The first 32 characters (including punctuation) are taken from each sample to form a new dataset.

Reuters. This dataset is a social news dataset with headlines and contains 40000 samples for training, 5000 samples for validation, and 5000 samples for testing. The first 32 characters (including punctuation) are taken from each sample to form a new dataset.

AFP. This dataset is divided into 15 categories, with 200000 samples reserved for training, 20000 for validation, and 20000 for testing. The first 36 characters (including punctuation characters) are taken from each sample as a new dataset.

4.2. Experimental Environment and Parameter Setting

This experiment is conducted in Python 3.7 and Pytorch 1.8.0, using an Intel(R) Core i5-10600KF CPU and NvidiaRTX 3050Ti [45].

The main parameter Settings of the experimental model are shown in Table 2. On EJA and Reuters data sets, the padding size is set to 32; On the AFP dataset, the padding size is set to 36. At the same time, in order to avoid model over-fitting and ineffective consumption of computing resources, the training process of the model is terminated in advance when the training effect of the model is not improved after 1000 batches. In addition, when using the word2vec model to train the word vector in this paper, the dimension of the word vector is 300, the size of the training window is 5, the threshold of word frequency is 5, and the number of training iterations is 5.

4.3. Baseline Method

In order to verify the validity of the proposed method in English text classification, this paper compares the proposed method with the following methods.

TextRNN. It respectively inputs words, characters and words mixed into word2vec to get text features, and apply TextRNN as a classifier. TextRNN uses the forward LSTM module and the backward LSTM module respectively to obtain the sum of the forward and backward hidden vectors, and finally concatenates the two into the softmax layer to obtain the classification result.

Table 2. Parameter configuration

Parameter	value
batch-size	64
learning rate	$5e^{-5}$
CNN filter size	2,3,4
CNN filter num	256
epoch	30
dropout	0.4
Multihead attention heads	5

RCNN. It inputs words, characters and words into word2vec respectively to get text features, and uses RCNN as a classifier. RCNN is a hybrid network model composed of two layers. In the first layer, RNN is applied to learn the text representation with word embeddings, and in the second layer, CNN and maximum pooling are applied to extract the most obvious features.

ERNIE. ERNIE designed a new continuous multi-paradigm unified pre-training framework and utilized specific self-attention codes to control the content of predicted conditions.

Attention+CNN. It takes ERNIE and word2vec output as text features, and applies the attention mechanism and CNN as classifiers.

When the accuracy is unbalanced or irregular, the model cannot be evaluated accurately [46]. The F1 value is used as the performance evaluation index, and the accuracy rate and recall rate are used as the reference. F1 is shown in formula (29).

$$F1 = \frac{2PR}{P+R}. \tag{29}$$

Precision (P) indicates the precision rate, and Recall (R) indicates the recall rate.

4.4. Results Analysis

To evaluate the effectiveness of proposed method, experiments are conducted on three datasets. Taking accuracy rate, recall rate and F1 as evaluation indicators, the experimental results are shown in Table 3, Table 4 and Table 5. It can be noted that the model proposed in this paper achieves higher performance than other baseline methods. In order to find the internal reasons, the experimental results are analyzed in detail as follows.

The experimental results are shown in Table 3, Table 4 and Table 5. By comparing the experimental results of TextRNN-w, TextRNN-c and TextRNN-wc on the three datasets, it can be found that TextRNN-wc has the best classification performance, followed by TextRNN-w. The results show that TextRNN-wc with two-level features has a higher F1 value. This not only proves that character-level features and word-level features can be used well together, but also shows that character-level features and word-level features promote each other. At the same time, similar conclusions can be drawn from the comparison results of RCNN-w, RCNN-c and RCNN-wc.

Then, comparing the proposed method with the experimental results of ERNIE, the performance of the proposed method is obviously better. The results show that combining

Table 3. Results on EJA/%

Model	P	R	F1
TextRNN-w	79.88	78.84	79.36
TextRNN-c	79.15	78.31	78.73
TextRNN-wc	80.19	79.67	79.93
RCNN-w	78.26	77.54	77.90
RCNN-c	79.22	78.62	78.92
RCNN-wc	80.87	80.41	80.64
ERNIE	83.66	83.10	83.38
Attention+CNN	85.65	85.43	85.54
Proposed	86.48	86.24	86.36

Table 4. Results on Reuters/%

Model	P	R	F1
TextRNN-w	92.07	91.39	91.73
TextRNN-c	91.86	91.46	91.66
TextRNN-wc	92.55	91.81	92.18
RCNN-w	93.72	93.38	93.55
RCNN-c	91.66	91.30	91.48
RCNN-wc	94.88	94.40	94.64
ERNIE	93.67	93.61	93.64
Attention+CNN	94.85	94.97	94.91
Proposed	95.81	95.71	95.76

the dynamic feature representation model ERNIE with the traditional static feature representation model word2vec can improve the performance of text classification. In addition, the experimental results of TextRNN-wc, RCNN-wc and proposed method show that the proposed method uses dynamic word vectors to represent character-level features, which can represent text information more accurately and contribute to the representation of text features.

Finally, on the three data sets, the proposed method has better classification performance than Attention+CNN. Although Attention+CNN introduces CNN and attention mechanism, it does not consider that there may be feature information loss in the process

Table 5. Results on AFP/%

Model	P	R	F1
TextRNN-w	91.96	92.56	92.26
TextRNN-c	91.83	91.53	91.78
TextRNN-wc	92.87	92.51	92.69
RCNN-w	93.12	92.42	92.77
RCNN-c	92.63	92.37	92.50
RCNN-wc	93.74	93.56	93.65
ERNIE	94.29	94.15	94.22
Attention+CNN	95.03	94.89	94.96
Proposed	96.14	96.10	96.12

of feature extraction. By stacking text embedding matrix before extracting local key features from CNN, the proposed method retains more text information, which can make the model obtain higher performance.

4.5. Ablation Experiment

Although the above experiments prove that the proposed method has good text classification performance, in order to display the functions of each part of the model. In this section, we will conduct ablation experiments on the proposed method in order to better demonstrate the contributions of each part of the model. The results of ablation experiments are shown in Table 6, Table 7, and Table 8.

Table 6. Ablation results on EJA/%

Model	P	R	F1
LSTM	83.76	83.38	83.57
LSTM+ABC	85.44	85.20	85.32
LSTM+attention	84.50	84.66	84.58
Proposed	86.48	86.24	86.36

Table 7. Ablation results on Reuters/%

Model	P	R	F1
LSTM	94.03	93.77	93.90
LSTM+ABC	94.86	94.74	94.80
LSTM+attention	95.34	95.20	95.27
Proposed	95.81	95.71	95.76

Table 8. Ablation results on AFP/%

Model	P	R	F1
LSTM	93.26	92.98	93.12
LSTM+ABC	95.22	95.12	95.17
LSTM+attention	94.83	95.01	94.92
Proposed	96.14	96.10	96.12

The results of the ablation experiment are shown in Table 6, Table 7 and Table 8. Comparing LSTM with LSTM+attention, it can be seen that LSTM+attention has better classification performance than LSTM. This experimental result proves that the attention layer can highlight the features that are highly relevant to text semantics and enhance their weight. From the experiments of LSTM and LSTM+ABC, it can be found that

LSTM+ABC has higher classification performance. This shows that the network parameter optimization based on ABC is helpful for text classification. Finally, the four cases are compared, and the proposed method achieves higher performance on the three data sets. The results show that the convolution layer can make up for the lack of local feature extraction ability in the attention layer, and the model is more efficient when the attention layer constructs richer semantic information.

Finally, we choose three advanced methods to compare with the method in this paper including GMT [47], SSF [48], SCLTW [49]. From the table 9, the proposed method in this paper obtains the optimal F1 value.

Table 9. Comparison results with different method (F1)/%

Model	GMT	SSF	SCLTW	Proposed
EJA	79.88	82.63	85.25	86.36
Reuters	91.08	92.71	94.65	95.76
AFP	91.92	93.25	95.33	96.12

5. Conclusion

Aiming at the difficulty of text representation and feature extraction in English text classification tasks, this paper proposes a method of English short text classification with mixed features and multiple attention. Firstly, combining word-level features and character-level features, multilevel feature representation of text data is carried out. Then the attention layer and convolution layer are used to extract the global and local features of the text respectively. Finally, the two features are fused and text categories are predicted by the full connection layer and softmax layer. At the same time, the improved ABC is used to optimize the network parameters, and better classification effect is obtained. Experiments are carried out on three public data sets and compared with the current mainstream classification models. The experimental results show that the proposed method has good classification performance compared with the baseline model. Although the method proposed in this paper has certain performance, it also has some obvious shortcomings. The number of parameters in the model is large and the calculation cost is high. The next step will be to verify the effectiveness of the algorithm on more data sets. At the same time, considering the deficiency of the model, the model is studied in a finer granularity.

References

1. Jha V, Savitha R, Shenoy P D, et al. A novel sentiment aware dictionary for multi-domain sentiment classification[J]. Computers & Electrical Engineering, 2018, 69: 585-597.
2. Yuan Z, Wu S, Wu F, et al. Domain attention model for multi-domain sentiment classification[J]. Knowledge-Based Systems, 2018, 155: 1-10.
3. Minaee S, Kalchbrenner N, Cambria E, et al. Deep learning–based text classification: a comprehensive review[J]. ACM computing surveys (CSUR), 2021, 54(3): 1-40.

4. X. Meng, X. Wang, S. Yin, et al. Few-shot image classification algorithm based on attention mechanism and weight fusion[J]. Journal of Engineering and Applied Science. 70, 14 (2023). https://doi.org/10.1186/s44147-023-00186-9.
5. Kowsari K, Brown D E, Heidarysafa M, et al. Hdltex: Hierarchical deep learning for text classification[C]//2017 16th IEEE international conference on machine learning and applications (ICMLA). IEEE, 2017: 364-371.
6. Sun N, Du C. News text classification method and simulation based on the hybrid deep learning model[J]. Complexity, 2021, 2021: 1-11.
7. Lyu S, Liu J. Convolutional recurrent neural networks for text classification[J]. Journal of Database Management (JDM), 2021, 32(4): 65-82.
8. Du C, Huang L. Text classification research with attention-based recurrent neural networks[J]. International Journal of Computers Communications & Control, 2018, 13(1): 50-61.
9. Sari W K, Rini D P, Malik R F. Text Classification Using Long Short-Term Memory with GloVe[J]. Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI), 2019, 5(2): 85-100.
10. Liu J, Yang Y, Lv S, et al. Attention-based BiGRU-CNN for Chinese question classification[J]. Journal of Ambient Intelligence and Humanized Computing, 2019: 1-12.
11. Tao H, Tong S, Zhao H, et al. A radical-aware attention-based model for chinese text classification[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2019, 33(01): 5125-5132.
12. Hao M, Xu B, Liang J Y, et al. Chinese short text classification with mutual-attention convolutional neural networks[J]. ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP), 2020, 19(5): 1-13.
13. Wang P, Fan E, Wang P. Comparative analysis of image classification algorithms based on traditional machine learning and deep learning[J]. Pattern Recognition Letters, 2021, 141: 61-67.
14. Jin N, Wu J, Ma X, et al. Multi-task learning model based on multi-scale CNN and LSTM for sentiment classification[J]. IEEE Access, 2020, 8: 77060-77072.
15. Senthil Kumar N K, Malarvizhi N. Bi-directional LSTM-CNN combined method for sentiment analysis in part of speech tagging (PoS)[J]. International Journal of Speech Technology, 2020, 23: 373-380.
16. Lin J, Ma J, Zhu J, et al. Short-term load forecasting based on LSTM networks considering attention mechanism[J]. International Journal of Electrical Power & Energy Systems, 2022, 137: 107818.
17. Hu Z, Chen L, Luo Y, et al. Eeg-based emotion recognition using convolutional recurrent neural network with multi-head self-attention[J]. Applied Sciences, 2022, 12(21): 11255.
18. Zhou H. Research of text classification based on TF-IDF and CNN-LSTM[C]//Journal of Physics: Conference Series. IOP Publishing, 2022, 2171(1): 012021.
19. Chen J, Kakillioglu B, Velipasalar S. Background-aware 3-D point cloud segmentation with dynamic point feature aggregation[J]. IEEE Transactions on Geoscience and Remote Sensing, 2022, 60: 1-12.
20. Jia X, Wang L. Attention enhanced capsule network for text classification by encoding syntactic dependency trees with graph convolutional neural network[J]. PeerJ Computer Science, 2022, 8: e831.
21. Bang J, Park J, Park J. GACaps-HTC: graph attention capsule network for hierarchical text classification[J]. Applied Intelligence, 2023: 1-18.
22. Huang W, Lin M, Wang Y. Sentiment analysis of Chinese e-commerce product reviews using ERNIE word embedding and attention mechanism[J]. Applied Sciences, 2022, 12(14): 7182.
23. Moradzadeh A, Teimourzadeh H, Mohammadi-Ivatloo B, et al. Hybrid CNN-LSTM approaches for identification of type and locations of transmission line faults[J]. International Journal of Electrical Power & Energy Systems, 2022, 135: 107563.

24. Singh R, Saurav S, Kumar T, et al. Facial expression recognition in videos using hybrid CNN & ConvLSTM[J]. International Journal of Information Technology, 2023, 15(4): 1819-1830.
25. Chung J, Jang B. Accurate prediction of electricity consumption using a hybrid CNN-LSTM model based on multivariable data[J]. PloS one, 2022, 17(11): e0278071.
26. ETNER H. Multi-Label Text Analysis With A CNN And LSTM Based Hybrid Deep Learning Model[J]. Adyaman Üniversitesi Mühendislik Bilimleri Dergisi, 2022, 9(17): 447-457.
27. Zhou S, Guo S, Du B, et al. A Hybrid Framework for Multivariate Time Series Forecasting of Daily Urban Water Demand Using Attention-Based Convolutional Neural Network and Long Short-Term Memory Network[J]. Sustainability, 2022, 14(17): 11086.
28. Hasib K M, Azam S, Karim A, et al. Mcnn-lstm: Combining cnn and lstm to classify multi-class text in imbalanced news data[J]. IEEE Access, 2023.
29. Long S, Han S C, Wan X, et al. Gradual: Graph-based dual-modal representation for image-text matching[C]//Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. 2022: 3459-3468.
30. Li Y A, Han C, Jiang X, et al. Phoneme-Level Bert for Enhanced Prosody of Text-To-Speech with Grapheme Predictions[C]//ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2023: 1-5.
31. Kim Y, Kim J H, Lee J M, et al. A pre-trained BERT for Korean medical natural language processing[J]. Scientific Reports, 2022, 12(1): 13847.
32. Soni S, Chouhan S S, Rathore S S. TextConvoNet: A convolutional neural network based architecture for text classification[J]. Applied Intelligence, 2023, 53(11): 14249-14268.
33. Xu X, Li D, Zhou Y, et al. Multi-type features separating fusion learning for Speech Emotion Recognition[J]. Applied Soft Computing, 2022, 130: 109648.
34. Ferjani I, Hidri M S, Frihida A. SiNoptiC: Swarm intelligence optimisation of convolutional neural network architectures for text classification[J]. International Journal of Computer Applications in Technology, 2022, 68(1): 82-100.
35. Ding Y, Jia M, Miao Q, et al. A novel time-frequency Transformer based on self-attention mechanism and its application in fault diagnosis of rolling bearings[J]. Mechanical Systems and Signal Processing, 2022, 168: 108616.
36. Meng S, Zhou Y, Gao Z. Refined self-attention mechanism based real-time structural response prediction method under seismic action[J]. Engineering Applications of Artificial Intelligence, 2024, 129: 107380.
37. Zhang Y, Zheng J, Jiang Y, et al. A text sentiment classification modeling method based on coordinated CNN-LSTM-attention model[J]. Chinese Journal of Electronics, 2019, 28(1): 120-126.
38. Gong J, Zhang J, Guo W, et al. Short Text Classification Based on Explicit and Implicit Multi-scale Weighted Semantic Information[J]. Symmetry, 2023, 15(11): 2008.
39. Lin X, Xiong G, Gou G, et al. Et-bert: A contextualized datagram representation with pre-training transformers for encrypted traffic classification[C]//Proceedings of the ACM Web Conference 2022. 2022: 633-642.
40. Li J, Katsis Y, Baldwin T, et al. SPOT: Knowledge-Enhanced Language Representations for Information Extraction[C]//Proceedings of the 31st ACM International Conference on Information & Knowledge Management. 2022: 1124-1134.
41. Peng B, Zhang T, Han K, et al. BVMHA: Text classification model with variable multihead hybrid attention based on BERT[J]. Journal of Intelligent & Fuzzy Systems (Preprint): 1-12.
42. F. Xiao, S. Yu and Y. Li, "Efficient Large-Capacity Caching in Cloud Storage Using Skip-Gram-Based File Correlation Analysis," in IEEE Access, vol. 11, pp. 111265-111273, 2023, doi: 10.1109/ACCESS.2023.3322725.
43. L. Teng et al., "FLPK-BiSeNet: Federated Learning Based on Priori Knowledge and Bilateral Segmentation Network for Image Edge Extraction," in IEEE Transactions on Network and Service Management, vol. 20, no. 2, pp. 1529-1542, June 2023, doi: 10.1109/TNSM.2023.3273991.

44. S. Yin, J. Liu, L. Teng. An Improved Artificial Bee Colony Algorithm for Staged Search[J]. TELKOMNIKA Telecommunication, Computing, Electronics and Control. 14(3):1099-1104, 2016.
45. Y. Jiang, S. Yin. Heterogenous-view Occluded Expression Data Recognition Based on Cycle-Consistent Adversarial Network and K-SVD Dictionary Learning Under Intelligent Cooperative Robot Environment. Computer Science and Information Systems, vol. 20, no. 4, 2023. https://doi.org/10.2298/CSIS221228034J.
46. Yin S, Li H, Sun Y, et al. Data Visualization Analysis Based on Explainable Artificial Intelligence: A Survey[J]. IJLAI Transactions on Science and Engineering, 2024, 2(2): 13-20.
47. Abdulla H H H A, Awad W S. Text Classification of English News Articles using Graph Mining Techniques[C]//ICAART (3). 2022: 926-937.
48. Fkih F, Alsuhaibani M, Rhouma D, et al. Novel Machine Learning-Based Approach for Arabic Text Classification Using Stylistic and Semantic Features[J]. Computers, Materials & Continua, 2023, 75(3).
49. Guo J, Zhao B, Liu H, et al. Supervised contrastive learning with term weighting for improving Chinese text classification[J]. Tsinghua Science and Technology, 2022, 28(1): 59-68.

Tianying Wen is with the Department of Education, Liaoning National Normal College. She had published several papers related to her major. Research direction: education analysis, data analysis, artificial intelligence.

Received: March 14, 2024; Accepted: June 21, 2024.

Computer Science and Information Systems 21(4):1629–1649 https://doi.org/10.2298/CSIS240314051Z

A Novel Industrial Big Data Fusion Method Based on Q-learning and Cascade Classifier

Xi Zhang[1], Jiyue Wang[2,⋆], Ying Huang[3], and Feiyue Zhu[2]

[1] School of Mechanical Engineering, Zhengzhou University of Science and Technology
Zhengzhou 450064 China
zhangxi_9534@163.com

[2] School of Vehicle and Traffic Engineering, Zhengzhou University of Science and Technology
Zhengzhou 450064 China
wangjiyue004@126.com

[3] School of Electronics and Electrical Engineering, Zhengzhou University of Science and Technology
Zhengzhou 450064 China
huangying_hy31@163.com

Abstract. The traditional industrial big data fusion algorithm has low efficiency and difficulty in processing high-dimensional data, this paper proposes a Q-learning-based cascade classifier model for industrial big data fusion. By combining cascade classifier and softmax classifier, feature extraction and data attribute classification of source industrial big data are completed in this cluster. In order to improve the classification rate, an improved Q-learning algorithm is proposed, which makes the improved algorithm randomly select actions in the early stage, and dynamically change in the late stage in the random selection of actions and actions with the highest reward value. It effectively improves the defects of traditional Q-learning algorithm that it is easy to fall into the local optimal and has slow convergence speed. The experimental results show that compared with other advanced fusion algorithms, the proposed method can greatly reduce the network energy consumption and effectively improve the efficiency and accuracy of data fusion under the same data volume.

Keywords: industrial big data fusion, Q-learning, cascade classifier, feature extraction.

1. Introduction

With the rapid development of industrial Internet of Things (IoT) technology, wireless sensor networks, as the core component of IoT sensing layer, have been widely used in various environmental monitoring fields [1-3]. In practice, most sensor nodes are powered by batteries, which leads to limited network resources. Due to the uneven geographical distribution of a large number of nodes, there is too much redundant information in the data [4], which increases energy consumption and transmission delay. In addition, due to the widespread interference in the application environment of the Internet of Things, it will directly weaken the data communication transmission capability and reduce the

⋆ Corresponding Author

accuracy of data acquisition, affecting the overall performance of the Internet of Things system.

In order to solve the above problems, some fusion technologies for industrial big data came into being [5,6]. The main idea is to fuse data from multiple different source nodes to eliminate redundancy and reduce data transmission volume, so as to improve network performance, extend network life and reduce energy consumption. The traditional fusion algorithms are mostly based on BP neural network, SOFM and other shallow network models. These models are prone to problems such as over-fitting, model training trapping local minimum, slow convergence rate, etc., resulting in reduced algorithm efficiency, weak feature extraction and classification ability, and unable to effectively process high-dimensional data [7,8]. In view of the above defects, the application of deep model has become a new direction for the integration of industrial big data. In 2015, LeCun et al. [9] proposed the concept of "deep learning". Then, convolutional neural network (CNN) [10], stack auto-encoder (SAE) and other models have been widely used. Currently, deep learning has been studied in the field of data fusion. References [11-13] proposed to use the SAE model in deep learning to carry out feature fusion on industrial big data, significantly improving the accuracy of fusion. In reference [14], combining the deep model and sparse filtering algorithm, the BSSFM model was designed to extract data features, which improved the efficiency of fusion and reduced energy consumption. However, the use of SAE and other models would generate a large number of weight parameters in the training process, increasing the difficulty of model training.

Reference [15] studied a multi-sensor data fusion method in a complex environment. According to the results of trust, the idea of evidence iterative fusion was introduced to correct evidence conflicts from the level of evidence sources. Then it fused the modified evidence data, that is, it realized the fusion of multi-sensor data. Reference [16] proposed a multi-source sensor data fusion method based on manifold learning. T-SNE algorithm was adopted to reduce the dimension of high-dimensional multi-source sensor data, and the low-dimensional probability matrix was updated to generate a reasonable exclusion gradient between the points with small distance from the multi-source sensor data and completed the data fusion. Reference [17] proposed a multi-sensor data fusion method based on data correlation. MDF technology was used to identify hidden association features in group data, and the fusion of multi-sensor data was completed through the fusion of association features. Reference [18] proposed a multi-sensor data fusion method for UAV detection and tracking to measure the reliability of the data set. According to the reliability results, the Gaussian process model was constructed to complete the multi-sensor data fusion. However, the above methods have low efficiency in the process of fusion.

The multi-modal data fusion methods can be roughly divided into two categories: model-independent fusion methods and model-based fusion methods. Among them, the model independent method is simpler but less practical, and the fusion process is easy to produce losses. Model-based fusion method is more complex, but it has high accuracy and practicability, and is also the mainstream method used at present [19].

In the process of multimodal fusion, the time of fusion is an important consideration. According to different fusion periods or fusion levels, there are three model-independent fusion methods, each of which has its own characteristics [20]. In different experiments, different fusion methods can be tried to get better results. Some of the characteristics of the modes, such as the different data acquisition rates, present new challenges on how to

synchronize the entire fusion process. The following is a detailed overview of the three fusion methods. Table 1 compares the three data fusion methods.

Table 1. Performance comparison of three model-independent fusion methods

Method	Information loss	Fusion difficulty	Fault tolerance	Fusion level	Fusion stage
Early fusion	Middle	Hard	Poor	Low	After feature extraction
Post fusion	Big	Middle	Middle	Middle	Post-decision
Hybrid fusion	Small	Easy	Good	High	Simultaneously

Early fusion, also known as feature fusion, is a fusion method that is carried out immediately after the feature extraction of the modes. The advantage of feature fusion is that the correlation between multiple features from different modes can be exploited early on, and is suitable for situations where the modes are highly correlated. For example, early fusion is used when combining audio and video features of speech recognition. However, it is difficult to extract features, so it is not the most ideal fusion method.

This method is difficult to represent the time synchronization cabinet between multimodal features. Since the representation, distribution, and density of various modes may vary, simple connections between properties may ignore the properties and correlations unique to each mode, and may create redundancy and data dependencies between the data. Features that need to be fused are also required to be represented in the same format before fusion. As the number of features increases, it is difficult to obtain cross-correlations between these features. Figure 1 shows an early fusion approach.

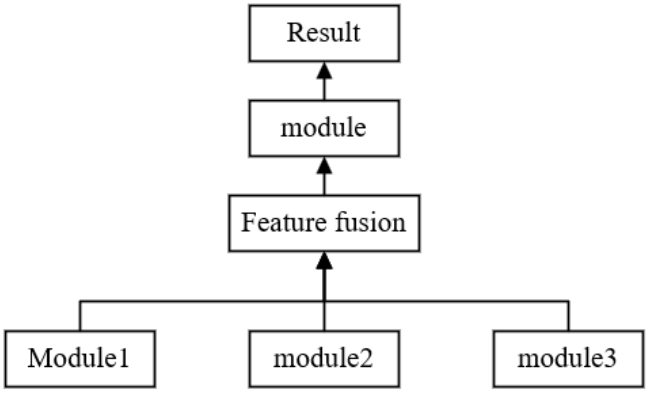


Fig. 1. Early fusion process

Late fusion, also known as decision-level fusion, refers to the fusion that occurs after a decision (classification or regression) has been made for each mode. For post-fusion, it is necessary to use corresponding models to train different modes, and then fuse the output results of these models. Compared with the earlier fusion cooperation, this fusion

approach can handle simple data asynchronism. Another advantage is that it allows the use of methods that are best suited to analyzing each single mode, such as Hidden Markov Model (HMM) for audio and Support Vector Machines (SVM) for images.

However, the late fusion ignores the low-level interaction between multiple modes, and the fusion is difficult. Since different classifiers require different decisions, the learning process becomes both time-consuming and laborious. Figure 2 shows the structure of the late fusion method.

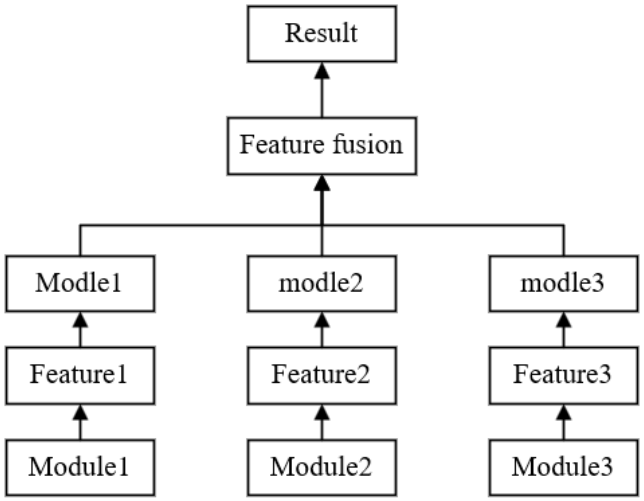


Fig. 2. Post fusion process

References [21,22] adopted fuzzy set, neural network, evidence theory and FIM algorithms for data fusion respectively, which did not consider the characteristics of sensor nodes and assumed that the collected data was correct. They did not judge the credibility of the source of the data, so the accuracy of the data was not guaranteed. Reference [23] proposed a fusion algorithm based on variable weights (VWFFA), which considered some influential factors in the communication process to weight the data. An improved fuzzy fusion algorithm was proposed in reference [24], which was weighted by considering the influence of external conditions on the calculation reliability. Both reduce redundancy and improve the accuracy of data, but they lack pertinence for different fusion stages, and the accuracy and real-time performance of data need to be further improved.

In order to improve the quality of industrial big data classification and fusion, this study proposes a new data fusion method from the perspective of deep learning.

2. Related Works

Model-based data fusion methods have a wider application range and better effect than model-independent methods, and current research is more inclined to such methods. Com-

mon methods include multi-core learning method, image model method and neural network method.

2.1. Multi-Kernel Learning (MKL)

The Multi-Kernel Learning (MKL) method is an extension of the kernel support vector machine (SVM) method and is the most commonly used method before deep learning, which allows different views of the data to be checked using different methods [25]. Because kernel can be regarded as similar function between data points, this method can better fuse heterogeneous data and use flexibly. Mirbeygi et al. [26] used MKL to sort the similarity of music artists from three aspects: acoustics, semantics and artists' social view, and proposed a new multi-core learning (MKL) algorithm, which could learn similar spatial items to generate similar spaces and combine all feature spaces into a unified embedded space in an optimal way. Figure 3 is the process of MKL.

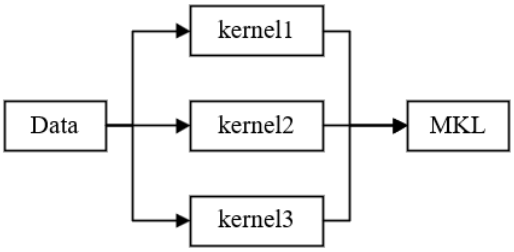


Fig. 3. Multi-kernel learning process

In some applications, there may be different information sources from different modes or results corresponding to different experimental methods, and each information source has one or more kernel questions of its own. The advantages of this method are flexible kernel selection, convex loss function (minimum value is the minimum value), and global optimal solution can be used to train the model and improve the model performance. Better MKL algorithms can be designed to improve accuracy, reduce complexity and training time.

Because in many applications, people propose many possible kernel functions, not choosing one of them but using them together, there is a lot of work in multi-kernel learning methods. The high time and space complexity is the main reason that the multi-core learning method can not be widely used. Another disadvantage is the large memory footprint and a slight dependence on training data.

2.2. Image Model Method

Image model method is also a common fusion method, which mainly fuses shallow or deep graphics through image segmentation, splicing and prediction operations, so as to obtain the final fusion result1.

Common image models are divided into generative (joint probability) models and discriminative (conditional probability) models. The use of image models in many studies, especially in statistical natural language processing, focuses on generative models that attempt to model the joint probability distribution of inputs and outputs." In the early days, generative models were mainly used, such as dynamic Bayesian Networks (DynamicBayesian Networks) and hidden Markov models. In later studies, discriminative models were more popular, simpler and easier to learn than generative models. Common discrimination models, such as Conditional Random Field (CRF), classify and label the components of the image2. Table 2 compares the generative model with the discriminant model [27,28].

The main advantage of image models is that they are easy to explore the spatial and temporal structure of the data, and the interpretability of the model is enhanced by embedding expert knowledge into the model. The disadvantage is that there are complex dependencies between features, and the generalization of the model is not strong.

Table 2. Comparison of generation model and discrimination model

Comparison item	Generative model	Discrimination model
Characteristic	Finding the optimal classification surface between different categories	A posteriori probability modeling
Distinction	Model the joint distribution	Model the conditional distribution
Sample size	more	Less than the generative model
Accuracy rate	Low	Higher than the generation model

2.3. Neural Network Method

Neural network method is one of the most widely used methods at present. Long Short-Term Memory (LSTM) and Recurrent Neural networks (RNNS) are often used to fuse multi-modal information. For example, bidirectional long short-term memory network is used for multi-modal emotion recognition; multimodal Recurrent Neural Networks (m-RNN) are used to directly take the image representation, word vector and implicit vector as the input of multimodal judgment, which shows a good effect in image subtitle processing and other tasks [29].

Some researchers have achieved better results through model patchwork than multi-core learning and image models. The application of neural network method in multi-mode fusion has strong learning ability and good expansibility. The drawback is that as the number of modes increases, deep learning becomes less interpretable and needs to rely on large amounts of training data. Table 3 compares the three model-based fusion approaches.

Table 3. Performance comparison of three model-independent fusion methods

Method	workload	Model interpretability	Feature dependence	Applications
MKL	big	weak	simple	EasyMKL
Image-based	small	strong	Complex	Multimodal data classification
NNM	big	weak	simple	Multi-modal emotion recognition

Neural network structure optimization based on Genetic algorithm (GA) is one of the earliest meta-heuristic search algorithms for neural network structure search and optimization. In the early 2000s, an algorithm called Neural Evolution of Augmented Topology (NEIT), which also uses GAs to evolve increasingly complex neural network structures, received a lot of attention. Shinozaki et al. P applied GAs and covariance matrix evolution strategies to optimize the structure of DNN and parameterized the structure of DNN to a simple binary vector based on directed acyclic graph representation. Since the GA search space can be very large and each model in the search space is expensive to evaluate, parallel searches using large GPU clusters speed up the process. These neural network structure search and optimization techniques can be easily extended to multimodal settings if a suitable representation of the network architecture is designed and it is not very expensive to train and test multiple architectures during the search process.

3. Q-learning Algorithm

The basic idea of reinforcement learning is to maximize the cumulative reward value obtained by the Agent from the environment in order to learn the optimal strategy to achieve the goal [30]. Reinforcement learning is a self-supervised learning method, which is different from supervised and unsupervised learning [31]. It does not require any data to be given in advance, but obtains learning information and updates model parameters by receiving environment's reward for action. The most important idea of reinforcement learning is continuous interaction with the environment, whose interaction elements include state, behavior and reward [32]. The process of reinforcement learning is shown in Figure 4. At time t, the Agent accepts the reward value R given by the environment and the state s at the next moment. At time $t+1$, the agent responds to the environment with an action a.

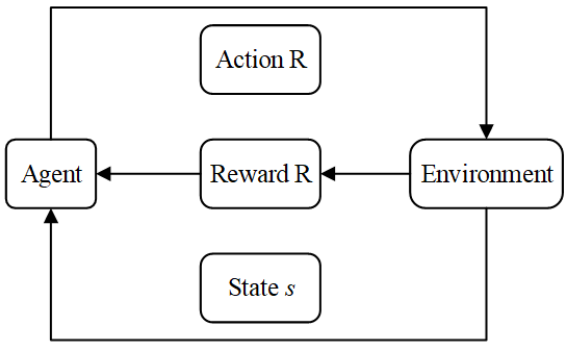


Fig. 4. Basic model of reinforcement learning

Reinforcement learning is divided into two types: model-based reinforcement learning and model-free reinforcement learning. In this paper, Q-learning algorithm in model-free reinforcement learning is selected to solve the problem and establish the algorithm model

[33]. In Q-learning algorithm, the environment will make the corresponding reward value R according to the action feedback of the Agent. The algorithm combines state s and action a together to construct Q-table and store the Q value of each update. It selects the action that can get the maximum reward according to the Q-table. After each action is selected, Q-table is updated. The Q-table update rules are as follows:

$$Q(s,a) \leftarrow Q(s,a) + \alpha[R + \gamma maxQ(s',a') - Q(s,a)]. \tag{1}$$

Where s represents the current state. a represents the action selected from the current state. s' is the next state. a' represents the action selected from the next state. α represents the learning rate of control convergence ($0 < \alpha < 1$). γ represents the discount factor ($0 \leq \gamma < 1$), $\gamma = 0$ indicates immediate reward, and $\gamma = 1$ indicates future reward).

Where α stands for learning rate. If the learning rate is higher, the previous learning results will be retained less, and the Agent will more prefer the present and future results. As a discount factor, if $\gamma \to 0$, the Agent tends to choose the immediate reward; if $\gamma \to 1$, it means that the Agent is more far-sighted and tends to consider the influence of the subsequent reward. Algorithm 1 shows the update steps of Q-learning algorithm.

Algorithm 1 Power method

Input: Parameters;
Output: State;
1: Initialize $Q(s,a)$, $s \in S, a \in A$;
2: for each episode
3: Initialize current-s
4: **while** not finished (all states) **do**
5: At a certain time t, an optional action a is selected according to the current state s through the greedy dream strategy;
6: Perform action a to obtain the state s' at time $t+1$ and the reward R.
7: Update Q-table;
8: Compute formula (1);
9: Record next state s' and convert it to the current state;
10: **end while**

Q-learning algorithm selects actions in two ways. One is exploration, which randomly explores an action and a state to transform; The other is utilization, for all the actions that may be selected to search for the next state, generate the maximum reward value and the action corresponding to the next state value.

4. Data Fusion Strategy

In this paper, the deep learning algorithm is used to design a hierarchical automatic dimension reduction classifier to process low-dimensional data from multi-source sensors and obtain ideal feature extraction results [34]. The specific implementation process is detailed as follows. Data dimension reduction is the process of mapping data from high-dimensional feature space to low-dimensional feature space, which can avoid the interference of chaotic and invalid data and improve the quality and efficiency of computation.

The designed automatic dimension reduction classifier (ADRS) is a single hidden layer structure. The ADRS structure is shown in Figure 5.

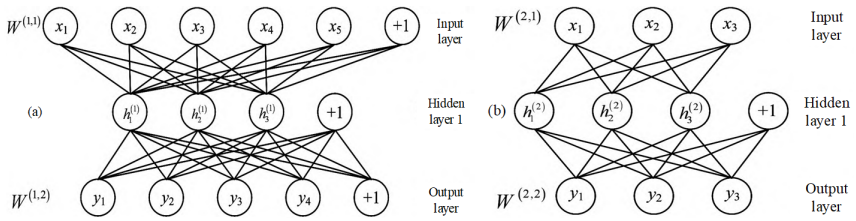


Fig. 5. ADRS structure. (a) k=1, (b) k=2

The input layer refers to the data input into the neural network. The input data must be numeric, meaning that non-numeric content needs to be converted to numeric value. Typically, data processing is the most time-consuming part of creating a machine learning model.

The hidden layer is composed of most of the neurons in the neural network and is the core part that processes the data to obtain the desired output. The data will pass through the hidden layer and undergo many weight and bias adjustments.

The output layer is the end product of processing data in a neural network and can represent different things. Typically, the output layer is made up of neurons, each representing an object.

The objective of ADRS dimension reduction classifier is to find the optimal fusion parameters in the sensor nodes (W, b). Fusion parameters determine the performance of data fusion. Selecting the optimal fusion parameters can improve the data fusion capability.

The global optimal solution of multi-source sensor network was found to optimize the initial weights and thresholds of the classification process. The essence of deep learning training is to update the weight. Before training a new model, it is necessary that each parameter has a corresponding initial weight. The selection of initial weight has certain influence on the placement of local minimum points and the improvement of network convergence speed. A neuron is a nonlinear unit with multiple inputs and single outputs. Only when the sum of inputs exceeds a certain value can the output respond, which is generally called the threshold value.

The input sensor node information x can achieve dimension reduction under the action of the output result y, so as to obtain the output result $a^{(k,2)}$ of the hidden layer. And $a^{(k,2)}$ is regarded as the characteristic result obtained after dimensionality reduction of input items for classification.

In the calculation process, to ensure the sparse robustness of features of $a^{(k,2)}$, three constraint functions should be used to achieve this, as shown in Equation (2):

$$minL(W, b_{argW,b}) = \frac{1}{m}\sum_{i=1}^{m}[x^i, y^i] + \frac{\lambda}{2}\sum_{l=1}^{n_l-1}\sum_{i=1}^{s_l}\sum_{j=1}^{s_l+1}[W_{ji}^{(k,l)}]^2 + \beta\sum_{j=1}^{S_2}KL(\rho/\bar{\rho}_j). \quad (2)$$

Where $L(W, b)$ represents the node energy loss function. ρ, λ and β represent the weight coefficient value of the classifier. KL represents constant constraint term in sparsity condition, $KL = 0.5||y^{(i)} - x^{(i)}||^2$. $x^{(i)}$ and $y^{(i)}$ represent constraint constant of input layer and output layer respectively. m indicates the number of samples in the ADRS. n_l indicates the number of network layers of the sensor. s_l represents the number of units contained in layer l. $W_{ji}^{(k,l)}$ represents the weight of connection between layer l and unit j in layer $l+1$ in the $k-th$ ADRS network. $W^{(k,l)}$ represents cell bias at each level in the sensor network.

In order to minimize the value of loss function, parameters in sensor network are trained by gradient descent algorithm to achieve dimension reduction. The specific implementation process is detailed as follows:

- Step 1. For all layers $l-l$, the element bias variable parameter satisfies $\Delta W^{(k,l)} = 0$, $\Delta b^{(k,l)} = 0$.
- Step 2. Set $i \in [1, m]$. The calculation formula is as follows:

$$\Delta W^{(k,l)} = \Delta W^{(k,l)} + \nabla_{W^{(k,l)}} \delta^{(k,l+1)} (\alpha^{(k,l)})^T. \tag{3}$$

$$\Delta b^{(k,l)} = \Delta b^{(k,l)} + \nabla_{b^{(k,l)}} \delta^{(k,l+1)}. \tag{4}$$

  Where $\alpha^{(k,l)}$ represents the loss parameter of unit n in layer l. $\delta^{(k,l+1)}$ represents the residual of element n in layer $l+1$. T stands for loss constraint.
- Step 3. Update the parameter values in ADRS.

$$W^{(k,l)} = W^{(k,l)} - a[\frac{1}{m} \Delta W^{(k,l)}] + \lambda W^{(k,l)}. \tag{5}$$

$$b^{(k,l)} = b^{(k,l)} - a[\frac{1}{m} \Delta b^{(k,l)}]. \tag{6}$$

- Step 4. Return to step 2, loop calculation, until meet preset conditions, then output ($W^{(k,1)}$, $b^{(k,1)}$, $W^{(k,2)}$, $b^{(k,2)}$).

After the above dimension reduction, the classifier is designed. Cascade automatic classifier ADRS is obtained by several dimension reduction processors through the cascaded action. ADRS has an obvious advantage that the output result of the input item can only be obtained after several dimensional reduction processing, eliminating most redundant information. The parameter values of each layer in ADRS can be calculated by the greedy floor algorithm, even if the output results of the upper layer are used as the input items of the next layer. The specific implementation process is as follows.

Set the number of hidden layers in ADRS to N_k. After the input data x is processed by the greedy algorithm, the parameters and output result $a^{(1,2)}$ are obtained, as shown in Figure 5(a). Then $a^{(1,2)}$ is taken as the input item, and the output result $a^{(2,2)}$ is obtained after the next round of greedy algorithm [35]. The process is shown in Figure 5(b).

Repeat training process. For the ADRS with N_k layers, after a greedy training, parameter set $(W^{(k,1)}, b^{(k,1)})|k = 1, \cdots, N_k$ is obtained. $(W^{(k,1)}, b^{(k,1)})$ is regarded as the connection weight value between ADRS levels. The multi-source sensor coding classifier is shown in Figure 6.

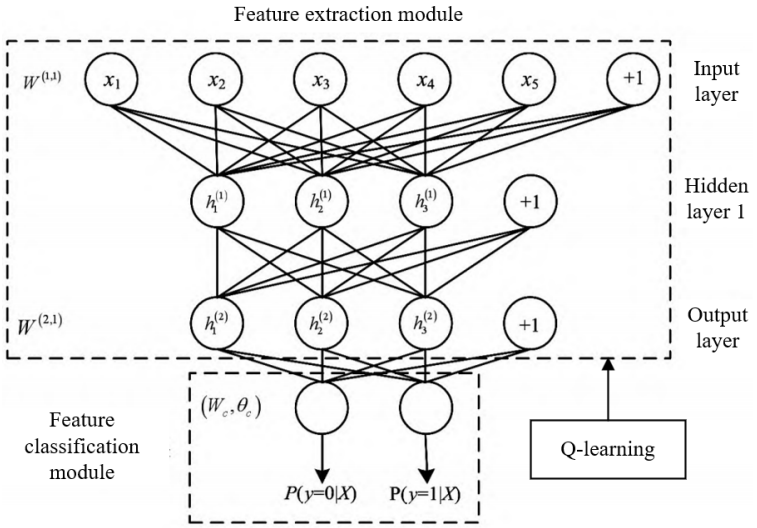


Fig. 6. Multi-source sensor feature classification model

The result of feature extraction plays a decisive role in the accuracy of subsequent data fusion and is the key to the successful completion of data fusion. Therefore, ADRS and Softmax classifier are combined to realize feature classification through cascading. The implementation process is as follows:

- Step 1. By training industrial big data samples to get parameter $(W^{(k,1)}, b^{(k,1)})|k = 1, \cdots, N_k$.
- Step 2. Through Q-learning processing of sample features, the best data features are obtained.
- Step 3. After training, data features are obtained. After supervised training and the function of Softmax classifier, output parameters (W_c, θ_c) are obtained.
- Step 4. Consider the parameters obtained in steps 2 and 3 as the original data. In the ADRS model, the supervised training algorithm is used to adjust the original data. Then, the adjusted sensor network and Softmax classifier [36] are used as the extraction module and classification module in the feature extraction model to complete the feature extraction and classification of multi-source sensor data.

In the process of solving problems, Q-learning algorithm generally uses the greedy strategy to make selection. This strategy selects the corresponding action with the maximum reward value in each selection, which is called greedy. However, it does not take the global consideration into account. In the initialization process, the Q value is set as the equal value or random value, that is, the learning is carried out in the environment without prior knowledge. At the same time, the learning rate is generally not dynamically adjusted after initialization, and the learning process is characterized by large decision-making space, slow learning speed and uncertain learning effect [37]. As a result, the result is easy to fall into local optimal and the optimal solution cannot be found. In this paper, Q-learning algorithm is improved, and a "Sigmoid "function is introduced as a dynamic selection strategy, so that the improved Q-learning algorithm can randomly select

actions in the early stage and dynamically select the optimal actions in the late stage. This improved method can improve the phenomenon that the greedy strategy falls into the local optimal in the early stage and may choose the sub-optimal action in the late stage of learning.

This paper introduces "Sigmoid" as an action selection strategy, and its expression is as follows:

$$P(s_t, a) = \frac{exp(Q(s_t, a))}{1 + exp(Q(s_t, a))}. \tag{7}$$

According to Equation (7), when Q value of the initial state is set to 0, each action has the same selection probability and is more random in the early stage of selection. When the learning reaches a certain level, the reward value of each action starts to stand out, and the probability P changes with the update of Q value. The system will gradually tend to choose the action with high reward value.

In this paper, three kinds of selection strategies are discussed, which are "completely random" strategy, "semi-random" strategy and greedy strategy. We introduce parameter B and have the following options.

1. When $B = 0$, then $P > B$. Implement the greedy dream strategy.
2. When $B = 1$, then $P < B$. Implement the "completely random" strategy, that is, the actions are randomly selected throughout the selection process.
3. When $B \in (0, 1)$. Implement the "semi-random" strategy, that is, a "Sigmoid "function is introduced as a dynamic selection strategy. When $P \geq B$, the algorithm carries out the exploration strategy and selects the action with the maximum current reward value. When $P < B$, the algorithm adopts a "completely random" strategy and selects actions at random.
4. Combined with the improvement objectives of this paper, the strategy selection when $B \in (0, 1)$ is implemented, and $B = 0.5$ is set through a large number of experiments.

5. Experiments and Analysis

Matlab platform is used to simulate and analyze the data fusion algorithm, and the facility industry monitoring system is taken as the application scenario. In order to highlight the performance of the proposed data fusion, STDF [38], CCA [39] and FKCI [40] are used for comparative analysis.

The simulation parameters are shown in Table 4, where the number of nodes and the network scope refer to the random distribution of 100 sensor nodes in the perception area within the range of $100m \times 100m$. In order to compare the efficiency of various data fusion algorithms, the un-optimized LEACH protocol is adopted, and the energy consumption of sending, receiving and fusion data of nodes is calculated according to the first type of wireless communication energy consumption model.

The classification error rate of feature extraction with each algorithm is shown in Table 5. Where n, d and c represent the network layer number, the dimension of input data and the number of data classification respectively. As can be seen from Table 5, for low-dimensional and low-category data, the classification error rate of proposed method is

Table 4. Experiments parameters

Parameter	Value	Parameter	Value
Network range	$100m \times 100m$	Packet length	2000bit
Note number	100	Cluster message length	100bit
Sink node coordinate	(50,50)	Packet head length	100bit
Initial node energy	0.5J	Maximum simulation number	2000
Convolution kernel size	3×3	Convolution step	1

basically the same as that of STDF and CCA. With the increase of the input data dimension, the number of parameters of CCA and FKCI increases sharply, and the performance begins to decline, resulting in a significant increase in the error rate. In contrast, the error rate of proposed method based on the depth model is always at a low level.

Table 5. Inaccuracy of feature extraction and classification with different algorithms%

Method	d=200, c=4	d=600, c=6	d=900, c=12	d=2500, c=12
STDF(n=4)	8.4	16.7	19.6	28.1
STDF(n=5)	8.1	14.2	16.2	25.9
CCA	7.8	13.7	15.9	24.6
FKCI	7.1	12.5	13.4	20.2
Proposed	6.4	8.2	8.9	10.5

The average time of feature extraction and classification process of each algorithm is shown in Table 6. The analysis shows that Proposed has higher execution efficiency and can achieve faster data fusion compared with STDF, CCA and FKCI. Based on the powerful dimension reduction ability of CNN, as the dimension of input data increases, Compared with the other two algorithms, the feature extraction time of Proposed is greatly shortened, showing the advantages of processing high-dimensional data.

Table 6. Average time of feature extraction and classification with different algorithms/ms

Method	d=200, c=4	d=600, c=6	d=900, c=12	d=2500, c=12
STDF(n=4)	4.71	7.59	9.12	38.37
STDF(n=5)	5.49	8.04	9.58	40.96
CCA	5.22	7.54	8.69	36.54
FKCI	4.27	6.93	8.44	32.82
Proposed	3.18	3.26	3.71	9.50

This paper considers the effect of different number of nodes on the performance of wireless sensor networks in the $200m \times 200m$ region. Assuming that the flow rate is 20 packets/s, the average accuracy rate (AAR), average delay time (ADT) and average energy consumption (AEC) of the data within 10s are calculated and normalized, and the result is shown in Figure 7 and table 7.

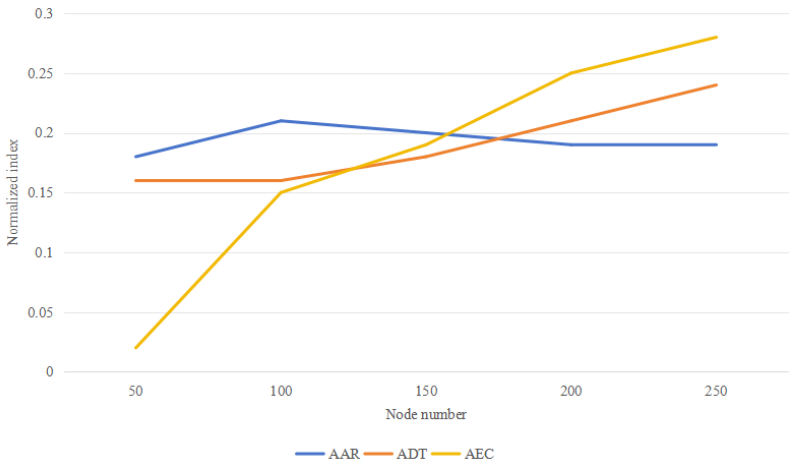


Fig. 7. Effect of node number on algorithm performance

Table 7. Effect of node number on algorithm performance

Node number	AAR	ADT	AEC
50	0.18	0.16	0.02
100	0.21	0.16	0.15
150	0.2	0.18	0.19
200	0.19	0.21	0.25
250	0.19	0.24	0.28

In a fixed area, AAR, ADT and AEC increase with the increase of nodes. When the number of nodes is increased from 100 to 150, the increase rate of data accuracy decreases, while the time delay and energy consumption increase significantly. Therefore, this paper selects 100 nodes as reference values for subsequent experiments.

MTS420/400 is used as the sensor plate model, the sensor plate includes humidity, temperature, PH, carbon dioxide concentration, smoke concentration and light sensor as a communication component, and the random high-frequency noise signal is added to the sensor to simulate the effect of the actual sensor noise. The parameters of the MTS420/400 are shown in Table 8.

Table 8. Effect of node number on algorithm performance

Parameter	MTS420/400
Temperature range/(°C)	-3941.3
Humidity range	0100
Signal-to-noise ratio/dB	01

This paper compares these four algorithms by counting the accuracy rate of receiving data from base station. Figure 8 shows the relationship between the accuracy rate of receiving data and the communication time of the base station. As can be seen from figure 8 and table 9, when FKCI, CCA, STDF and proposed method (node=25) are adopted for data fusion, the AAR values are 83.5%, 79.2%, 68.8% and 88.7%, respectively. Both the STDF and proposed method exclude unreliable data, while the CCA algorithm does not distinguish whether the data is correct or not, and all the data are weighted and fused, among which some incorrect data have high weights, affecting the overall accuracy rate. The algorithm in this paper considers not only the amount of data and time delay, but also the reliability of the data. High reliability will increase the weight value. Therefore, compared with the other three algorithms, the data accuracy of the proposed algorithm is increased by 5.2%, 9.5% and 19.9%, respectively.

Table 9. AAR/%

Node	Proposed	FKCI	CCA	STDF
5	85.1	80.2	79.5	76.4
10	86.7	79.3	77.5	72.5
15	88.6	81.4	76.4	71.9
20	89.4	81.7	76.9	72.3
25	88.7	83.5	79.2	68.8

In this section, the average end-to-end delay of data is used to measure the real-time performance of data. The shorter the delay, the better the real-time performance. In the simulation, data is set to eventually arrive at the base station. In this algorithm, there are three kinds of data with priorities ranging from 0 to 2, and each hop delay is required to be lower than 200ms, 400ms and 600ms respectively. Other algorithms do not consider

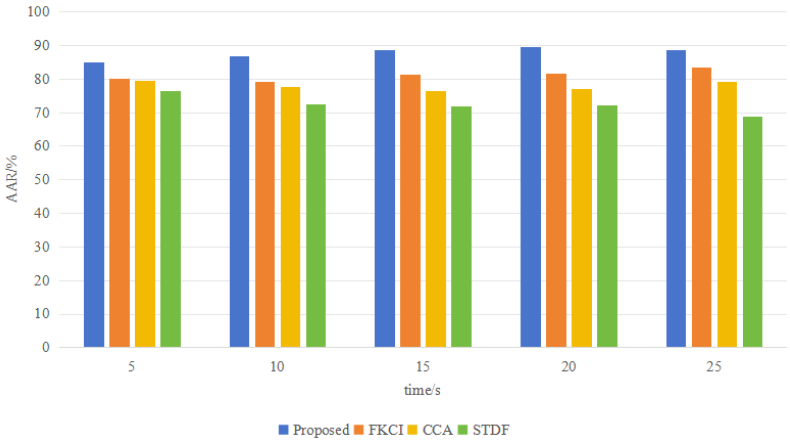


Fig. 8. The relationship between the accuracy rate of BS receiving data and the communication time

priorities. Figure 9 shows the relationship between data average end-to-end latency and traffic.

The wireless communication model [41,42] adopted in this paper is shown in equations (10-12).

$$E_{Tx}(l,d) = lE_{elec} + l\theta_{fs}d^2, d < d_0. \tag{8}$$

$$E_{Tx}(l,d) = lE_{elec} + l\theta_{amp}d^4, d \geq d_0. \tag{9}$$

$$E_{Rx}(l,d) = lE_{elec}. \tag{10}$$

Where E_{elec} represents the energy consumption of the transmitter running the transmitter or receiver, θ_{amp} represents the energy consumption of the signal amplifier, and d_0 is the threshold of free space energy attenuation. Through network simulation calculation, the relationship between the average energy consumption of nodes in the network and the communication time is obtained, as shown in Figure 10. As can be seen from Figure 10 and table 10, the average energy consumed by nodes gradually increases with the communication time. STDF algorithm uploads data directly, which results in a large number of data packets and consumes a lot of energy. Compared with STDF algorithm, CCA algorithm adopts fuzzy weighting method for data fusion, which reduces the amount of data and energy consumption. However, it does not distinguish whether the data is correct or not, and all the fusion processing, at the cost of data accuracy in exchange for less energy consumption. The algorithm in this paper removes part of the wrong data, and then merges and uploads, saving energy. FKCI algorithm in the initial phase out some of the inaccurate data, and within the cluster, the cluster between the fusion respectively, reduces the amount of data. In terms of energy consumption, compared with FKCI, STDF and CCA, the proposed algorithm is reduced by 21%, 9% and 4% respectively.

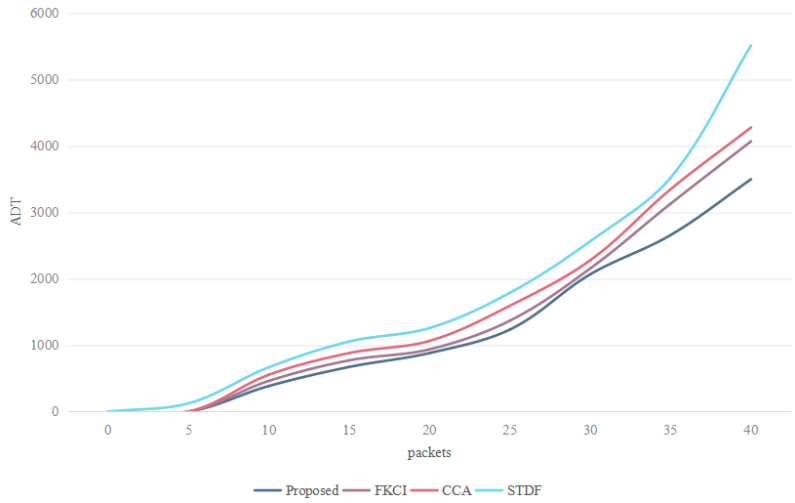


Fig. 9. Relationship between ADT and traffic

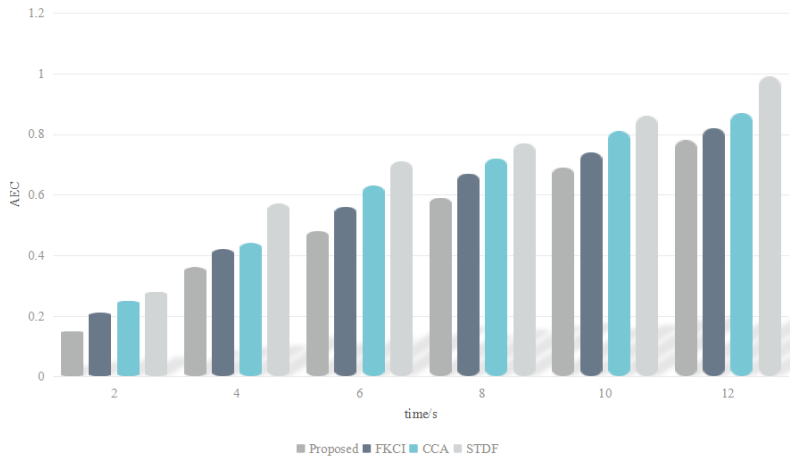


Fig. 10. Comparison of energy consumption

Table 10. Comparison of energy consumption

Time	Proposed	FKCI	CCA	STDF
2	0.15	0.21	0.25	0.28
4	0.36	0.42	0.44	0.57
6	0.48	0.56	0.63	0.71
8	0.59	0.67	0.72	0.77
10	0.69	0.74	0.81	0.86
12	0.78	0.82	0.87	0.99

6. Conclusion

Based on the characteristics of large amount and high complexity of data collected by industrial multi-source sensors, in order to meet the actual demand, the deep learning autoencoder theory and Q-learning are used to conduct in-depth research on the multi-source sensor data fusion method. The performance of the fusion method is studied from both theoretical and experimental aspects. The method can accurately extract and classify data features, reduce the energy consumption of data fusion, and greatly improve the quality of multi-source sensor data fusion. Therefore, the proposed method has obvious advantages and can provide a new reference idea for the development of multi-source sensor data fusion technology.

Acknowledgments. This paper was supported by Research Project of Zhengzhou University of Science and Technology: Vibration Analysis and Control Based on Flexible Joints and Loads of Industrial Robots (2022XJKY05); Key Teaching reform project of Zhengzhou University of Science and Technology, "Research on Cultivating Interdisciplinary Innovative Practice Teaching Based on Robot Technology" (2022JGZD03); 2023 Teacher Development Research Project of Zhengzhou University of Science and Technology, Research on Career Development Dilemma and Implementation Path of Young Teachers in Application-oriented Universities in the New Era (JSFZ-ZXKT2023004); Key research projects of universities in Henan Province, "Development and Research of Intelligent Gauge Detection System of Rail Transit Based on Dynamic Distance Measurement" (24B460027), "Application Research of Rhodes Correlation Failure Model Based on Big Data Acquisition in high-speed Cutting Analysis" (24B460028); "Research on Structure Optimization and vibration Damping Performance of high-frequency needling Equipment for Nonwoven Fabrics" (23B460011), "Research on Key Technologies of Supercapacitor Module Equalization and Safety Monitoring and early Warning based on Evidence Theory" (23B480003).

References

1. Malik P K, Sharma R, Singh R, et al. Industrial Internet of Things and its applications in industry 4.0: State of the art[J]. Computer Communications, 2021, 166: 125-139.
2. Franco J, Aris A, Canberk B, et al. A survey of honeypots and honeynets for internet of things, industrial internet of things, and cyber-physical systems[J]. IEEE Communications Surveys & Tutorials, 2021, 23(4): 2351-2383.
3. Wu Y, Dai H N, Tang H. Graph neural networks for anomaly detection in industrial internet of things[J]. IEEE Internet of Things Journal, 2021.
4. Liu C, Su Z, Xu X, et al. Service-oriented industrial internet of things gateway for cloud manufacturing[J]. Robotics and Computer-Integrated Manufacturing, 2022, 73: 102217.
5. Peng Li, Asif Ali Laghari, Mamoon Rashid, Jing Gao, Thippa Reddy Gadekallu, Abdul Rehman Javed, Shoulin Yin*, "A Deep Multimodal Adversarial Cycle-Consistent Network for Smart Enterprise System," in IEEE Transactions on Industrial Informatics, 2022, doi: 10.1109/TII.2022.3197201.
6. Nguyen T H T, Van Pham N, Nguyen V N N, et al. Forecasting Wind Speed Using A Hybrid Model Of Convolutional Neural Network And Long-Short Term Memory With Boruta Algorithm-Based Feature Selection[J]. Journal of Applied Science and Engineering, 2022, 26(8): 1055-1062.
7. Jiang B, Li J, Yue G, et al. Differential privacy for industrial internet of things: Opportunities, applications, and challenges[J]. IEEE Internet of Things Journal, 2021, 8(13): 10430-10451.

8. Shoulin Yin, Hang Li, Asif Ali Laghari, et al. A Bagging Strategy-Based Kernel Extreme Learning Machine for Complex Network Intrusion Detection[J]. EAI Endorsed Transactions on Scalable Information Systems. 21(33), e8, 2021. http://dx.doi.org/10.4108/eai.6-10-2021.171247
9. LeCun Y, Bengio Y, Hinton G. Deep learning[J]. nature, 2015, 521(7553): 436-444.
10. Gao S. A two-channel attention mechanism-based MobileNetV2 and bidirectional long short memory network for multi-modal dimension dance emotion recognition[J]. Journal of Applied Science and Engineering, 2022, 26(4): 455-464.
11. Xiong G, Ma W, Zhao N, et al. Multi-type diesel engines operating condition recognition method based on stacked auto-encoder and feature transfer learning[J]. IEEE Access, 2021, 9: 31043-31052.
12. Mahdavifar S, Alhadidi D, Ghorbani A. Effective and efficient hybrid android malware classification using pseudo-label stacked auto-encoder[J]. Journal of Network and Systems Management, 2022, 30(1): 1-34.
13. Pang G D, Lin Y C, Qiu Y L, et al. Dislocation Density-Based Model and Stacked Auto-Encoder Model for Ti-55511 Alloy with Basket-Weave Microstructures Deformed in $\alpha + \beta$ Region[J]. Advanced Engineering Materials, 2021, 23(4): 2001307.
14. Xie Y, Gong M, Wang S, et al. Community discovery in networks with deep sparse filtering[J]. Pattern Recognition, 2018, 81: 50-59.
15. Muzammal M, Talat R, Sodhro A H, et al. A multi-sensor data fusion enabled ensemble approach for medical data from body sensor networks[J]. Information Fusion, 2020, 53: 155-164.
16. Xiao F. Multi-sensor data fusion based on the belief divergence measure of evidences and the belief entropy[J]. Information Fusion, 2019, 46: 23-32.
17. Pan Y, Hong R, Chen J, et al. Performance degradation assessment of a wind turbine gearbox based on multi-sensor data fusion[J]. Mechanism and machine theory, 2019, 137: 509-526.
18. Fei S, Hassan M A, Xiao Y, et al. UAV-based multi-sensor data fusion and machine learning algorithm for yield prediction in wheat[J]. Precision agriculture, 2022: 1-26.
19. Tong W, Chu X, Jiang W, et al. A multimodal dual-fusion entity extraction model for large and complex devices[J]. Computer Communications, 2023, 210: 1-9.
20. Jiang, Y., Yin, S.: Heterogenous-view Occluded Expression Data Recognition Based on Cycle-Consistent Adversarial Network and K-SVD Dictionary Learning Under Intelligent Cooperative Robot Environment. Computer Science and Information Systems, vol. 20, no. 4, 2023. https://doi.org/10.2298/CSIS221228034J
21. Saranya S S, Fatima N S. IoT-Based Patient Health Data Using Improved Context-Aware Data Fusion and Enhanced Recursive Feature Elimination Model[J]. IEEE Access, 2022, 10: 128318-128335.
22. Qiu S, Zhao H, Jiang N, et al. Multi-sensor information fusion based on machine learning for real applications in human activity recognition: State-of-the-art and research challenges[J]. Information Fusion, 2022, 80: 241-265.
23. Li M, Chen H, Shi X, et al. A multi-information fusion "triple variables with iteration" inertia weight PSO algorithm and its application[J]. Applied Soft Computing, 2019, 84: 105677.
24. Li M, Chen H, Shi X, et al. A multi-information fusion "triple variables with iteration" inertia weight PSO algorithm and its application[J]. Applied Soft Computing, 2019, 84: 105677.
25. Wang Z, Chen D, Che X. Multi-kernel learning for multi-label classification with local Rademacher complexity[J]. Information Sciences, 2023, 647: 119462.
26. Mirbeygi M, Mahabadi A, Ranjbar A. Speech and music separation approaches-a survey[J]. Multimedia Tools and Applications, 2022, 81(15): 21155-21197.
27. Qin X, Ban Y, Wu P, et al. Improved image fusion method based on sparse decomposition[J]. Electronics, 2022, 11(15): 2321.
28. Satya Sreedhar P S, Nandhagopal N. Classification Similarity Network Model for Image Fusion Using Resnet50 and GoogLeNet[J]. Intelligent Automation & Soft Computing, 2022, 31(3).

29. Y. Liu, Y. Shi, F. Mu, J. Cheng, C. Li and X. Chen, "Multimodal MRI Volumetric Data Fusion With Convolutional Neural Networks," in IEEE Transactions on Instrumentation and Measurement, vol. 71, pp. 1-15, 2022, Art no. 4006015, doi: 10.1109/TIM.2022.3184360.
30. Brunke L, Greeff M, Hall A W, et al. Safe learning in robotics: From learning-based control to safe reinforcement learning[J]. Annual Review of Control, Robotics, and Autonomous Systems, 2022, 5: 411-444.
31. Gronauer S, Diepold K. Multi-agent deep reinforcement learning: a survey[J]. Artificial Intelligence Review, 2022, 55(2): 895-943.
32. Agarwal R, Schwarzer M, Castro P S, et al. Deep reinforcement learning at the edge of the statistical precipice[J]. Advances in neural information processing systems, 2021, 34: 29304-29320.
33. Wei W, Tang J. Cooperative Output Regulation By Q-learning For Discrete Multi-agent Systems In Finite-time[J]. Journal of Applied Science and Engineering, 2022, 26(6): 853-864.
34. Liu I J, Jain U, Yeh R A, et al. Cooperative exploration for multi-agent deep reinforcement learning[C]//International Conference on Machine Learning. PMLR, 2021: 6826-6836.
35. Han X, Han Y, Chen Q, et al. Distributed flow shop scheduling with sequence-dependent setup times using an improved iterated greedy algorithm[J]. Complex System Modeling and Simulation, 2021, 1(3): 198-217.
36. Jisi A and Shoulin Yin. A New Feature Fusion Network for Student Behavior Recognition in Education [J]. Journal of Applied Science and Engineering. vol. 24, no. 2, pp.133-140, 2021.
37. Li Q, Meng X, Gao F, et al. Approximate cost-optimal energy management of hydrogen electric multiple unit trains using double Q-learning algorithm[J]. IEEE Transactions on Industrial Electronics, 2021, 69(9): 9099-9110.
38. Fawzy D, Moussa S, Badr N. The Spatiotemporal Data Fusion (STDF) approach: IoT-based data fusion using big data analytics[J]. Sensors, 2021, 21(21): 7035.
39. Cui H, Li S. Controllable Clustering Algorithm for Associated Real-Time Streaming Big Data Based on Multi-Source Data Fusion[J]. Wireless Communications and Mobile Computing, 2022, 2022.
40. Zhen C. Using big data fuzzy K-means clustering and information fusion algorithm in English teaching ability evaluation[J]. Complexity, 2021, 2021.
41. K. Chen et al., "Recent advances in data-driven wireless communication using Gaussian processes: A comprehensive survey," in China Communications, vol. 19, no. 1, pp. 218-237, Jan. 2022, doi: 10.23919/JCC.2022.01.016.
42. M. Akrout, A. Feriani, F. Bellili, A. Mezghani and E. Hossain, "Domain Generalization in Machine Learning Models for Wireless Communications: Concepts, State-of-the-Art, and Open Issues," in IEEE Communications Surveys & Tutorials, vol. 25, no. 4, pp. 3014-3037, Fourthquarter 2023, doi: 10.1109/COMST.2023.3326399.

Xi Zhang graduated from Zhongyuan University of Technology with a master's degree in mechanical engineering in 2015. After graduation, he had been working in the field of vibration and noise control and automation control. Since 2017, he has been a full-time teacher, teaching assistant and lecturer of Zhengzhou University of Science and Technology.

Jiyue Wang graduated from chang'an university with a master's degree in mechanical engineering in 2013. He has eight years of mechanical structure and automatic control design experience. Since 2015, he has been a full-time teacher, associate professor and master tutor of zhengzhou university of science and technology. He is the director of industrial robot research institute, engaged in the application of industrial robot technology and hydraulic automatic control research.

Ying Huang graduated from Southwest Minzu University with a master's degree in electronic information in 2023. After graduation, she has been working in the field of topology, modeling, and control techniques of switching converters. Since 2023, she has been a full-time teacher of Zhengzhou University of Science and Technology.

Feiyue Zhu, 2022 graduated from Henan University of Science and Technology with a master's degree in power engineering, and has been a full-time instructor at Zhengzhou University of Science and Technology since 2022. After graduation, he has been engaged in the field of new energy automobile battery pack liquid cooled plate heat exchange analysis and structural optimisation research work.

Received: March 14, 2024; Accepted: June 29, 2024.

Computer Science and Information Systems 21(4):1651–1672 https://doi.org/10.2298/CSIS240314052P

Remora Optimization Algorithm-based Adaptive Fusion via Ant Colony Optimization for Traveling Salesman Problem

Lin Piao

Department of Education, Liaoning National Normal College
No. 45, Chongdong Road, Huanggu District, Shenyang 110032 China
pllnco@163.com

Abstract. The traditional ant colony optimization (ACO) is easy to fall into local optimal when solving large-scale traveling salesman problem (TSP), and the convergence speed is slow. In order to enhance the local search ability of ACO, speed up the efficiency of ACO and avoid the premature problem, this paper proposes a novel remora optimization algorithm-based adaptive fusion via ant colony optimization for solving TSP. Firstly, an improved K-means clustering method is used to obtain the best clustering results and the optimal solutions of each class quickly by adaptive clustering strategy based on the maximum and minimum distance and class density. By using an improved Remora optimization algorithm, adjacent classes are fused to effectively improve the accuracy of the initial solution. In addition, the initial solution is optimized by the k-opt strategy. Finally, the random recombination strategy is used to recombine the pheromone and random excitation to make the algorithm jump out of the local optimal as far as possible and improve the accuracy of the algorithm. The experimental results show that the proposed algorithm not only guarantees the accuracy of solution, but also improves the stability when solving large-scale TSP.

Keywords: TSP, ACO, Remora optimization algorithm, K-means clustering, Adaptive fusion.

1. Introduction

The travelling salesman problem (TSP) is a classic problem in the field of combinatorial optimization. It can be described as a businessman who starts from any city, visits every city without repetition or omission, and finally returns to the starting place. Its goal is to find a shortest path that includes all cities [1]. TSP problem is a NP problem, which has important reference value in the fields of vehicle routing, network routing and shop scheduling etc. Traditional methods such as linear programming, dynamic programming, branch and bound and k-opt have been proposed by many scholars. These methods can obtain better solutions for small-scale TSP problems. With the gradual increase of data scale, the above methods are easy to fall into local optimal solution when solving medium- and large-scale TSP problems, and the time complexity increases greatly. For the medium and large scale TSP problem, swarm intelligence algorithms such as ant colony optimization algorithm (ACO) [2], particle swarm optimization algorithm (PSO) [3], simulated annealing algorithm (SA)[4] and genetic algorithm (GA) [5] can obtain acceptable solutions within an expected time range, so more scholars pay attention to them.

For a classical TSP problem, if the brute force solution is used, the time complexity is $O(n!)$. However, with the continuous development of TSP problems, the required time will increase geometrically when the scale is larger. Even if the improved traditional algorithm is used to solve large-scale problems, it still needs a very large time cost and low efficiency. Therefore, in recent years, many scholars have improved ant colony algorithm in combination with other algorithms [5,6].

Reference [7] proposed the combination of PSO and ACO, using PSO to improve the parameters of ACO, and set up an evaluation function to guide it to converge in the direction of high index. This algorithm was beneficial to improve the accuracy of the algorithm, but the running speed was slow. Starting from genetic algorithm, Reference [8] made use of the global search ability of genetic algorithm and the fast convergence of ant colony algorithm to plan the path in a short time. However, its scale was small and the accuracy of the algorithm needed to be further improved. When selecting correction points, reference [9] combined greedy strategy with ant colony algorithm, and then used algorithm A^* to search out a solution as the initial solution of ant colony algorithm, and modified the updating mode of pheromone to further improve the accuracy. However, the iteration times were too few and the accuracy was not improved enough, resulting in the ant colony algorithm ending the iteration without sufficient optimization.

Recently, Sharma et al. [10] proposed an improved discrete butterfly optimization algorithm. Greedy strategy was used to initialize the population. 2-opt method, 4-opt method and simulated annealing method were combined to improve the optimization ability of the algorithm. However, the introduction of two-bridge experiment made the time complexity of the algorithm very high. Panwar et al. [11] introduced Hamming distance into grey wolf optimizer (GWO) to make it suitable for combinatorial optimization problems, and used 2-opt method to enhance local optimization ability. In the discrete cuckoo search algorithm, Xu et al. [12] used k-means algorithm to divide medium- and large-scale TSP problems into multiple small-scale TSP problems, so as to accelerate the operation efficiency of medium- and large-scale TSP problems. However, due to the lack of interaction ability between cities of different sizes, the final solution effect was poor. Wu et al. [13] proposed a discrete sparrow search algorithm (DSSA), which improved the ability of the algorithm to leap out of the local optimal solution by utilizing the division of labor mechanism of sparrow and the exploration and development capability of the global disturbance strategy balance algorithm. At the same time, the 2-opt method was used to enhance the local search ability, and the results were better than the discrete gray Wolf algorithm. k-opt method or its deformation methods are effectively used in the above algorithms.

Some scholars combined new technology to improve ant colony algorithm. Reference [14] combined K-means algorithm and ant colony algorithm to optimize TSP problem. By clustering, the problem was decomposed into several sub-problems, which were then connected to obtain solutions, so that the algorithm had better performance in dealing with large-scale problems. However, because K-means itself needed to determine the value of K, and only the accuracy of the fusion subclass solution depended on the problem itself, the algorithm depended on the type of TSP problem, the application range was small, and the accuracy was insufficient. Reference [15] used the least variance method to cluster cities, and proposed a new city connection method, and then used the circular search method to connect subclass solutions to obtain the final solution. This algorithm had a

good solution for small and medium-sized TSP problems. However, none of the above algorithms can dynamically determine the number of clusters, and the algorithm needs to be further improved when solving large-scale TSP problems.

In this paper, a novel remora optimization algorithm-based adaptive fusion via ant colony optimization for solving TSP is proposed. By using the characteristics of clustering, a large-scale TSP problem is decomposed into several sub-problems, and the subclass solutions are fused according to the fusion strategy of ant colony algorithm and Remora algorithm. Finally, k-opt strategy is used to further optimize the solution to obtain a higher precision solution. This algorithm can guarantee the accuracy of the solution and reduce the running time of the algorithm.

2. Related Works

2.1. TSP

The traveling salesman problem can be expressed as an undirected graph $G = (S, E)$. Where $S = 1, 2, \cdots, N$ represents the set of vertices. E is the edge set, edge $e_{ij} \in (i, j \in S, i \neq j)$. Let c_{ij} be the distance of e_{ij} and the decision variable be:

$$y(x) = \{1 \quad e_{ij} incurrent solution 0 \quad otherwise \tag{1}$$

The optimization objectives of this problem are as follows:

$$min \sum_{i=1}^{N} \sum_{j=1}^{N} x_{ij} c_{ij}. \tag{2}$$

$$s.t. \sum_{j \neq i} x_{ij} = \sum_{i \neq j} x_{ji} = 1, i \in S. \tag{3}$$

$$\sum_{j \neq i} x_{ij} \leq |V| - 1, V \subset S. \tag{4}$$

Where $|V|$ represents the number of vertices in the set S. The first constraint means that each vertex is connected to only two edges, and the second constraint means that no sub-loop is generated. Equation (2) is the target of optimization in this paper. This problem is a typical combinatorial optimization problem, and no definite algorithm can find the global optimal solution in the polynomial time range.

2.2. ACO for Solving TSP

TSP is a classical combinatorial optimization problem. Combinatorial optimization is to organize, group and sort some discrete events according to mathematical methods. The specific content of the traveling salesman problem is as follows: Suppose a traveling businessman needs to sell goods, he needs to visit all the cities and finally return to the origin, on the premise that these cities can only be visited once, and he needs to find the shortest visit path and its order. When the number of cities to be planned is larger, the time complexity and spatial complexity of the solution will rise and increase exponentially. With

the increasing complexity of the problem, it is no longer effective to use the traditional method of solving [16,17].

Nowadays, the mainstream algorithms for solving TSP generally fall into two categories: precise algorithms and heuristic algorithms. The former is suitable for small scale TCP, while the latter is suitable for medium and large scale TSP. The precise algorithms mainly include delimiting method and planning method. Heuristic algorithms include Genetic Algorithm (GA), Simulated Annealing algorithm (SA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), etc. Among them, ACO provides a new way to solve the TCP by using the inspiration of biology due to its evolutionary thought.

In the classical ant colony algorithm, the initial N ants are randomly placed in the initial city. The ants need to move on to the next city according to a certain probability. The ants decide on the next city according to the state transition probability rule (5).

$$P_{ij}^{k}(t) = \frac{[\tau_{ij}(t)^{\alpha}][\eta_{ij}]^{\beta}}{\sum_{k}[\tau_{ij}(t)^{\alpha}][\eta_{ij}]^{\beta}}. \tag{5}$$

Where, τ_{ij} is the pheromone value of the city and connection path in the process of t iteration. η_{ij} is the reciprocal value of the length of the connecting path between city i and j, and is a heuristic information. If the path length between city i and j is shorter, then the next city chosen by ants is more likely to be j. $Allowed_k$ represents the city that ant $k \in Allowed_k$ has not visited. α and β determine the importance of pheromones and heuristics, respectively. When all ants have completed a complete path construction, that is, an iteration is completed, the pheromone update will be carried out according to formulas (6) and (7).

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \sum_{k=1}^{m} \Delta\tau_{ij}^{k}(t). \tag{6}$$

$$\Delta\tau_{ij}^{k} = \{1/L_k(t) \quad if \quad ant \quad passes \qquad path \quad (i,j)0 \quad otherwise \tag{7}$$

Where ρ represents pheromone volatilization parameter. m is the total number of ants in each iteration, represents the pheromone value released by ant k on the path (i,j), and it is the travel path length of ant k. By setting the maximum number of iterations, the running of the algorithm ends when the number of iterations reaches the maximum value, and the optimal solution of the algorithm is obtained. If stagnation occurs during iteration, the process is initialized to get rid of stagnation and continue iteration until the set iteration upper bound is reached.

2.3. K-means Clustering Algorithm

K-means clustering algorithm is a widely used clustering algorithm, and it is also one of the most basic clustering algorithms [18]. The basic flow is as follows:

1. First, we need to determine the value of k, where k represents the number of classes to be clustered.
2. k data points are randomly selected from the data set as the center of mass.
3. For each point in the data, the distance between it and each centroid is calculated, k distances are obtained, the centroid of the smallest distance is selected, and the point is divided into this class.

4. After traversing all points, the center of mass of each class is recalculated and compared with the existing center of mass.
5. If the distance between the newly calculated centroid and the existing centroid is less than a certain set threshold, it means that the centroid tends to be stable. Without repeating the above steps, we can consider that the clustering has reached the expected result and the algorithm terminates. If not, it needs to iterate step 3 to step 5 until the center of mass meets the above conditions.

2.4. Max-min Distance Strategy

Max-min distance strategy is a common strategy for optimizing K-means clustering algorithm. The basic idea is to limit the distance between classes by setting thresholds, so as to achieve the purpose of solving different clustering problems without setting k value.

The basic steps are as follows:

1. Randomly select a point from the data as the initial clustering centroid z_l.
2. The sample point z_2 farthest from the first cluster centroid is selected as the second cluster centroid.
3. Calculate the distance between the remaining sample points and z_l and z_2, and find the minimum value between the two values, that is:

$$d_{ij} = ||x_i - z_j||, j = 1, 2. \tag{8}$$

$$d_i = min[d_{ij}], i = 1, 2, \cdots, N. \tag{9}$$

4. If

$$d_l = \max_i [min[d_{i1}, d_{i2}]] > \theta^* ||z_l - z_2||. \tag{10}$$

5. Assuming that there are k clustering centers, it calculates the distance between the sample point and each clustering center:

$$d_l = \max_i [min[d_{i1}, d_{i2}, \cdots, d_{ik}]] > \theta^* ||z_l - z_2||. \tag{11}$$

   If this is true, the fifth step is repeated until the above conditions are not met, that is, no new clustering centers appear.
6. According to the principle of minimum distance, all sample points are divided into each cluster center, that is, it calculates:

$$d_{ij} = ||x_i - z_j||. \tag{12}$$

3. Proposed TSP Solution Method

Aiming at the problem that the size of the traditional clustering algorithm k needs to be set in advance and the number of clusters cannot be changed according to the situation, the K-means clustering algorithm based on the maximum and minimum distance is chosen. The number of classes can be changed according to the set distance size, and once the

specified threshold is exceeded, the cluster is no longer. However, it is still necessary to set the threshold value in advance, which cannot achieve the function of adaptive clustering. And when the number of nodes in TSP problem is too large, it is difficult to find a balance between running time and precision by ant colony algorithm. This paper proposes a novel remora optimization algorithm-based adaptive fusion via ant colony optimization for solving TSP, which can effectively ensure the accuracy of the solution and improve the convergence speed of the algorithm.

3.1. Adaptive Clustering Strategy Based on Area Density

Aiming at the problems of slow convergence and poor quality of traditional ant colony algorithm in dealing with large-scale problems, adaptive clustering is adopted in this paper to exploit the advantages of ant colony algorithm in dealing with small-scale problems.

Because the maximum minimum distance strategy still needs to determine the clustering threshold, the concept of area density is proposed.

Using a rectangle so that all points are inside the rectangle, it determines the density according to the formula (13):

$$density = A \times S/count. \tag{13}$$

Here, $density$ represents the area density, S represents the rectangular area, $count$ represents the number of points in the TSP problem, and A represents the coefficient.

The results obtained by the area density strategy are used to replace the distance threshold in the max-min distance K-means clustering strategy, so that the clustering strategy can change the number of classes according to different types of problems and achieve the purpose of dynamically determining k.

Considering the impact of the size of each class on the total TSP problem, a class that is too large or too small will have a greater impact on the final result. In order to avoid the error caused by this situation, the amount of data within the class needs to be controlled.

Considering the influence of the number of points within a class on the result, this paper limits the number of points for each class to between 50 and 120, as shown in **Algorithm 1** for specific operations.

Specific treatment methods are as follows:

- First, the number of points for each class is calculated, if $p - number > 120$, the class is pulled out to re-cluster, the distance threshold is 0.9 times the previous, and the above process is repeated until all the class points are less than 120.
- Second, the number of points for each class is calculated here, if the number of points $p - number < 30$, the class nearest the center of mass is forced to merge, if the number of points is $30 < p - number < 50$, then try to merge with nearby classes. If the fusion result number $p - number < 120$, then fusion is allowed, if the number $p_number > 120$, it does not allow fusion, separates them into classes. Here, $p - number$ represents the number of city points in the class.

3.2. Initialization Optimization Based on ROA

ROA is a meta-heuristic optimization algorithm inspired by the parasitic feeding behavior of Remora in the ocean. Remora does this action by attaching itself to other hosts (such

Algorithm 1 Interclass point processing flow

Require: Input the results after the initial clustering
Ensure: Output the final clustering results

```
input data
if count > 120 then
    Narrowing τ, re-clustering
end if
if count < 50 then
    if count < 30 then
        Fusion directly
    else
        if Points after fusion < 120 then
            Perform fusion
        else
            Separate class
        end if
    end if
end if
Output result
```

as humpback whales or swordfish) to move, hunt, and evade predators. Similar to other meta-heuristic algorithms, ROA is mainly divided into three stages, namely initialization, exploration, and development.

Initialization stage At this stage, ROA generates initial populations in the search space in a random manner. The population vector is represented by X and contains N search agents with dimension d. The specific mathematical model is:

$$X = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,d} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,d} \\ \vdots & \vdots & \cdots & \vdots \\ x_{N,1} & x_{N,2} & \cdots & x_{N,d} \end{bmatrix} \tag{14}$$

Where N represents population size and d represents problem dimension.

Exploration stage By attaching to a swordfish, the Remora can move long distances within the search space, so the Remora's movement trajectory is the same as that of a swordfish. In addition, Remora also moves around its host in a small area to probe the surrounding feeding environment and determine whether to change the host.

– Swordfish strategy. Swordfish is one of the fastest fish in the world. By attaching to the swordfish, Remora mora quickly moves in the search space at a long distance. Its position update mode is as follows:

$$X(t+1) = X_{best}(t) - (r_1 \times (\frac{X_{best}(t) + X_{rand}(t)}{2}) - X_{rand}(t)). \tag{15}$$

Where $X_{best}(t)$ represents the global optimal position. r_1 indicates a random number between 0 and 1. $X_{rand}(t)$ represents random individuals in a population.

– Empirical attack. To maximize the benefits of feeding, Remora also moves around its host in small areas, surveying the surrounding prey distribution to determine if it needs to change hosts. Mathematically expressed as:

$$X(t+1) = X(t) + r_G \times (X(t) - X_{pre}(t)). \tag{16}$$

Where $r_G \in [0,1]$ represents random numbers that follow Gaussian distribution. $X_{pre}(t)$ represents the population position of the last iteration.

Development stage Remora does this process by attaching itself to a whale, so it hunts in the same way that whales do. If the host does not change, the remora attaches itself to the proper location and waits for the host to feed it.

– Whale strategy. At this point, the whales complete their hunting activities through the bubble net attack strategy. In turn, the remora performs its feeding by attaching itself to the whale. The mathematical model is as follows:

$$X(t+1) = D \times e^a \times \cos(2\pi\alpha) + X(t). \tag{17}$$

$$\alpha = r_2(a-1) + 1. \tag{18}$$

$$a = -(1 + \frac{t}{T}). \tag{19}$$

$$D = |X_{best}(t) - X(t)|. \tag{20}$$

Where D represents the position difference between prey and predator. $r_2 \in [0,1]$ represents a random number between 0 and 1. t and T represent the current and maximum number of iterations, respectively.

– Host feeding. When the host is not changed, Remora uses factor C to select a suitable adsorption site for the host to feed. The mathematical model is as follows:

$$X(t+1) = X(t) + A. \tag{21}$$

$$A = B \times (X(t) - C \times X_{best}(t)). \tag{22}$$

$$B = 2 \times V \times r_3 - V. \tag{23}$$

$$V = 2 \times (1 - \frac{t}{T}). \tag{24}$$

Where C represents remora factor, and its value is 0.1. r_3 represents a random number between 0 and 1. The procedure for executing the ROA algorithm is shown in **Algorithm 2**.

Algorithm 2 ROA

1: The search space, population size N, maximum number of iterations T, Remora factor C are initialized and the initial population is generated.
2: Perform boundary revision on the search agent and calculate its fitness value.
3: If $H(i) = 0$, the remora attaches to the whale and uses the bubble net attack strategy to prey. When $H(i) = 1$, Remora attaches to the swordfish to complete the long-distance movement.
4: Perform an empirical attack, replace if the new position produced is better than the original, and reset the random number set H, otherwise perform host feeding.
5: Determine whether the program meets the termination condition, if so, jump out of the loop and output the current best solution, otherwise return to step 2.

Adaptive weight factor improvement The exploration stage of ROA mainly relies on attachment to swordfish, as shown in Equation (15), which utilizes the best position of the population X_{best} and the random position X_{rand} respectively. However, X_{best} has good convergence performance but does not have extensive search ability, which cannot guide the population to fully explore the solution space, and it is easy to ignore potential areas. Therefore, an adaptive weight factor is proposed in this paper, which enables X_{best} to have adaptive disturbance ability, ensure convergence ability, and explore the surrounding neighborhood extensively, so as to find more potential solutions. The mathematical model is as follows:

$$w(t+1) = r_G \times \sin(\frac{\pi \times t}{4T}). \tag{25}$$

Generally, the traditional adaptive weight factor has less fluctuation in the curve in the early stage of iteration, which gives the particles less disturbance. The large fluctuation in the middle and late iteration provides the algorithm with superior global exploration performance and can guide the population to approach the optimal region or adjacent region. The improved location update method is:

$$X(t+1) = w(t) \times X_{best}(t) - (r_1 \times (\frac{X_{best}(t) + X_{rand}(t)}{2}) - X_{rand}(t)). \tag{26}$$

Lens opposition-based learning ROA implements iterative optimization by attaching to different hosts. Specifically, Remora attaches itself to a swordfish and a whale for global exploration and local exploitation, respectively. In addition, Remora can autonomously decide whether to change the host according to the surrounding environment to improve the success rate of feeding. However, the above search strategy mainly depends on the host. If the host's predatory ability is not strong (falling into the local optimal trap), then the Remora will also fall into it, resulting in a "precocious" phenomenon, which affects the optimization performance of the algorithm. In order to alleviate this trend, lens opposition-based learning (LOBL) [19] is introduced to generate the lens reverse solution in the search space by using the convex lens imaging principle, and the optimal individual is selected to enter the next iteration through the greedy strategy. The introduction of this strategy is conducive to improving the diversity of the whole population, that is, using the rich search information of the lens reverse solution to improve the ability of the population to jump out of the local extreme value and accelerate the convergence of the algorithm.

As shown in Figure 1, assuming that point x exists in one-dimensional space $[L_B, U_B]$, a lens with a height of h^* and a position of x^* is formed under the action of a lens. The y axis represents a convex lens, located at a midpoint in space.

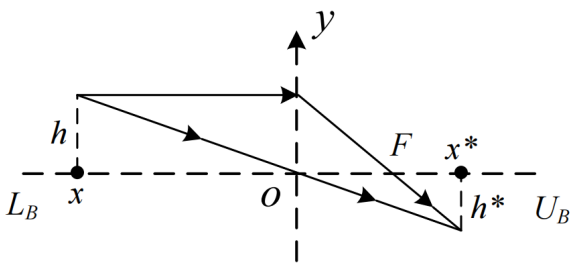


Fig. 1. LOBL

According to the principle of lens imaging:

$$\frac{(L_B, U_B)/2 - x}{x^* - (L_B, U_B)/2} = \frac{h}{h^*}. \tag{27}$$

L_B and U_B represent the upper and lower bounds of the search space. Let $k = h/h^*$, that is, formula (27) can be transformed as follows:

$$x^* = \frac{L_B + U_B}{2} + \frac{L_B + U_B}{2k} - \frac{x}{k}. \tag{28}$$

It can be seen from equation (28) that if $k = 1$, then it is equivalent to the reverse learning strategy, and it can also be said that the reverse learning of lens imaging is a special case of the reverse learning strategy.

The optimization process of the initial value based on the modified ROA is shown in **Algorithm 3**.

Algorithm 3 Optimization process of initial value based on modified ROA

1: Initializing parameter population size N, maximum number of iterations T, factor C of Remora.
2: The search agent boundary is modified, the fitness value is calculated, and the best individual and fitness value are recorded.
3: Calculate the lens inverse solution of the search agent, the fitness value and save the optimal individual according to the greedy strategy.
4: If $H(i) = 0$, the remora attaches to the whale and uses the bubble net attack strategy to prey. When $H(i) = 1$, Remora attaches to the swordfish to complete the middle and long distance adaptive movement.
5: Perform an empirical attack, replace if the new position produced is better than the original, and reset the random number set H, otherwise perform host feeding.
6: Judge whether the program meets the termination condition, if yes, jump out of the loop and output the current optimal segmentation threshold, otherwise return to step (2).

From the above introduction, we can see that ROA mainly has three stages, namely, the initialization stage, the exploration stage and the development stage. In the initialization stage, a search agent with population size N is generated in the search space, where the spatial dimension is d, then the operation complexity of this stage is $O(N \times d)$. In the exploration phase, ROA completes the position update through swordfish strategy and experience attack, then the operation complexity of this part is $O(N \times d)$. In the development phase, the location is updated through whale strategy and host feeding, and the computational complexity is $O(N \times d)$. If the exploration phase and development phase are repeated T times, the ROA operation complexity is $O(N \times d \times T)$. In the improved ROA, LOBL strategy is introduced to improve algorithm performance, and its complexity is $O(N \times d \times T)$. In addition, adaptive weight factors are introduced to enhance inter-group communication learning, and the complexity is $O(T)$. The rest is the same as the original algorithm. In summary, the improved ROA operation complexity is $O(N \times d \times T)$, which shows that the algorithm proposed in this paper does not improve the operation complexity.

3.3. Improved k-opt Method

Figure 2 shows the 2-opt optimization process. This method selects two non-adjacent points i and j to find a shorter path according to the following formula.

$$D(i, i+1) + D(j, j+1) > D(i, j) + D(i+1, j+1). \tag{29}$$

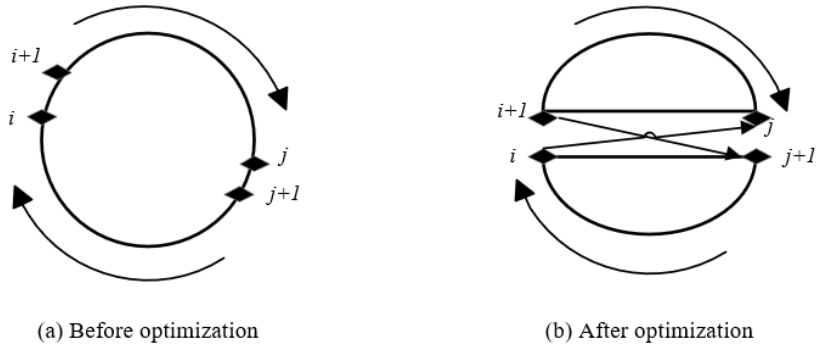
Where $D(i, j)$ represents the distance between two points i and j. If the condition is true, then the sequence $L(i+1, j)$ is reversed. $L(i, j)$ represents a sequence of paths in an existing path that starts with i and ends with j. Through continuous optimization of the existing sequence, a shorter sequence is finally constructed. Similarly, Figure 3 shows the 3-opt optimization process. This method selects three non-adjacent points i, j, and k to optimize the current path according to the following formula. If the condition is true, then the sequence $L(j+1, k)$ is reversed.

$$D(i, i+1) + D(j, j+1) + D(k, k+1) > D(i, k) + D(i+1, j+1) + D(j, k+1). \tag{30}$$

Isoart et al. [20] systematically discussed the advantages and disadvantages of different k values in the k-opt method. The 2-opt and 3-opt methods are optimization processes based on greedy search, so the above methods also inherit the shortcomings of greedy search, that is, they eventually lead to local optimal solutions. At the same time, the time complexity of 2-opt and 3-opt methods is closely related to the data size. In theory, the maximum number of sequence reversals by 2-opt method is $O(n^2)$, and the maximum number of sequence reversals by 3-opt method is $O(n^3)$.

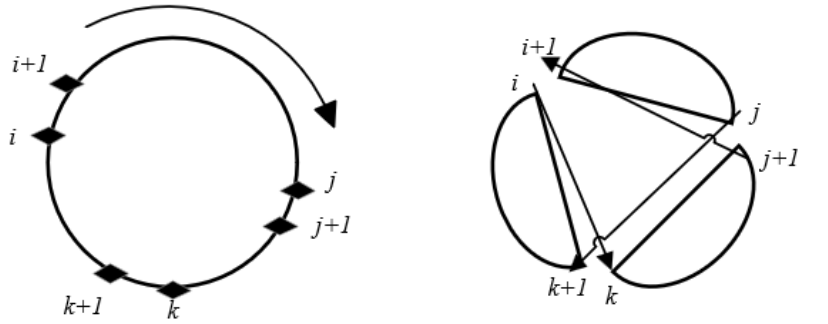
Improved 2-opt and 3-opt methods are proposed in order to alleviate the local optimal solution of 2-opt and 3-opt methods. In the improved 2-opt method, the set J of all subsequent nodes that are not adjacent to the current node i is found, and the set is constructed as follows:

$$Diff_2 = D(i, i+1) + D(J, J+1) - D(i, J) - D(i+1, J+1). \tag{31}$$



(a) Before optimization (b) After optimization

Fig. 2. Optimization process of 2-opt method



(a) Before optimization (b) After optimization

Fig. 3. Optimization process of 3-opt method

Where, the set $Diff_2$ represents the difference between all subsequent nodes and the current node after the sequence exchange. Find the maximum $max - S_2$ in the set $Diff_2$, if the condition $max - S_2 > 0$ is true, the sequence between node j and node i corresponding to $max - S_2$ is reversed, that is, the current sequence $L(i+1, j)$ is reversed. In the improved 3-opt method, the set K of all nodes that are not adjacent to the current node i and j is found and the set $Diff_3$ is constructed

$$Diff_3 = D(i, i+1)+D(j, Jj+1)+D(K, K+1)-D(i, K)-D(i+1, j+1)-D(j, K+1). \tag{32}$$

Compared with the original 2-opt method, the sequence reversal times of the improved 2-opt algorithm do not exceed $O(n)$. The improved 3-opt method reverses the sequence no more than $O(n/2)$. Secondly, for any node, the improved 2-opt method reverses the sequence at most 1 times, while the original 2-opt algorithm reverses the sequence at most $n - 2$ times.

3.4. Random Recombination Strategy

When using the ant colony guidance strategy, it is easy to fall into a local optimal situation. In order to avoid falling into local optimality as much as possible, this paper adopts random recombination strategy to make the algorithm jump out of stagnation. The details are as follows:

1. After N stagnation, the original pheromone matrix will be abandoned and a new one will be established.
2. The new pheromone matrix is set to the initial pheromone matrix, that is, every value except the diagonal is the initial value τ_0.
3. Each initial value of the new factor matrix is given a random multiplier incentive, and finally the new information matrix replaces the existing pheromone matrix, as shown in formula (33). Where the upper limit of magnification is the quotient of the current highest pheromone and the initial value τ_0.

$$\begin{bmatrix} 0 & x_{1,1}\tau_0 & x_{1,2}\tau_0 & \cdots & x_{i,j}\tau_0 \\ x_{1,1}\tau_0 & 0 & x_{2,1}\tau_0 & x_{2,2}\tau_0 & x_{i,j}\tau_0 \\ x_{1,2}\tau_0 & x_{2,1}\tau_0 & \cdots & x_{3,1}\tau_0 & x_{i,j}\tau_0 \\ x_{1,3}\tau_0 & x_{2,2}\tau_0 & x_{3,1}\tau_0 & 0 & x_{i,j}\tau_0 \\ x_{i,j}\tau_0 & x_{i,j}\tau_0 & x_{i,j}\tau_0 & x_{i,j}\tau_0 & 0 \end{bmatrix} \tag{33}$$

4. Experiment and Result Analysis

This algorithm adopts Matlab2017a simulation software, based on AMD Ryzen 5 3550H with Radeon Vega Mobile Gfx2.10 GHz processor, 8GB memory, and 64-bit Win10 operating system. This paper selects typical problem sets from TSPLIB, which are commonly used to test and optimize algorithms.

4.1. Parameter Setting

In this paper, the optimal parameters of ant colony algorithm are deeply studied, which is conducive to the application of ant colony algorithm in optimization problems. In order to make the experiment more reasonable, this paper conducts several tests, and the statistical results show that the parameter Settings in Table 1 are more reasonable and the experimental results are better.

Table 1. Basic parameter setting

α	β	ξ	ρ	$q0$	m	Tc
1	4	0.1	0.3	0.7	50	2000

α is the information heuristic factor, and the smaller α denotes the stronger randomness, but it is easy to fall into local optimal. The greater β denotes the greater probability

of ants to choose the local shorter path, which can accelerate the convergence speed, but the accuracy of the algorithm diversity and settlement is poor. ξ is the volatilization rate of local pheromone. ρ is the global pheromone volatilization rate. $q0$ is a pseudo-random factor that affects the algorithm's exploration of new paths. The smaller the $q0$, the greater the probability of ants to explore new paths; conversely, the larger $q0$ denotes the stronger greed. Ants concentrate on the shortest path, which speeds up the convergence speed, but results in poor search ability. m is the number of ants, and m ants search in parallel. The larger m denotes more solutions that can be obtained, and the accuracy of the algorithm is improved correspondingly, while the time complexity is increased. Tc indicates the total number of iterations.

The pheromone matrix needs to be initially stimulated in the ant guidance strategy. The value of N multiplier needs to integrate convergence speed, local optimization, initial solution accuracy, so the value of N should not be too high or too low. According to the experiment results in Table 2 and Table 3, N in this paper is set at 10.

Table 2. kroA200 solutions under different excitation multiples (Standard solution 29368)

N	Average solution	Optimal solution	Error rate %	Iteration number
2	29712	29564	0.66	908
5	29631	29548	0.62	1058
10	29631	29548	0.62	1058
20	29682	29564	0.66	1849
30	29789	29564	0.66	1843
40	29634	29548	0.62	1058
50	29682	29564	0.66	318

Table 3. Lin318 solutions under different excitation multiples (Standard solution 42029)

N	Average solution	Optimal solution	Error rate %	Iteration number
2	43839	43130	2.63	1979
5	43557	42775	1.79	1788
10	43167	42704	1.61	1842
20	43782	42834	1.92	1821
30	43786	42999	2.31	1988
40	43649	43001	2.32	1525
50	43709	43229	2.86	1459

4.2. Determination of Area Density Strategy Coefficient A

The area density strategy provides a limiting condition for the first clustering and a basic sample for the subsequent clustering adjustment, that is, the area density determines the number of iterations required for the subsequent adjustment. Therefore, in order to enable

subsequent operations to quickly reach the set standards, a reasonable coefficient of area density strategy should be selected to make the first clustering result as close as possible to the required result, that is, the number of each class is close to the value range of 50-120.

In this paper, different coefficients A will be selected, and the selection of coefficients will be judged by the results of the first clustering. As can be seen from Table 4 and Table 5, the optimal value range is [0.01,0.001]. In order to find better results, the value of this range is further refined to facilitate the solution of the optimal value.

As can be seen from Table 6 and Table 7, when $A = 0.002$, there is a better clustering result, and the number of iterations required for subsequent optimization will be effectively reduced. Therefore, the A of this paper is 0.002.

Table 4. kroA200 solutions under different excitation multiples

A value	Cluster number	Minimum number	Maximum number
1	2	100	100
0.1	2	100	100
0.01	2	100	100
0.001	11	12	27
0.0001	101	1	6

Table 5. Lin318 solutions under different excitation multiples

A value	Cluster number	Minimum number	Maximum number
1	1	318	318
0.1	2	150	168
0.01	2	150	168
0.001	9	19	57
0.0001	101	1	11

Table 6. kroA200 solutions under different excitation multiples

A value	Cluster number	Minimum number	Maximum number
0.002	5	32	49
0.004	2	100	100
0.006	2	100	100
0.008	2	100	100

4.3. Algorithm Performance Analysis

A. Performance analysis of adaptive clustering strategy

Table 7. Lin318 solutions under different excitation multiples

A value	Cluster number	Minimum number	Maximum number
0.002	4	61	89
0.004	2	150	168
0.006	2	150	168
0.008	2	150	168

In order to show the influence of adaptive clustering strategy on this algorithm, two sets of experiments were set in this paper to compare the accuracy of the solution, namely ROA+ant colony guidance strategy+random recombination strategy, and ROA+adaptive clustering strategy+ant colony guidance strategy+random recombination strategy, as shown in Table 8. The formula for calculating the error rate is:

$$\theta = \frac{L_{ROA-best} - L_{best}}{L_{best}}. \tag{34}$$

Where $L_{ROA-best}$ represents the theoretical optimal solution, L_{best} represents the optimal solution of the algorithm, and θ represents the error rate. Each group is tested 10 times, with the maximum iteration number of 2000 each time, and the excitation rate N is selected to be 10 times, and the average solution and error rate are calculated according to the results.

Table 8. Optimization scheme

Group number	Name	Scheme combination
A	ROA+ACO+RRS	ROA+ACO+random recombination strategy
B	ROA+AC+ACO+RRS	ROA+adaptive clustering+ACO+random recombination strategy

As can be seen from Table 9, in terms of average solution, optimal solution and error rate, the data in group B is better than that in group A, because group B provides a better initial solution, which makes the algorithm optimized around the initial solution at the initial stage, resulting in improved accuracy of the solution. In addition, the average solution of group B algorithm is better, which also shows that group B algorithm is more stable.

B. Performance analysis of ant colony guidance strategy

In order to show the influence of ant colony guidance strategy on this algorithm, three groups of experiments are set up in this paper to compare the accuracy of the solution, which are divided into specific ROA, ROA+ adaptive clustering+random recombination strategy, ROA+adaptive clustering+ant colony guidance strategy+ random recombination strategy. The calculation formula of error rate is consistent with that mentioned above, as shown in Table 10. Each group is tested 10 times, with the maximum iteration number of 2000 each time, and the excitation rate N is selected to be 10 times, and the average solution and error rate are calculated according to the results.

It can be seen from Table 11 that the optimal solution of group C is improved to A certain extent compared with group A and Group B. In addition, the solution of group A

Table 9. Performance comparison results of different optimization strategies

TSP instance	opt	scheme	average solution	optimal solution	error rate%
KroB150	26131	A	26352	26142	0.04
KroB150	26131	B	26217	26131	0
KroA200	29369	A	29705	29495	0.41
KroA200	29369	B	29506	29402	0.12
Pr226	80369	A	80982	80529	0.78
Pr226	80369	B	80658	80486	0.14
F1417	11861	A	12089	12016	1.31
F1417	11861	B	12128	11963	0.86

is fluctuated greatly because the ant colony guidance strategy is not added. In contrast, the average solution of group C is better than that of group A due to the addition of ant colony guidance strategy. Under the condition of ensuring the error rate, the addition of ant colony guidance strategy makes the algorithm more stable.

Table 10. Optimization scheme

Group number	Name	Scheme combination
A	ROA	ROA
B	ROA+AC+RRS	ROA+adaptive clustering+random recombination strategy
C	ROA+AC+ACO+RRS	ROA+adaptive clustering+ACO+random recombination strategy

Table 11. Performance comparison results of different optimization strategies

TSP instance	opt	scheme	average solution	optimal solution	error rate%
KroB150	26131	A	26625	26306	0.67
KroB150	26131	B	28817	28817	10.29
KroB150	26131	C	26217	26131	0
KroA200	29369	A	29772	29573	0.68
KroA200	29369	B	30341	30255	3.03
KroA200	29369	C	29506	29402	0.12
Pr226	80369	A	80982	80529	0.78
Pr226	80369	B	90195	90121	12.13
Pr226	80369	C	80658	80476	0.13

C. Performance analysis of random recombination strategy

In order to show the influence of random recombination strategies on this algorithm, two sets of experiments are set up to compare the accuracy of the solution, namely ROA+adaptive clustering+ant colony guidance strategy, ROA+adaptive clustering+ant colony guidance strategy+random recombination strategy. The calculation formula of error rate is consistent with the previous. Each group is tested 10 times, with the maximum iteration number of 2000 each time.

Table 12. Optimization scheme

Group number	Name	Scheme combination
A	ROA+AC	ROA+adaptive clustering+ACO
B	ROA+AC+ACO+RRS	ROA+adaptive clustering+ACO+random recombination strategy

As can be seen from Table 13, when the optimal solution is the same, the random recombination strategy improves the average solution significantly. As can be seen from U574 and Rat575, compared with group A, the random recombination strategy has a great improvement in the average solution and optimal solution for the evenly distributed TSP problem, which enables the algorithm to solve a wider range of problems with better solutions.

Table 13. Performance comparison results of different optimization strategies

TSP instance	opt	scheme	average solution	optimal solution	error rate%
F1417	11861	A	12565	11964	0.86
F1417	11861	B	12129	11964	0.86
U574	36905	A	41255	39588	7.27
U574	36905	B	39857	38751	5.01
Rat575	6773	A	7246	7009	3.47
Rat575	6773	B	7183	6924	2.22

4.4. Comparison with Traditional ACO

In order to compare proposed algorithm with traditional ACO, typical TSP problems such as eil51, eil76, KroA100, KroB100, KroB150, KraoA200, pr226, Lin318, fl417, pr439 are compared in this paper. Where, the time length refers to the time taken from the beginning of the algorithm to finding the optimal solution of the algorithm, as shown in Table 14.

Table 14. Comparison between traditional ACO and proposed method

No. TSP instance	No. opt	Proposed best	Proposed error rate%	Proposed time/m	ACO best	ACO error rate%	ACO time/m
eil51	426	**426**	0	5.31	426	0	3.86
eil76	538	**538**	0	6.86	538	0	4.96
KroA100	21282	**21282**	0	12.55	21282	0	9.79
KroB100	22141	**22141**	0	12.57	22141	0	10.71
KroB150	26131	**26131**	0	14.62	26306	0.68	12.65
KraoA200	29368	**29402**	0.11	15.95	29564	0.67	16.79
pr226	80369	**80475**	0.14	15.65	80529	0.79	17.46
Lin318	42029	**42496**	1.11	17.98	42705	1.61	25.77
fl417	11861	**11963**	0.85	18.61	12032	1.43	29.64
pr439	107217	**108602**	1.29	21.43	109422	2.07	30.65

It can be seen from Table 14 that in small-scale cities Eil51 and Eil76, proposed method and ACO can both find theoretical optimal solutions. In the medium-scale cities KroA100, KroB100, KroB150 and KroA200, the theoretical optimal solution can be found by proposed method except KroA200, while the traditional ACO can not find the theoretical optimal solution. In large-scale cities Lin318, F1417 and Pr439, none of the two algorithms can find the theoretical optimal solution, but proposed method has the highest accuracy. In addition, in terms of time, proposed method takes significantly longer than traditional algorithm to find initial solutions in small and medium scale due to the need to find current initial solutions. However, as the scale increases, the time complexity of traditional algorithm becomes higher and higher, while proposed method can rapidly converge and quickly find optimal solutions.

The rank sum test method can be used to clarify the differences between algorithms, and the results are shown in Table 15. When the data is less than 0.05, there is a significant difference between the two samples. It can be seen from the results in the table that there are obvious differences between the proposed algorithm and the traditional ACO in most TSP problems.

Table 15. Comparison table of rank sum tests

TSP instance	ACO	Proposed
eil51	**0.054**	0.049
eil76	0.013	0.037
KroA100	0.041	0.049
KroB100	**0.266**	0.063
KroB150	0.015	0.037
KraoA200	0.016	0.017
pr226	0.014	0.033
Lin318	0.003	0.007
fl417	0.001	0.004

4.5. Comparison with Other Advanced Methods

In order to further verify the performance of the algorithm, this paper compares the proposed method with the advanced algorithms in recent years. The comparison data includes NNM [21], DRSO [22] and SASL [23] algorithms, which all have better results in solving typical TSP problems. The experimental results are shown in Table 16. It can be seen from table 16 that the accuracy of proposed method is obviously improved when compared with the current newer algorithm, especially in the medium scale and large scale, the quality of the solution is obviously better than other algorithms. It can be concluded that the performance of proposed algorithm is better than that of other optimization algorithms.

Table 16. Comparison between different methods

No. TSP instance	No. opt	Proposed best	Proposed error rate%	NNM best	NNM error rate%	DRSO best	DRSO error rate%	SASL best	SASL error rate%
eil51	426	**426**	0	437	2.58	431	1.08	426	0
berlin52	7542	**7542**	0	7635	1.23	7545	0.03	7542	0
eil76	538	**538**	0	561	4.28	548	1.77	543	0.91
KroA100	21282	**21282**	0	22025	3.50	21286	0.02	21320	0.11
KroB100	22141	**22141**	0	23021	3.99	22372	1.03	22141	0.11
eil101	629	**629**	0	674	7.22	644	2.35	641	1.91
KroB150	26131	**26131**	0	27577	5.53	263508	0.83	26454	1.20
Rat195	2323	**2337**	0.6	2489	7.15	2599	8.12	2355	1.31
KraoA200	29368	**29402**	0.11	31829	8.83	29518	0.51	29599	0.71
pr439	107217	**108602**	1.29	116379	8.55	12148	10.21	99788	3.65

5. Conclusions

In order to solve the problem that ant colony algorithm convergence speed is slow and the effect is not ideal when solving large-scale problems, this paper proposes an ant colony algorithm-based ROA that integrates adaptive clustering and ant colony guidance strategy. Firstly, the large-scale problem is decomposed into several sub-problems through adaptive clustering, which speeds up the convergence speed and improves the accuracy of understanding. Secondly, several sub-problems are fused by ROA fusion strategy. Finally, the ant colony guidance strategy and opt were used to optimize the fusion results to further improve the accuracy of understanding. In the typical TSP problem, it is shown that the improved ant colony algorithm has higher convergence speed and solving precision than the traditional ant colony algorithm.

Nowadays, ant colony algorithm is widely used in travel salesman problem, robot path planning and other problems. The improved ant colony algorithm proposed in this paper can be applied in these fields, and has higher solving accuracy and convergence speed than other algorithms. The improved ant colony algorithm proposed in this paper has good performance on a large scale, but there is still room for improvement. The accuracy of this paper and the speed of the improved algorithm still need to be further improved when solving the very large scale traveling salesman problem. Next, we will consider optimizing the mother ant guidance strategy, so as to further improve the solution accuracy and convergence speed when dealing with very large scale problems.

References

1. Halim A H, Ismail I. Combinatorial optimization: comparison of heuristic algorithms in travelling salesman problem[J]. Archives of Computational Methods in Engineering, 2019, 26: 367-380.
2. Wang Y, Han Z. Ant colony optimization for traveling salesman problem based on parameters optimization[J]. Applied Soft Computing, 2021, 107: 107439.
3. Zheng R, Zhang Y, Yang K. A transfer learning-based particle swarm optimization algorithm for travelling salesman problem[J]. Journal of Computational Design and Engineering, 2022, 9(3): 933-948.
4. lhan $\dot{I}$, G*ö*kmen G. A list-based simulated annealing algorithm with crossover operator for the traveling salesman problem[J]. Neural Computing and Applications, 2022: 1-26.

5. Stodola P, Michenka K, Nohel J, et al. Hybrid algorithm based on ant colony optimization and simulated annealing applied to the dynamic traveling salesman problem[J]. Entropy, 2020, 22(8): 884.
6. Hao T, Yingnian W, Jiaxing Z, et al. Study on a hybrid algorithm combining enhanced ant colony optimization and double improved simulated annealing via clustering in the Traveling Salesman Problem (TSP)[J]. PeerJ Computer Science, 2023, 9: e1609.
7. Ilin V, Simić D, Simić S D, et al. A hybrid genetic algorithm, list-based simulated annealing algorithm, and different heuristic algorithms for travelling salesman problem[J]. Logic Journal of the IGPL, 2022: jzac028.
8. Zhang P, Wang J, Tian Z, et al. A genetic algorithm with jumping gene and heuristic operators for traveling salesman problem[J]. Applied Soft Computing, 2022, 127: 109339.
9. Li W, Xia L, Huang Y, et al. An ant colony optimization algorithm with adaptive greedy strategy to optimize path problems[J]. Journal of Ambient Intelligence and Humanized Computing, 2022: 1-15.
10. Sharma S, Chakraborty S, Saha A K, et al. mLBOA: A modified butterfly optimization algorithm with lagrange interpolation for global optimization[J]. Journal of Bionic Engineering, 2022, 19(4): 1161-1176.
11. Panwar K, Deep K. Discrete Grey Wolf Optimizer for symmetric travelling salesman problem[J]. Applied Soft Computing, 2021, 105: 107298.
12. Xu H, Lan H. An Adaptive Layered Clustering Framework with Improved Genetic Algorithm for Solving Large-Scale Traveling Salesman Problems[J]. Electronics, 2023, 12(7): 1681.
13. Wu C, Fu X, Pei J, et al. A novel sparrow search algorithm for the traveling salesman problem[J]. IEEE Access, 2021, 9: 153456-153471.
14. Kusumahardhini N, Hertono G F, Handari B D. Implementation of K-Means and crossover ant colony optimization algorithm on multiple traveling salesman problem[C]//Journal of Physics: Conference Series. IOP Publishing, 2020, 1442(1): 012035.
15. Hamdan B, Bashir H, Cheaitou A. A novel clustering method for breaking down the symmetric multiple traveling salesman problem[J]. Journal of Industrial Engineering and Management, 2021, 14(2): 199-218.
16. Sun Yang, Teng Lin, Yin Shoulin, Li Hang. A new Wolf colony search algorithm based on search strategy for solving travelling salesman problem[J]. International Journal of Computational Science and Engineering, 18(1), pp:1-11, 2019.
17. Teng L. Brief Review of Medical Image Segmentation Based on Deep Learning[J]. IJLAI Transactions on Science and Engineering, 2023, 1(02): 01-08.
18. Jing Yu, Hang Li, Shoulin Yin. New intelligent interface study based on K-means gaze tracking[J]. International Journal of Computational Science and Engineering, vol. 18, no. 1, pp. 12-20, 2019.
19. Majumdar P, Mitra S, Bhattacharya D. Honey Badger algorithm using lens opposition based learning and local search algorithm[J]. Evolving Systems, 2023: 1-26.
20. Isoart N, Régin J C. A k-Opt Based Constraint for the TSP[C]//27th International Conference on Principles and Practice of Constraint Programming (CP 2021). Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021.
21. Shi Y, Zhang Y. The neural network methods for solving Traveling Salesman Problem[J]. Procedia Computer Science, 2022, 199: 681-686.
22. Mzili T, Riffi M E, Mzili I, et al. A novel discrete Rat swarm optimization (DRSO) algorithm for solving the traveling salesman problem[J]. Decision making: applications in management and engineering, 2022, 5(2): 287-299.
23. Gong X, Rong Z, Wang J, et al. A hybrid algorithm based on state-adaptive slime mold model and fractional-order ant system for the travelling salesman problem[J]. Complex & Intelligent Systems, 2023, 9(4): 3951-3970.

Lin Piao is with the Department of Education, Liaoning National Normal College. He had published several papers related to his works. Research direction: math analysis, data analysis, artificial intelligence.

Received: March 14, 2024; Accepted: June 29, 2024.

Computer Science and Information Systems 21(4):1673–1697 https://doi.org/10.2298/CSIS240314053W

GAN-DNADE: Image Encryption Algorithm Based on Generative Adversarial Network and DNA Dynamic Encoding

Xi Wang

School of Artificial Intelligence and Software Engineering,
Nanyang Normal University, Nanyang 473061, China
352720214@qq.com

Abstract. Aiming at the problems such as small key space and incomplete color channel encryption in traditional image encryption, this paper proposes a novel image encryption algorithm based on generative adversarial network (GAN) and DNA dynamic encoding. This paper introduces GAN into random key generation, and uses GAN to learn and train the random key generated by hyperchaotic system. A parallel chaotic system is used to generate two sets of pseudo-random sequences, and DNA dynamic encoding is introduced to further transform them to generate a new sequence. The pixel-level diffusion and scrambling of images within and between channels are carried out by using random sequences. The experimental results show that the randomness of GAN can significantly expand the key space, and the proposed algorithm has significant advantages in the security and anti-attack ability of ciphertext images.

Keywords: image encryption, generative adversarial network, DNA dynamic encoding, hyperchaotic system.

1. Introduction

Image has been widely used in Internet because of its characteristics of large amount of information and convenient transmission. Due to the characteristics of parallel information processing, optical information system has shown incomparable advantages of digital information system in mass information processing, especially the more complex the image processing and the more information, the more obvious this advantage. Night vision technology uses the two conditions of low light and infrared light to convert the low light or infrared light information invisible to the human eye from the target into visible light that can be felt by the human eye through signal sensing, acquisition, processing and display technology [1,2]. Among them, low-light level TV and low-light night vision play an important role in the monitoring of the enemy's fixed targets and the warning and security of our important targets [3,4]. However, because the low-light level image contains key information such as public security, military and space technology, the importance of protecting low-light level image from malicious attacks has gradually become prominent, and the encryption technology of low-light level image has been widely concerned by the international academic circle. In addition, unlike the visible image, the low-light image has complex noise, fuzzy texture characteristics, uneven illumination, low signal-to-noise ratio and less gray level. The traditional visible image encryption technology directly uses

the random sequence generated by chaotic system to encrypt, which has shortcomings in security. With the deepening of the research, the security risks brought by the linear relationship of optical transformation are gradually exposed [5,6].

Chaotic systems are highly sensitive to initial values and control parameters, with ergodicity, pseudo-randomness and unpredictability, so they have been widely used in image encryption [7,8]. The nonlinear characteristics of chaotic systems can effectively offset the security risks caused by linear transformations in optical encryption. However, some chaotic based encryption schemes often have some shortcomings [9], such as short cycle length caused by the limited precision of computers, which is one of the important problems of chaotic key stream generators [10]. In order to solve the randomness and security of keys, researchers have proposed many key generation schemes for chaotic systems or various other technologies [11,12].

With the rapid development of artificial intelligence technology, deep learning has been widely used in various fields of information security. Image information security in the era of artificial intelligence has attracted more and more attention from researchers [13]. Since generative adversarial networks (GANs) were first proposed in 2014 [14], GANs have become a hot topic in computer vision [15], natural language processing, and malicious attack detection [16]. Due to the randomness and difference of GAN training results, this paper uses chaotic sequences as training sets and control generators to obtain data samples with more randomness than chaotic sequences, so as to avoid problems such as long iteration period and time consuming of high-dimensional chaotic systems. It can obtain more random data samples than chaotic sequences, and speed up the generation of key, which saves time for batch encryption of low-light level images, and can meet the needs of real-time processing of low-light level images containing important military secrets.

Chaotic systems began to be applied to image encryption in the 1990s. Since then, image encryption based on chaotic system has been paid more and more attention. In recent years, researchers are still committed to improving the rationality, practicability and security of chaotic image encryption. Sekar et al. [17] designed an image encryption algorithm based on Deoxyribo Nucleic Acid (DNA) sequence manipulation and space-time chaos. The plaintext image was first converted into a DNA matrix and then scrambled. After multiple DNA sequence operations, the resulting matrix was finally converted into a ciphertext image. In 2019, using discrete chaotic mapping, Wang et al. [18] proposed an image encryption algorithm based on DNA sequence manipulation. The algorithm scrambled DNA horizontally and generated ciphertext images by XOR of scrambles matrix. In 2023, Li et al. [19] designed an image encryption algorithm using DNA computing, chaotic system and hash algorithm, which carried out DNA-level scrambling and diffusion of plaintext images. With the development of image encryption technology, there are also researchers working on related cryptanalysis work. For scramble-only image encryption algorithms, Wang et al. [20] proved that such encryption algorithms were not secure. In 2018, for the hyperchaotic image encryption algorithm using DNA computing, Fang et al. [21] pointed out the rationality, practicability and security problems, and cracked it through a selective plaintext attack. For the relevant tests that were often used to evaluate the security of image encryption algorithms, Zhang et al. [22] confirmed that these tests were only necessary conditions to ensure the security of the algorithm, but not sufficient conditions. For the image encryption algorithm based on chaotic mapping

and DNA coding, Yang et al. [23] simplified it into a substitution-scrambling structure and used selective plaintext attacks to crack it. In addition, it was particularly noteworthy that Singh et al. [24] reviewed some representative works in the field of image encryption and its cryptanalysis, and classified and summarized these works. More importantly, they also pointed out some challenging problems in the field of image encryption and its cryptanalysis.

There is no doubt that the problems identified in cryptanalysis work will be taken seriously by researchers when designing new image encryption algorithms. Therefore, the cryptanalysis of image encryption algorithm can promote the development and perfection of image encryption technology. So a novel image encryption algorithm based on generative adversarial network (GAN) and DNA dynamic encoding is proposed in this paper.

2. Related Works

In view of the low efficiency and poor security of traditional encryption algorithms, researchers have proposed many efficient image encryption algorithms based on the randomness of low-dimensional chaotic systems [25]. However, the structure of low-dimensional chaotic systems is too simple, which makes the generated pseudo-random sequences vulnerable to attacks. Yu et al. [26] proposed an image encryption algorithm based on the hyperchaotic system. The pseudo-random sequence generated by the hyperchaotic system had good randomness and wide range, which made the security of the encryption method significantly improved compared with the traditional low-dimensional chaotic system, but the image anti-attack ability was weak. Ouguissi et al. [27] introduced scrambling methods such as Arnold transform and bit combination scrambling respectively to improve security, but changing pixel positions did not destroy the statistical characteristics of the original image, resulting in low anti-statistical attack capability of the encrypted image. Wang et al. [28] put forward a synchronous scrambling diffusion algorithm based on bit-plane decomposition technology. This method had good encryption effect, but did not consider the problem of uneven distribution of bit-plane information. Wei et al. [29] proposed a selective encryption method based on the bit-plane information distribution. The algorithm had small computation, but had general security performance and low key space.

2.1. Randomness of Neural Networks

An artificial neural network consists of a large number of neurons connected to each other, each node represents a specific output function, and the connections between different nodes represent a weighted value of the signal passing through that connection. The output of the network is determined by the connection mode, weight value and excitation function of the network [30].

Suppose that the input $X = (x_1, x_2, \cdots, x_n)$ is a vector of length n, and the output $Y = (y_1, y_2, \cdots, y_n)$ is the result of training the neural network. V_e and W_s are the weights of the input layer and the output layer. V_{oe} and $W_{0,s}$ are input and output errors respectively. The forward training model of the neural network structure is:

$$y_i = g(w_{0,s} + \sum_{j=1}^{n} Z_j \cdot W_{s,j}). \tag{1}$$

Where $w_{0,s}$ is the offset of the output layer. $W_{s,j}$ represents the weight of the hidden layer node j to the output layer. The calculation formula from the hidden layer to the output layer is as follows:

$$Z_j = F(V_{oe,j} + x_k \cdot V_{e,j}). \tag{2}$$

Where $1 \leq j \leq n$, $1 \leq k \leq n$. Z_j represents the $j-th$ hidden layer node. $V_{oe,j}$ represents the bias of the $j-th$ node. $V_{e,j}$ represents the weight of the $j-th$ node. In order to make the network more powerful and generate a nonlinear mapping from input to output, the activation function selects the nonlinear function $F(\theta) = tanh(0\theta)$ so that it can learn complex data. The training error is:

$$er_i = y_i - x_i. \tag{3}$$

The backpropagation gradient process is:

$$\begin{cases} \sigma_{0,i} = a \cdot er_i \quad i = 1, 2, \cdots, n \\ \sigma_{h,j} = Z_j(1 - Z_j) \sum_{i=1}^{n} \sigma_{0,i} W_{s,j} \quad k = 1, 2, \cdots, n \end{cases} \tag{4}$$

The training process iterates by updating each weight and bias by backpropagating the gradient. Since there are many different neuron connections in a neural network, the overall behavior of the network system depends on the characteristics of each neuron and the interactions of different neurons. When the internal parameters of the neural network are randomly initialized, the stochastic gradient descent algorithm ensures the training randomization of different objects. Moreover, with the further complexity of the depth and structure of the neural network, the complexity of the internal parameter combination increases exponentially, which makes the parameter change of each neuron affect the output of the whole network. According to this characteristic, the randomness of neural network is used to further expand the random selection space of key and improve the security of encryption algorithm.

2.2. The Generation of Chaotic Sequences

For the plaintext image P with size $M \times N$, its hash value K is calculated by Keccak algorithm [31] and equally divided into 32 bytes, that is, $k_1, k_2, \cdots, k_{32}$. Using the following formula:

$$h_i = ((k_{j+1} \oplus k_{j+2} \oplus k_{j+3}) + k_{j+4} + k_{j+5} + k_{j+6})/256. \tag{5}$$

$$\begin{cases} x_0 = x_0' + 1 + abs(round(h_1) - h_1) \\ y_0 = y_0' + 1 + abs(round(h_2) - h_2) \\ z_0 = z_0' + 1 + abs(round(h_3) - h_3) \end{cases} \tag{6}$$

The initial state values x_0, y_0, z_0 of the chaotic system are calculated. Where, $i = 1, 2, 3$, $y = 6(i-1)$, $\oplus$ is the bitwise XOR operation. x_0', y_0', z_0' are the given value., $abs(r)$ is the absolute value of the parameter r, and $round(r)$ is the rounding of the parameter r. Input x_0, y_0, z_0 into the Lorenz chaotic system to generate chaotic sequences x, y, z. The used Lorenz chaos system is:

$$\begin{cases} \dot{x} = a(y - x) \\ \dot{y} = -xz + bx - y \\ \dot{z} = xy - cz \end{cases} \tag{7}$$

System control parameter is $(a, b, c) = (10, 28, 8/3)$. Finally, we adopt,

$$\begin{cases} X = mod(floor(10^{10} \times (x - floor(x))), N) + 1 \\ Y = mod(floor(10^{10} \times (y - floor(y))), M) + 1 \\ Z = mod(floor(10^{10} \times (z - floor(z))), 256) \end{cases} \tag{8}$$

Convert x, y, z to the sequence X, Y, Z. Where $mod(r_1, r_2)$ is to perform modular r_2 operation on the parameter r_1, and $floor(r)$ is to round down the parameter r.

2.3. GAN Key Generation Method

The randomness of key has become an important factor affecting the security of cryptosystem. GAN is a class of artificial intelligence algorithms for unsupervised machine learning, which consists of two neural networks that compete with each other in a zero-sum game framework [32].

$$\min_G \max_D v(G, D) = E_{x \sim p_d}(x)[\log D(x)] + E_{z \sim p_z(z)}[\log(1 - D(G(z)))]. \tag{9}$$

Where G is the generating network. D is the discrimination network. $v(G, D)$ is the loss function. x is the training set. $E_{x \sim p_d}$ is the distribution of the actual input data. $\log D(x)$ is the judgment value of the discriminator. $D(x)$ is the D network model used to determine whether the actual data and training z are noise from the input G network model. $E_{z \sim p_z(z)}$ is the distribution of noise data. $\log(1 - D(G(z))$ is the judgment value of the generated data. $G(z)$ is the data generated for the G network model. Through continuous game of maximum and minimum values, G network model and D network model are optimized alternately until the two models reach Nash equilibrium.

Quantum dots and quantum cellular automata are novel nanoelectronic devices that transmit information through coulomb interactions [33]. Compared with traditional technology, quantum cellular automata has the advantages of ultra-high integration, ultra-low power consumption and leadless integration. In recent years, scholars at home and abroad have used the structure of cellular neural networks and quantum cellular automata to construct QCNN on the basis of Schrodinger equation. Due to the quantum interaction between quantum dots, QCNN can obtain complex linear dynamic properties from the polarizability and quantum phase of each quantum cellular automaton, which can be used to construct nanoscale hyperchaotic oscillators. For a QCNN coupling two elements, it can be described by the following differential equation:

$$\dot{g}_1 = -2a_1\sqrt{1-g_1^2}\sin h_1. \tag{10}$$

$$\dot{h}_1 = (-b_1(g_1-g_2) + 2a_1 g_1 \cos h_1)/\sqrt{1-g_1^2}. \tag{11}$$

$$\dot{g}_2 = -2a_2\sqrt{1-g_2^2}\sin h_2. \tag{12}$$

$$\dot{h}_2 = (-b_2(g_2-g_1) + 2a_2 g_2 \cos h_2)/\sqrt{1-g_2^2}. \tag{13}$$

In the formula, g_1, g_2 are the polarizability. h_1, h_2 are the quantum phase. a_1, a_2 are the proportional coefficients of the energy between the midpoints of each unit. b_1, b_2 are the weighted influencing factors of the differences in the polarizability of adjacent units. When $a_1 = a_2 = 0.28$, $b_1 = 0.7$, $b_2 = 0.3$, the system is in a chaotic state.

Because QCNN is a high order hyperchaotic system, the iteration speed is slow and the computation is large. The hyperchaotic random matrix generated by QCNN is input into GAN as the training set, and the random data is learned to be used as the secure GAN key pool of the encryption system. The random numbers generated after training have similar characteristics to the random numbers generated by the chaotic system, that is, the new random numbers also have chaotic characteristics. The sensitivity is that when different chaotic random sequences generated by QCNN with different initial values or control parameters are used as learning objects, GANs will also learn and generate completely different random numbers. However, the random numbers generated by GANs also have some characteristics that are different from those of chaotic systems. For example, because the learning and training process is unsupervised, the generated random sequence of numbers is not subject to human control, that is, it has non-repeatability. In short, GAN key generation method can effectively improve the speed of key generation on the premise of guaranteeing the key characteristics.

3. Proposed Image Encryption Method

The GAN key generation method is applied to a new low-light level image encryption algorithm, and a 2-D coordinate pointer calculation method is designed. In each encryption, a specific location is calculated from the key pool using the plaintext related control parameters and user-set parameters, and then different random sequences are obtained from the key as scrambled keys. Only one scrambling encryption framework is given in the description of the algorithm in this paper, but in practice each scrambling operation is to set different operations related to plaintext and users, and the attacker cannot break the encryption algorithm through known plaintext or select plaintext attacks.

The key generated by chaotic system is trained by GAN to obtain GAN key pool, and two chaotic random phase masks are selected from GAN key pool for diffusion stage. The overall encryption process is shown in Figure 1. RM_1 and RM_2 are two phase masks. The original image is scrambled first, and then double random phase fractional Fourier transform (FFT) [34] is performed to obtain the final ciphertext image.

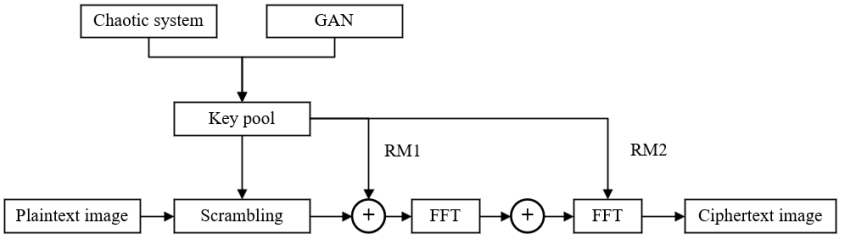


Fig. 1. Proposed encryption process

3.1. Pixel Replacement Based on DNA Dynamic Encoding

DNA dynamic encoding is performed on scrambled image T. Each pixel is encoded into 4 bases using different coding rules to obtain the DNA encoding sequence E. The encoding rule $R_{i,j}$ for each pixel depends on the location (i, j) of the pixel $T_{i,j}$ and the sequence Z.

$$R_{i,j} = (mod((i-1) \times N + j, 8) \oplus mod(Z_{(i-1)\times N+j}, 8)) + 1. \tag{14}$$

Where $i = 1, 2, \cdots, M$, $j = 1, 2, \cdots, N$. The DNA sequence was downloaded from the GenBank database and $4 \times M \times N$ bases are intercepted. DNA XOR operations are performed on these bases with the DNA encoding sequence E. Finally, encoding rule is used to decode the DNA of the operation result and reassemble it into an intermediate ciphertext image I.

3.2. Pixel Row Diffusion

It reorganizes the sequence Z into an $M \times N$ matrix in a column-first manner. Row spread operations are carried out in the direction of rows in the form of column vectors.

$$\begin{cases} C'_1 = I_1 \oplus I_N \oplus I_{N-1} \oplus Z_1 \\ C'_2 = I_2 \oplus C'_1 \oplus I_N \oplus Z_2 \\ C'_i = I_i \oplus C'_{i-1} \oplus C'_{i-2} \oplus Z_i \end{cases} \tag{15}$$

Where, C'_i is the $i-th$ column of the middle ciphertext image C' obtained after the row diffusion operation. I_i and Z_i are columns I and Z, respectively, $i = 3, 4, \cdots, N$. Then the column diffusion operation is carried out to obtain the final ciphertext image C.

$$\begin{cases} C_1 = C'_1 \oplus C'_M \oplus C'_{M-1} \oplus Z_1 \\ C_2 = C'_2 \oplus C_1 \oplus C'_M \oplus Z_2 \\ C_i = C'_i \oplus C_{i-1} \oplus C_{i-2} \oplus Z_i \end{cases} \tag{16}$$

Where C_i is row $i-th$ of C obtained after the column diffusion operation. C'_i and Z_i are row $i-th$ of C' and Z respectively, $i = 3, 4, \cdots, M$.

Case 1. When the chaotic sequence x is converted to sequence X, the modulus used by the original formula (6) is 256, not the column number N of the image.

Analysis. X in matrix form is used to implement line-by-line scrambling of pixels. Therefore, the Josef traversal variable step size should have a value range of $[1, N]$, not $[1, 256]$. Otherwise, when $N \gg 256$, the scrambling of pixels will be limited to a small range. The same is true for the transformation of chaotic sequence y.

Case 2. The DNA encoding rule R_{ij} is calculated according to the original formula (14), and its value range is $[0, 7]$. In addition, the chaotic sequence element used in the original formula (14) is $Z_{(i-1)\times N+1}$.

Analysis. According to **Algorithm 1**, the value range of the coding rule should be $[1, 8]$. In addition, in order to maximize the randomness and dynamics of the encoding rules, Z should be used more fully. Therefore, R_{ij} should be calculated using $Z_{(i-1)\times N+j}$ in equation (14).

Algorithm 1 Pixel diffusion effect elimination algorithm

Input: the ciphertext image C with size $M \times N$.
step 1: When $i \in [3, M]$, it repeats operation $C^1(i,:) = C(i,:) \oplus C(i-1,:) \oplus C(i-2,:)$.
step 2: Execute operation $C^1(2,:) = C(2,:) \oplus C(1,:) \oplus C^1(M,:)$.
step 3: Execute operation $C^1(1,:) = C(1,:) \oplus C^1(M,:) \oplus C^1(M-1,:)$.
step 4: When $j \in [3, N]$, it repeats operation $C^2(:,j) = C^1(:,j) \oplus C^1(:,j-1) \oplus C^1(:,j-2)$.
step 5: Execute operation $C^2(:,2) = C^1(:,2) \oplus C^1(:,1) \oplus C^2(:,N)$.
step 6: Execute operation $C^2(:,1) = C^1(:,1) \oplus C^2(:,N) \oplus C^2(:,N-1)$.

Case 3. The description of the diffusion process is inconsistent. In addition, if the number of rows or columns in the input image is less than 4, the diffusion process will not work properly.

Analysis. According to the display of row scrambling, it can be seen from the formula (15) that the row diffusion is carried out in the form of column vectors. That is, at the same time, the pixels on a column are correspondingly spread to other columns, which is to spread the column in the direction of the row. When the formula (15) is explained in the original paper, P_i also represents the i row of the image matrix. Similarly, columns can be diffused using equation (15).

The problem of the diffusion process not working properly is discussed only $N < 4$; $M < 4$ is similar. When $N = 1$, P_{N-1} in equation (15) is meaningless. When $N = 2$, equation (15) degenerates to:

$$P_i' = \begin{cases} (P_2 \oplus Q_1) mod 256, i = 1 \\ P_1 mod 256, i = 2 \end{cases} \tag{17}$$

Obviously, this is also the result of unreasonable and meaningless diffusion. When $N = 3$, equation (15) degenerates into:

$$P'_i = \begin{cases} (P_1 \oplus P_3 \oplus P_2 \oplus Q_1) mod 256, i = 1 \\ (P_2 \oplus P'_1 \oplus P_1 \oplus Q_1) mod 256, i = 2 \\ (P_2 \oplus P'_2 \oplus P'_1 \oplus Q_3) mod 256, i = 3 \end{cases} \tag{18}$$

According to equation (18), diffusion is irreversible. So if it knows P' and Q, it can only find P_3, it can not find P_1 and P_2. In addition, the module 256 operation in equation (15) is also redundant. Q_2 should also be used when diffusing P_2.

Case 4. GAN-DNADE downloads DNA sequences from the GenBank database and intercepts $4 \times M \times N$ bases.

Analysis. To encrypt a plaintext image with a size of $M \times N$, the encryptor must obtain at least $4 \times M \times N$ bases securely from the GenBank database. The same goes for the decryption party to complete the decryption. In other words, a plaintext image with a size of $M \times N$ is transmitted through an insecure channel, and GAN-DNADE is selected to achieve the encryption protection of the image, and both the encryptor and the decryptor need to securely download at least the same length of data from a third party. Obviously, this design makes encryption pointless. A reasonable design is to use a chaotic system to generate the required base data.

Case 5. By simple processing of ciphertext images, the encryption structure can be degraded from scramble-replace-diffusion structure to scramble-replace structure.

Analysis. As can be seen from equations (14) and (15), the diffusion process of GAN-DNADE not only has the effect of pixel diffusion, but also has the effect of pixel replacement caused by the difference with chaotic matrix Z. Since the ciphertext image C is known, $C'_i \oplus Z_i$ can be obtained according to equation (16).

$$C'_i \oplus Z_i = C_i \oplus C_{i-1} \oplus C_{i-2}. \tag{19}$$

Where $i = 3, 4, \cdots, M$. The same can be done for row 1 and row 2 as in equation (20).

$$\begin{cases} C'_1 \oplus Z_N \oplus Z_{N-1} \oplus Z_1 = C_1 \oplus (C'_N \oplus Z_N) \oplus (C'_{N-1} \oplus Z_{N-1}) \\ (C'_2 \oplus Z_2 \oplus Z_N = C_2 \oplus C_1 \oplus (C'_N \oplus Z_N) \end{cases} \tag{20}$$

At this time, although C' can not be directly obtained, the pixel diffusion effect of the column diffusion process can be completely eliminated, so that the processed ciphertext only has the replacement effect generated by the XOR with Z. Similarly, the pixel diffusion effect of the line diffusion process can also be eliminated, and finally a ciphertext image with only pixel replacement effect can be obtained. At this point, GAN-DNADE has degenerated into a scramble-replace replacement structure. However, the encryption effect of continuous secondary replacement is no different from that of a single replacement.

3.3. Multi-scale Feature Extraction and Fusion Module

In order to obtain more context information based on the existing image data and improve the feature extraction effect in the image, according to the idea of ASPP (Atrous Spatial

Pyramid Pooling) module, multi-scale features are extracted in parallel by using cavity convolution with different expansion rates, and the extracted multi-scale feature maps are fused. The specific structure of the multi-scale feature extraction and fusion module used by the network in this paper is shown in Figure 2. In figure 2, k is the convolution kernel size, s is the step size, p is the filling range, and d is the expansion rate. The features are extracted by using cavity convolution with expansion rates of 6, 12, and 18 respectively. In order to prevent the convolution degradation caused by excessive void rate, it is fused with the features after global average pooling. The use of this module increases the receptive field and obtains more context information, which helps to improve the feature extraction ability of the model for boundary details.

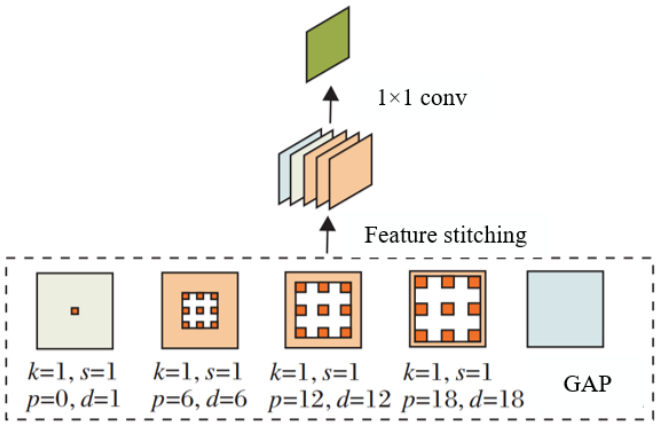


Fig. 2. Multi-scale feature extraction fusion diagram

In this paper, the features of different scales extracted by different expansion rates are spliced, and then 1×1 convolution is used to transform them into fixed-size feature maps. Because of the fusion of features of different scales, the ability of the network to extract features in the image is improved, and the loss of feature information is reduced, which is conducive to the subsequent sampling and recovery of detail information on the decoder.

3.4. Attention Module

At present, image semantic segmentation networks based on deep learning usually adopt multi-scale feature fusion or U-Net structure fusion of low-level and high-level semantic features, without considering the relation and correlation between each position or channel of feature map. Therefore, the network in this paper respectively conducts correlation modeling for the location dimension and channel dimension of the feature map. Firstly, Non-local [35] structure is used to achieve correlation modeling for the location of the feature space. Then a Non-local structure is used to model the correlation between channels, and the output results of the two modules are added and fused to obtain the global dependency relationship between features, and improve the ability of the network

to distinguish between normal organization areas, abnormal organization areas and backgrounds. It improves the segmentation effect of the transition region between normal and abnormal organization, and alleviates the adverse effects of void multiple on network learning to a certain extent.

In order to make use of the correlation between features of different regions and enhance each other's expression of their features, the network in this paper uses a modeling method similar to that in Non-local to model the location, as shown in Figure 3. Firstly, the correlation strength matrix between the features of any two points is calculated, that is, the original feature A is convolved to obtain the feature B (dimension is $(H \times W) \times C'$) and the feature C (dimension is $C' \times (H \times W)$), and the matrix product is carried out to obtain the correlation strength matrix between the features of any two points (dimension is $(H \times W) \times (H \times W)$). After softmax normalization operation, the attention diagram S of each position for other positions is obtained. The more similar the two features, the greater the response value. The response value in the location attention map is used as the weight to fuse the feature D, and the feature can be selectively enhanced or suppressed by the correlation between pixels.

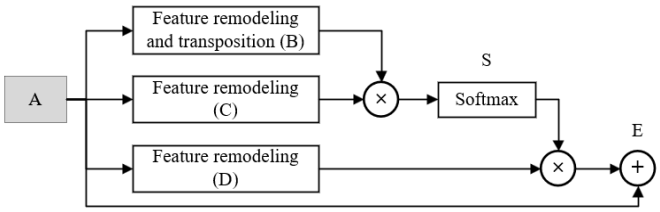


Fig. 3. Location attention module block diagram

For feature graph $S \in R^{N \times N}$,

$$S_{ij} = \frac{exp(B_i C_j)}{\sum_{i=1}^{N} exp(B_i C_j)}. \tag{21}$$

Where, C is the number of characteristic channels. H and W are the height and width of the feature graph. $N = H \times W$. S is the influence of the $i - th$ position on the $j - th$ position, that is, the degree of correlation/correlation between the $i - th$ position and the $j - th$ position, the larger the more similar.

Location attention module output feature map $E \in R^{C \times H \times W}$:

$$E_j = \alpha \sum_{i=1}^{N} (S_{ij} D_i) + A_j. \tag{22}$$

Where α is the scale coefficient, initialized to 0, and updated weights are assigned by gradual learning. E is the weighted sum of all location feature similarity and original location, with global context information.

The feature maps between different channels can be regarded as responses to specific categories, and there are certain dependencies between different responses. In order to

model this dependency explicitly, inspired by the Non-local module, the network in this paper conducts similar attention modeling for the channel dimension, as shown in Figure 4.

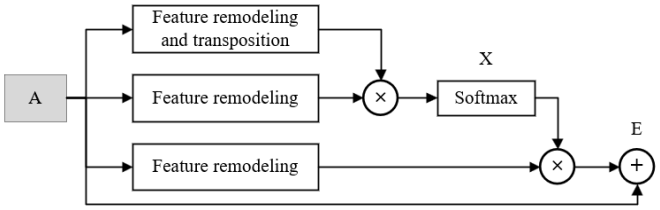


Fig. 4. Channel attention module diagram

By modeling the relationship between channels, the specific semantic response capability under channels is enhanced. The specific process is similar to location attention modeling, but the difference is that when obtaining the feature attention diagram $X \in R^{C\times C}$, dimensionality transformation and matrix product of any two channel features are performed to obtain the correlation strength of any two channels. The attention diagram between channels is also obtained through softmax operation. The attention force weighting between channels is used for fusion, so that each channel has a global association, and the features of stronger semantic response are obtained. The specific calculation process is as follows:

$$x_{ij} = \frac{exp(A_i A_j)}{\sum_{i=1}^{C} exp(A_i A_j)}. \tag{23}$$

$$E_j = \beta \sum_{i=1}^{C} (x_{ij} A_i) + A_j. \tag{24}$$

Where C is the number of characteristic channels. H and W are the height and width of the feature graph. x_{ij} is the influence of channel i on channel j. β is the scale coefficient, initialized to 0, updated by iterative learning, and the final output E is the weighted sum of each channel feature and the original feature. In this paper, we strengthen the sensitivity of the network to the boundary by means of intra-class feature response and interclass feature suppression.

3.5. Cryptographic Analysis

Based on the above analysis, GAN-DNADE can be described as:

$$C = f_3(f_2(f_1(P, K), K), K). \tag{25}$$

Where $f_1(r_1, r_2)$ represents the scrambling operation of input image r_1 under the control of secret key r_2. $f_2(r_1, r_2)$ indicates the replacement of r_1 under the control of r_2. $f_3(r_1, r_2)$ indicates the diffusion operation performed on r_1 under the control of r_2.

A simple processing of C can deform GAN-DNADE into a scramble-replace structure. Therefore, equation (25) can be further simplified as:

$$C = f_2'(f_1(P, K), K). \tag{26}$$

$f_2'(r_1, r_2)$ indicates the replacement of r_1 under the control of r_2. The following only discusses GAN-DNADE with the pixel diffusion effect eliminated. The scrambling procedure has no encryption effect on plaintext images with a single pixel value. For a plaintext image with a single pixel value, equation (26) can be simplified as:

$$O = f_2'(I, K). \tag{27}$$

Where I represents the input plaintext image of a single pixel value. O indicates the output ciphertext image. Therefore, $f_2'(r_1, r_2)$ can be determined using the plaintext image with a single pixel value and its corresponding ciphertext image, that is, the equivalent replacement matrix E^S can be determined. Taking a plaintext image of size 2×2 as an example, it uses the following formula:

$$P^0 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{28}$$

and corresponding ciphertext image:

$$C^0 = \begin{bmatrix} c_{0,1} & c_{0,2} \\ c_{0,3} & c_{0,4} \end{bmatrix} \tag{29}$$

It can determine the result of the replacement of 0 value pixel at each position. That is, the 0 value pixel is replaced by $c_{0,1}$ at $(1, 1)$, $c_{0,2}$ at $(1, 2)$, $c_{0,3}$ at $(2, 1)$, and $c_{0,4}$ at $(2, 2)$. Similarly, a plaintext image with a single pixel value of $v = 1, 2, \cdots, 255$ and its corresponding ciphertext image can be used to determine the replacement result of a pixel with a value of v at each position.

$$P^v = \begin{bmatrix} v & v \\ v & v \end{bmatrix} \tag{30}$$

$$C^v = \begin{bmatrix} c_{v,1} & c_{v,2} \\ c_{v,3} & c_{v,4} \end{bmatrix} \tag{31}$$

By stretching all C^v into one-dimensional row vectors and arranging them from top to bottom, an equivalent replacement matrix is obtained for any ciphertext image of size 2×2.

$$E^S = \begin{bmatrix} c_{0,1} & c_{0,2} & c_{0,3} & c_{0,4} \\ c_{1,1} & c_{1,2} & c_{1,3} & c_{1,4} \\ \vdots & \vdots & \vdots & \vdots \\ c_{255,1} & c_{255,2} & c_{255,3} & c_{255,4} \end{bmatrix} \tag{32}$$

Therefore, for any ciphertext image with a size of 2×2 and generated by K, the pixel replacement effect can be eliminated by E^S. Similarly, the above cryptanalysis procedure can be applied to a ciphertext image of size $M \times N$ to construct an equivalent replacement matrix of size $256 \times (M \times N)$.

$$E^S = \begin{bmatrix} c_{0,1} & c_{0,2} & \cdots & c_{0,M\times N-1} & c_{0,M\times N} \\ c_{0,1} & c_{1,2} & \cdots & c_{1,M\times N-1} & c_{1,M\times N} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ c_{255,1} & c_{255,2} & \cdots & c_{255,M\times N-1} & c_{255,M\times N} \end{bmatrix} \tag{33}$$

In the first example of E^S, each pixel $c_i^*(i = 1, 2, \cdots, M \times N)$ of any ciphertext image is found, thus determining the value of each pixel before the replacement operation. In this way, the pixel replacement effect of the GAN-DNADE can be eliminated, further degrading it to only pixel scrambling effect. Scramble-only image encryption algorithm has been proved to be insecure. In this paper, a simpler method is chosen to obtain the equivalent scramble-matrix E^S. First, replace the first 255 pixels of an all-zero valued plaintext image of size $M \times N$ with $1, 2, \cdots, 255$. After obtaining the corresponding ciphertext image, the pixel diffusion and replacement effect of the ciphertext image are eliminated. The processed ciphertext image has only pixel scrambling effect, in which the corresponding position of each non-zero plaintext pixel can be found. In this way, it is possible to determine the position of the first 255 non-zero pixels of the plaintext image after scrambling. Similarly, the position of the remaining $M \times N - 255$ plaintext pixels can be determined after scrambling. The position of up to 255 pixels can be determined at a time, so a maximum of $floor(M \times N/255) + 1$ selected plaintext image and its corresponding ciphertext image are required to determine E^S.

As can be seen the above analysis, GAN-DNADE can be fully cracked using up to $256floor(M \times N/255) + 1$ selective plaintext images and their corresponding ciphertext images. The following is a specific selective plaintext attack algorithm.

Algorithm 2 Selective plaintext attack algorithm

Input: To restore the ciphertext image C of the plaintext image, the size of which is $M \times N$.
step 1: When $v \in [1, 256]$, it repeats operations:
(a) A plaintext image P^{v-1} with a single pixel value of $v-1$ is constructed, and its corresponding ciphertext image C^{v-1} is obtained by encryption.
(b) Calling the Algorithm 1 to eliminate the pixel diffusion effect of C^{v-1}, stretch C^{v-1} into a one-dimensional row vector, and let $E^S(v, :) = C^{v-1}$.
step 2: Determine the number of plaintext image q required to obtain the equivalent scrambling matrix E^P.
step 3: When $w \in [1, q]$, it repeats operations:
(a) Construct the $w - th$ selected plaintext image P^w required for E^P acquisition, and encrypt to obtain its corresponding ciphertext image C^w.
(b) Calling the Algorithm 1 to eliminate the effect of C^w pixel diffusion.
(c) Using E^S to eliminate the pixel replacement effect of C^w.
(d) Finding the corresponding position of each plaintext image non-zero value pixel in C^w and save it to E^P.
step 4: Algorithm 1 is invoked to eliminate the pixel diffusion effect of C and C^2 is obtained.
step 5: Using E^S to eliminate the pixel substitution effect of C^2, C^3 is obtained.
step 6: Using E^P to eliminate the pixel substitution effect of C^3, P^R is obtained.
Output: Recovering plaintext images P^R as well as E^S and E^P.

3.6. Image Decryption Process

The decryption process is the inverse process of the encryption process, which is mainly divided into three steps: pixel level inverse diffusion, reverse bit level scrambling diffusion, and pixel level inverse scrambling of the image. When the receiver receives the ciphertext image C', and the two sequences Z_1 and Z_2, the decryption process can be implemented by following steps:

Step 1. Decompose the secret image C' to the three-color channel, denoted as C_R, C_G and C_B, splice the color channel image and carry out pixel-level inverse diffusion to obtain the pixel matrix P. The pixel matrix is decomposed into 8 bit planes, and the bits between bit planes are diffused in reverse bit level.

Step 2. The vertical reverse diffuses bits of the bit planes 2, 4, 6, 8, while the lateral reverse diffuses bits of the bit planes 1, 3, 5, 7, converting bit-level bits to the pixel level.

Step 3. Channel image R is transformed by inverse Arnold to get P_R, channel image G is transformed by inverse Zigzag to get P_G, channel image B is transformed by inverse row and column to get P_B, and finally three color channel images are merged to get decrypted image.

4. Experimental Results and Analysis

Based on Pycharm 202.1 1.1 platform, three standard test images of 512×512 TIF format color Lena, Baboon and Peppers are selected for simulation test. Firstly, the proposed algorithm is used to encrypt the test image. The three color plaintext images and corresponding ciphertext images are shown in Figure 5. The encrypted image presents random noise distribution and no meaningful information can be obtained visually, indicating that the random effect of the encryption algorithm can ensure the security of ciphertext images.

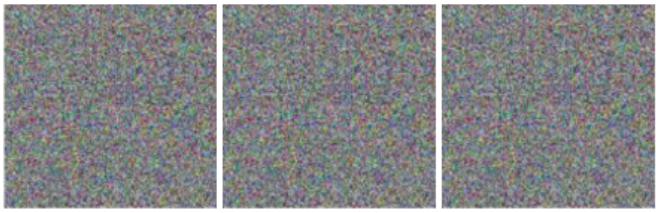
4.1. Key Space Analysis

Key space refers to the value space of the key of the encryption algorithm. In order to resist brute force attacks, the key space should be at least 2^{100} to ensure the security of encryption [11]. In the proposed encryption algorithm, 156-bit key is required as the initial value to generate two chaotic sequences, so the corresponding key space is 2^{312}. The parameters of each neuron in the neural network, such as weight, bias and other information, can be used as a key needed to encrypt an image. Since the weight of each neuron ranges from negative infinity to positive infinity, in order to facilitate the calculation of the key space, the weight of each neuron is quantified to a space of 256 sizes, so that the increased encryption complexity of each neuron is 2^8. The number of neurons used for training in this paper is 10, and each neuron has two weight information. Therefore, the key space expanded by 2^{160} after neural network randomization, corresponding to the increased key space of the two sequences is 2^{320}. Therefore, the total key space of this algorithm is 2^{632}.

The advantage of adding GAN is that the processing of neural network is equivalent to an extra layer of key protection. If the attacker cannot provide the training set, learning rate and the initial weight set by the neural network during random initialization, it is impossible to obtain the same neural network, and then the correct scrambling and diffusion



(a) Original image



(b) Encrypted image



(c) Decrypted image

Fig. 5. Three test images and corresponding ciphertext images

sequence cannot be obtained, and finally the decryption of ciphertext images cannot be realized. At the same time, the number, weight, bias and other information of neurons in the neural network will affect the size of the key space. The more neurons, the larger the key space, the better the performance of resisting violent attacks. Table 1 shows the key space comparison of several existing color image encryption algorithms. It can be seen that the key space of the proposed algorithm is higher than that of other methods. Therefore, the proposed algorithm has a better ability to resist brute force attacks.

Table 1. Key space comparison with different algorithms

Item	Reference [36]	Reference [37]	Reference [38]	GAN-DNADE
Key space calculation	$(10^{14})^4 \times 10^8$	$(10^{16})^7 \times 10^{15} \times 10^{17}$	$(2^{52})^{10}$	$(2^{156})^2 \times 2^{320}$
Key space size	$\approx 2^{535}$	$\approx 2^{515}$	2^{520}	2^{632}

4.2. Histogram Analysis

The histogram of the image is mainly used to reflect the probability of each gray level in the image, and is the main index to evaluate the ability of statistical analysis. The original image distribution has obvious features, and the image encryption system should do its best to eliminate these features and make the histogram of the encrypted image gently distributed as much as possible.

Figure 6 shows the histogram of pixel distribution of Lena image before and after encryption. Compared with the original image, the histogram distribution of the encrypted three-channel image is uniform, indicating that the frequency of different pixels in the ciphertext image is very close, and the distribution law of image pixels is difficult to analyze, thus ensuring the ability of the algorithm to resist statistical attacks and known ciphertext attacks.

4.3. Pixel Correlation Analysis

Usually digital images have strong correlations in horizontal, vertical, and diagonal directions. In order to avoid statistical attacks, encrypted images must eliminate the strong correlation of pixels in all directions to avoid attackers using channel correlation to restore the channel image and then restore the original image. In the experiment, 1000 pairs of horizontal, vertical and diagonal pixels of ciphertext three-channel images are selected to test the correlation between their adjacent pixels. When the plaintext image has a high correlation between adjacent pixels, the adjacent pixels of the plaintext three-channel image are centrally distributed on the diagonal, and the correlation coefficient is close to 1; otherwise, the value is close to 0. Table 2 compares the correlation coefficients of the plaintext images and their encrypted images of the four algorithms. Compared with other methods, the correlation coefficients of the proposed algorithm in the horizontal, vertical and diagonal directions of ciphertext images are significantly reduced, and the correlation distribution further presents a random state, indicating that the proposed algorithm has better encryption effect and scrambling performance in terms of pixel correlation.

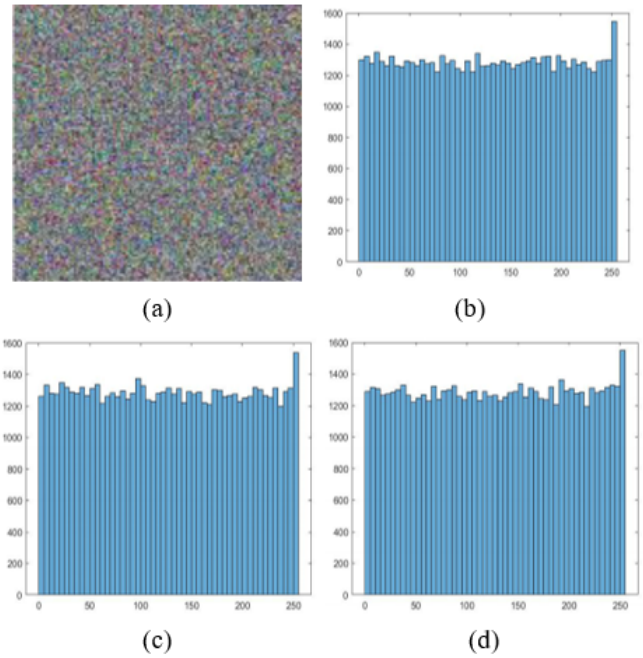


Fig. 6. Histogram analysis. (a) encrypted Lena image. (b) histogram of the red channel. (c) histogram of the green channel. (d) histogram of the blue channel

Table 2. Correlation coefficients of adjacent pixels of plaintext/ciphertext images of four algorithms

Image	Method	Horizontal	vertical	diagonal
Lena	Raw image	0.9752	0.9869	0.9627
Lena	Reference [36]	-0.0002	-0.0023	-0.0021
Lena	Reference [37]	-0.0003	0.0006	-0.0068
Lena	Reference [38]	0.0005	0.0010	0.0005
Lena	GAN-DNADE	-0.0001	-0.0014	-0.0012
Baboon	Raw image	0.9459	0.8672	0.8577
Baboon	Reference [36]	-0.0045	-0.0002	0.0001
Baboon	Reference [37]	-0.0031	-0.0002	0.0001
Baboon	Reference [38]	0.0014	0.0014	0.0029
Baboon	GAN-DNADE	-0.0013	0.0009	-0.0034
Peppers	Raw image	0.9649	0.9678	0.9572
Peppers	Reference [36]	-0.0008	-0.0067	0.0069
Peppers	Reference [37]	-0.0048	0.0052	0.0037
Peppers	Reference [38]	0.0045	0.0002	0.0025
Peppers	GAN-DNADE	-0.0032	-0.0001	-0.0031

4.4. Information Entropy Analysis

Information entropy is one of the important indicators to measure system uncertainty, which is mainly used to evaluate the unpredictability of color images. If the image pixels with gray level of 256 are evenly distributed, the maximum theoretical information entropy is 8.

In order to test the ability of the proposed algorithm to resist entropy attack, Lena, Baboon and Peppers were selected for the test in the experiment. The three images were encrypted and the information entropy was calculated respectively. The average information entropy of the three-channel ciphertext image is shown in Table 3. It can be seen that the average information entropy of the color channel images R, G and B of the ciphertext image is close to the ideal value 8, indicating that the encrypted image obtained by the algorithm in this paper is similar to other methods and has high uncertainty, which is sufficient to resist the information entropy attack.

Table 3. Average entropy results

Test image	Reference [36]	Reference [37]	Reference [38]	GAN-DNADE
Lena	7.9982	7.9996	7.9993	7.9997
Baboon	7.9979	7.9991	7.9988	7.9995
Peppers	7.9975	7.9992	7.9994	7.9995

4.5. Analysis of Anti-attack Capability

Selective plaintext attack among various attack types has obviously stronger cracking ability. If the encryption algorithm can resist selected-plaintext attacks, it can also effectively resist other types of attacks.

Relevant studies verify that if a secure encryption system can effectively generate random permutation and diffusion on all-black and all-white images, it has a high ability to resist plaintext attacks. To this end, the experiment verifies the anti-selective plaintext attack capability of the algorithm by encrypting all-black and all-white images, and the results are shown in Table 4. According to the security analysis results in Table 4, the proposed algorithm can effectively generate permutation and diffusion sequences to realize image encryption, and each evaluation index is close to the theoretical value of ciphertext image security, indicating that the proposed algorithm has a high ability to resist selected plaintext attacks.

Anti-differential Attack Analysis The number of pixel change rate (NPCR) and the unified average changing intensity (UACI) are important indicators for testing differential attacks.

$$NPCR = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} D(i,j) \times 100\%. \tag{34}$$

Table 4. Encryption security analysis of all black/all white images

security analysis	all black	all white
Horizontal correlation	0.0022	-0.0009
Vertical correlation	-0.0017	0.0008
Diagonal correlation	-0.0034	0.0018
NPCR/%	99.6202	99.6329
UACI/%	33.4552	33.4356
information entropy	7.9993	7.9994

$$UACI = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} \frac{|C_1(i,j) - C_2(i,j)|}{255} \times 100\%. \tag{35}$$

$$D(i,j) = \begin{cases} 0, C_1(i,j) = C_2(i,j) \\ 1, C_1(i,j) \neq C_2(i,j) \end{cases} \tag{36}$$

Where $C_1(i,j)$ and $C_2(i,j)$ are the pixel values at the encrypted image (i,j) before and after randomly changing any pixel values of the image. A larger NPCR value indicates that the encryption algorithm is more sensitive to changes in the original image. Similarly, a larger UACI value also predicts a larger average change intensity of images. The experiment compares the NPCR value and UACI value of the ciphertext image after encryption and the ciphertext image generated after pixel change by randomly changing any pixel value of the image. The results are shown in Table 5. The average NPCR and UACI values of the proposed algorithm are 99.64% and 33.49% respectively. Compared with other methods, the performance of the proposed algorithm is improved to some extent, indicating that the proposed algorithm has improved the capability of anti-differential attack compared with the existing methods.

Table 5. Anti-differential attack analysis of 3 test images%

Image	Method	NPCR	UACI
Lena	Reference [36]	99.6524	33.4558
Lena	Reference [37]	99.6098	33.4567
Lena	Reference [38]	99.6574	33.4657
Lena	GAN-DNADE	99.6351	33.5158
Baboon	Reference [36]	99.6086	33.4467
Baboon	Reference [37]	99.6061	33.4583
Baboon	Reference [38]	99.6604	33.4660
Baboon	GAN-DNADE	99.6295	33.4616
Peppers	Reference [36]	99.6321	33.4519
Peppers	Reference [37]	99.5999	33.3697
Peppers	Reference [38]	99.6285	33.4639
Peppers	GAN-DNADE	99.6397	33.4781

Anti-noise Attack Analysis In order to verify the robustness of images, the experimental data of Lena ciphertext images under different intensity noise attacks are compared with those of other algorithms. The results of anti-noise attacks are shown in Table 6.

Table 6. Anti-noise attack analysis for Lena

Salt-and-pepper noise	0.01	0.05	0.10
MSE (GAN-DNADE)	84.32	408.74	803.57
PSNR (GAN-DNADE)	28.90	25.33	21.54
MSE (Reference [36])	85.47	410.02	807.97
PSNR (Reference [36])	28.82	22.01	19.06
MSE (Reference [37])	107.31	538.93	1060.71
PSNR (Reference [37])	35.67	45.93	20.77
MSE (Reference [38])	96.14	255.71	896.32
PSNR (Reference [38])	27.94	21.58	16.94

With the increase of noise intensity, the means quare error (MSE) value of Lena image gradually increases, and the PSNR value gradually decreases. The results show that with the increase of noise attack intensity, the number of error pixels in decrypted images increases gradually, and the quality of image recovery decreases. After adding salt and pepper noise, the MSE and PSNR values of the proposed algorithm are improved compared with other literatures, indicating that the proposed algorithm has better robustness against noise attacks.

Anti-clipping Attack Analysis Ulteriorly, the experimental data of Lena ciphertext images under different reduction size attacks are compared with those of other algorithms. The experimental results are shown in Table 7 and Figure 7. As can be seen from Table 7, as the clipping size increases, more information is lost in ciphertext images. The larger the value of MSE is, the smaller the value of PSNR is, indicating that the number of error pixels in the clipped and decrypted image is increasing compared with the original image, and the recovery effect is worse. At the same time, under different clipping sizes, the MSE and PSNR values of the proposed algorithm are significantly better than those of other literatures, which indicates that the proposed algorithm has better robustness against clipping attacks.

In addition to the security of the encryption algorithm, the encryption speed is also an important index in the practical application process. Tables 8,9,10 lists the time required to encrypt the three images using this algorithm. As can be seen from tables, compared with other encryption algorithms, the proposed GAN-DNADE in this paper meets the needs of fast encryption.

5. Conclusion

In this paper, a new scheme based on GAN key generation is proposed, and a plaintext related image encryption algorithm is designed according to the generated GAN key pool.

Table 7. Anti-clipping attack analysis for Lena

Clipping attack	1/16	1/4	1/2
MSE (GAN-DNADE)	322.53	1098.64	2278.10
PSNR (GAN-DNADE)	21.98	21.89	14.93
MSE (Reference [36])	345.94	1275.10	2377.42
PSNR (Reference [36])	22.75	17.08	14.37
MSE (Reference [37])	579.99	2289.91	4578.35
PSNR (Reference [37])	20.58	20.82	11.59
MSE (Reference [38])	467.25	1788.63	3678.71
PSNR (Reference [38])	19.35	19.61	11.55

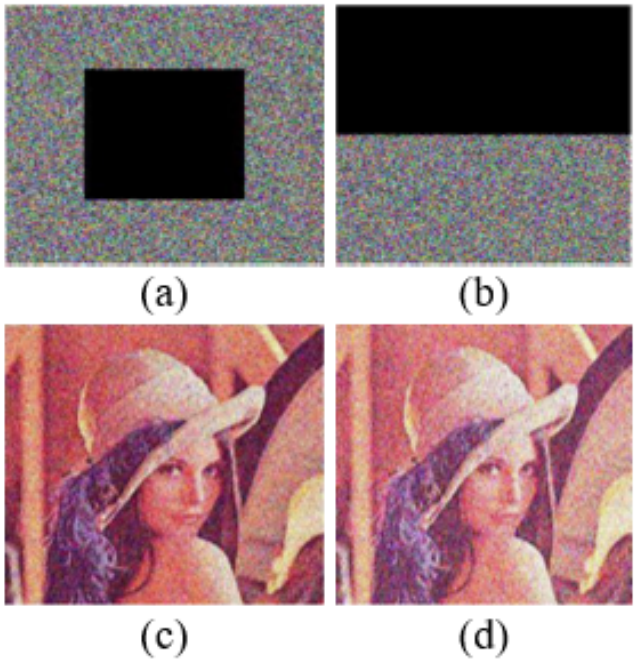


Fig. 7. The encrypted images and their decrypted images after data loss. (a) 1/16 clipping. (b) 1/2 clipping. (c) decrypted image with 1/16 clipping. (d) decrypted image with 1/2 clipping.

Table 8. Encryption time comparison for Lena

Method	size	time/s
Reference [36]	512×512	1.533
Reference [37]	512×512	1.247
Reference [38]	512×512	0.956
GAN-DNADE	512×512	0.165

Table 9. Encryption time comparison for Baboon

Method	size	time/s
Reference [36]	512×512	1.469
Reference [37]	512×512	1.188
Reference [38]	512×512	0.831
GAN-DNADE	512×512	0.154

Table 10. Encryption time comparison for Peppers

Method	size	time/s
Reference [36]	512×512	1.381
Reference [37]	512×512	1.092
Reference [38]	512×512	0.766
GAN-DNADE	512×512	0.122

Learning chaotic random keys of GAN-DNADE using GANs. The main conclusions are as follows:

1. The scheme in this paper trains and generates GAN key pool, which not only has the advantage of chaotic random key, but also has the feature of non-repeatability, which greatly improves the key generation speed and increases the security of the encryption system.
2. The random phase mask used in the algorithm based on GAN key pool is related to plaintext, so it can resist the attacks of known plaintext and selected plaintext.
3. The encryption scheme proposed in this paper can effectively improve the nonlinear characteristics of low-light level image encryption through chaotic system, so that the encryption scheme can effectively cope with various statistical analysis.
4. The key generation method is not only suitable for the image encryption algorithm proposed in this paper, but also can be applied to other encryption schemes.

References

1. Vandenbosch L, Fardouly J, Tiggemann M. Social media and body image: Recent trends and future directions[J]. Current opinion in psychology, 2022, 45: 101289.
2. S. Yin, H. Li, A. A. Laghari, T. R. Gadekallu, et al. An Anomaly Detection Model Based on Deep Auto-encoder and Capsule Graph Convolution via Sparrow Search Algorithm in 6G Internet-of-Everything[J]. IEEE Internet of Things Journal, 2024. DOI: 10.1109/JIOT.2024.3353337.
3. Jung C, Kwon G, Ye J C. Exploring patch-wise semantic relation for contrastive learning in image-to-image translation tasks[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022: 18260-18269.
4. T. Yu, L. Chen, N. Xu, H. Xu, X. Hu and X. Zhang, "Fengyun-3E Low Light Observation and Nighttime Lights Product," in IEEE Transactions on Geoscience and Remote Sensing, vol. 61, pp. 1-12, 2023, Art no. 4703612, doi: 10.1109/TGRS.2023.3292236.
5. Li Y, Cai F, Tu Y, et al. Low-Light Image Enhancement Under Non-uniform Dark[C]//International Conference on Multimedia Modeling. Cham: Springer Nature Switzerland, 2023: 190-201.

6. J. Liu, J. Zhang, S. Yin. Hybrid chaotic system-oriented artificial fish swarm neural network for image encryption. Evolutionary Intelligence, 16, 77-87, 2023. https://doi.org/10.1007/s12065-021-00643-5.
7. Li C, Sprott J C, Zhang X, et al. Constructing conditional symmetry in symmetric chaotic systems[J]. Chaos, Solitons & Fractals, 2022, 155: 111723.
8. Gao S, Wu R, Wang X, et al. A 3D model encryption scheme based on a cascaded chaotic system[J]. Signal Processing, 2023, 202: 108745.
9. S. Yin, H. Li, L. Teng, A. A. Laghari, V. V.a Estrela. Attribute-based Multiparty Searchable encryption model for Privacy Protection of Text Data[J]. Multimedia Tools and Applications, 2023. ttps://doi.org/10.1007/s11042-023-16818-4.
10. Gao X, Mou J, Xiong L, et al. A fast and efficient multiple images encryption based on single-channel encryption and chaotic system[J]. Nonlinear dynamics, 2022, 108(1): 613-636.
11. Zhou S, Wang X, Zhang Y, et al. A novel image encryption cryptosystem based on true random numbers and chaotic systems[J]. Multimedia Systems, 2022: 1-18.
12. M. Ji'e, D. Yan, S. Sun, F. Zhang, S. Duan and L. Wang, "A Simple Method for Constructing a Family of Hamiltonian Conservative Chaotic Systems," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 69, no. 8, pp. 3328-3338, Aug. 2022, doi: 10.1109/TCSI.2022.3172313.
13. Z. Zhang, H. A. Hamadi, E. Damiani, C. Y. Yeun and F. Taher, "Explainable Artificial Intelligence Applications in Cyber Security: State-of-the-Art in Research," in IEEE Access, vol. 10, pp. 93104-93139, 2022, doi: 10.1109/ACCESS.2022.3204051.
14. Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets[J]. Advances in neural information processing systems, 2014, 27.
15. Gao N, Xue H, Shao W, et al. Generative adversarial networks for spatio-temporal data: A survey[J]. ACM Transactions on Intelligent Systems and Technology (TIST), 2022, 13(2): 1-25.
16. Amin M, Shah B, Sharif A, et al. Android malware detection through generative adversarial networks[J]. Transactions on Emerging Telecommunications Technologies, 2022, 33(2): e3675.
17. Sekar J G, Periyathambi E, Chokkalingam A. Hybrid chaos-based image encryption algorithm using Chebyshev chaotic map with deoxyribonucleic acid sequence and its performance evaluation[J]. International Journal of Electrical & Computer Engineering (2088-8708), 2023, 13(6).
18. Wang Y, Cui W, Tao Y. A color image chunking encryption algorithm based on DNA and compound chaotic system[J]. Multimedia Tools and Applications, 2023: 1-21.
19. Li H, Zhang L, Cao H, et al. Hash Based DNA Computing Algorithm for Image Encryption[J]. Applied Sciences, 2023, 13(14): 8509.
20. Wang Z, Xu M, Zhang Y. Review of quantum image processing[J]. Archives of Computational Methods in Engineering, 2022, 29(2): 737-761.
21. Fang P, Liu H, Wu C, et al. A block image encryption algorithm based on a hyperchaotic system and generative adversarial networks[J]. Multimedia Tools and Applications, 2022, 81(15): 21811-21857.
22. X. Zhang, D. Gu, T. Wang and Y. Huang, "Old School, New Primitive: Toward Scalable PUF-Based Authenticated Encryption Scheme in IoT," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 42, no. 12, pp. 4569-4582, Dec. 2023, doi: 10.1109/TCAD.2023.3286260.
23. X. Zhang, D. Gu, T. Wang and Y. Huang, "Old School, New Primitive: Toward Scalable PUF-Based Authenticated Encryption Scheme in IoT," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 42, no. 12, pp. 4569-4582, Dec. 2023, doi: 10.1109/TCAD.2023.3286260.
24. Singh K N, Singh A K. Towards integrating image encryption with compression: A survey[J]. ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), 2022, 18(3): 1-21.

25. Zhang D, Shafiq M, Wang L, et al. Privacy-preserving remote sensing images recognition based on limited visual cryptography[J]. CAAI Transactions on Intelligence Technology, 2023. https://doi.org/10.1049/cit2.12164.
26. Yu J, Xie W, Zhong Z, et al. Image encryption algorithm based on hyperchaotic system and a new DNA sequence operation[J]. Chaos, Solitons & Fractals, 2022, 162: 112456.
27. Ouguissi H, Saadi S, Merrad A, et al. Hybrid scheme for safe speech transmission based on multiple chaotic maps, watermarking and Arnold scrambling algorithm[J]. Multimedia Tools and Applications, 2023, 82(1): 327-346.
28. Wang X, Zhao M, Feng S, et al. An image encryption scheme using bit-plane cross-diffusion and spatiotemporal chaos system with nonlinear perturbation[J]. Soft Computing, 2023, 27(3): 1223-1240.
29. Wei C, Li G. A selective image encryption scheme using LICC hyperchaotic system[J]. IET Image Processing, 2022, 16(12): 3342-3358.
30. J. -W. Lee et al., "Privacy-Preserving Machine Learning With Fully Homomorphic Encryption for Deep Neural Network," in IEEE Access, vol. 10, pp. 30039-30054, 2022, doi: 10.1109/ACCESS.2022.3159694.
31. Sideris A, Sanida T, Dasygenis M. A Novel Hardware Architecture for Enhancing the Keccak Hash Function in FPGA Devices[J]. Information, 2023, 14(9): 475.
32. Wu, B., Zhang, T., Mao, L.: Large-scale Image Classification with Multi-perspective Deep Transfer Learning. Computer Science and Information Systems, Vol. 20, No. 2, 743-763. (2023), https://doi.org/10.2298/CSIS220714015W.
33. Zhang G, Xu Y, Rauf M, et al. Breaking the Limitation of Elevated Coulomb Interaction in Crystalline Carbon Nitride for Visible and Near-Infrared Light Photoactivity[J]. Advanced Science, 2022, 9(21): 2201677.
34. Lu L, Ren W X, Wang S D. Fractional Fourier transform: Time-frequency representation and structural instantaneous frequency identification[J]. Mechanical Systems and Signal Processing, 2022, 178: 109305.
35. Yan, Z., Hongle, D., Lin, Z., Jianhua, Z.: Personalization Exercise Recommendation Framework based on Knowledge Concept Graph. Computer Science and Information Systems, Vol. 20, No. 2, 857-878. (2023), https://doi.org/10.2298/CSIS220706024Y.
36. Yu F, Kong X, Chen H, et al. A 6D fractional-order memristive Hopfield neural network and its application in image encryption[J]. Frontiers in Physics, 2022, 10: 847385.
37. Kaur G, Agarwal R, Patidar V. Color image encryption scheme based on fractional Hartley transform and chaotic substitution-permutation[J]. The Visual Computer, 2022, 38(3): 1027-1050.
38. A. Sambas et al., "A Novel 3D Chaotic System With Line Equilibrium: Multistability, Integral Sliding Mode Control, Electronic Circuit, FPGA Implementation and Its Image Encryption," in IEEE Access, vol. 10, pp. 68057-68074, 2022, doi: 10.1109/ACCESS.2022.3181424.

Xi Wang (1980-), is with the School of Artificial Intelligence and Software Engineering, Nanyang Normal University. Research direction is computer application and AI.

Received: March 14, 2024; Accepted: June 29, 2024.

Computer Science and Information Systems 21(4):1699–1723 https://doi.org/10.2298/CSIS240314054Z

Multi-frame Network Feature Fusion Model and Self-attention Mechanism for Vehicle Lane Line Detection

Guang Zhu[1], Yajuan Liu[2], and Jiyue Wang[1,3,⋆]

[1] School of Vehicle and Traffic Engineering, Zhengzhou University of Science and Technology
Zhengzhou 450064 China
zhuguangzz@163.com
[2] School of Foreign Languages, Zhengzhou University of Science and Technology
Zhengzhou 450064 China
xdwangxd@163.com

Abstract. The traditional lane detection networks mainly use independent single frame images to extract features first and then detect them, which cannot deal with the scene with complex background well. Therefore, this paper proposes a lane parallel detection network based on multi-frame network feature fusion model and self-attention mechanism according to the scene characteristics that vehicles can obtain continuous images during normal driving. Firstly, a parallel feature extraction structure is designed. On the one hand, a single frame network with high precision is used to extract the features of the current frame. On the other hand, a lightweight multi-frame network is designed to extract features of low-resolution multi-frame temporal images. And the recurrent neural network module is used to fuse the extracted temporal features and obtain multi-frame features. Self-attention mechanism can effectively capture the relevant information of internal features. Then the fusion module of single frame feature, multi-frame feature and self-attention feature is designed. The feature map of lane line is output by up-sampling network. The experimental results show that the network in this paper has significant improvement in both objective detection accuracy and subjective effect compared with other methods.

Keywords: vehicle lane detection, multi-frame network feature fusion, self-attention mechanism.

1. Introduction

With the continuous breakthrough of artificial intelligence theory and technology represented by deep neural network, autonomous driving and advanced assisted driving have entered a period of rapid development. In practical application, it can not only effectively reduce driving accidents caused by human factors, protect people's life safety, but also greatly improve traffic efficiency, which is the inevitable development trend of intelligent transportation. Lane detection, as an important part of fully automatic driving and advanced assisted driving, is of great significance to the vehicle's environmental awareness ability and lane keeping system [1-3]. At present, although there are some commercial lane detection applications, their application scenarios are limited and cannot be applied

⋆ Corresponding Author

to complex scenarios such as short-time lane occlusion and road surface light and dark changes. Therefore, lane detection in complex scenarios is an important research direction in the field of automatic driving [4].

Due to the excellent performance of neural networks in image segmentation, scholars have proposed various lane detection networks. At present, most lane detection networks are mainly developed from semantic segmentation networks based on single frame images. Li et al. [5] proposed the classical semantic segmentation network (SegNet), which was a symmetric network structure. It recorded the position index of the pooled value in the feature map through a specially designed pooling index, and directly assigned the value to the corresponding position during up-sampling, so as to recover the image segmentation information. Liu et al. [6] proposed U-shaped network (U-NET), which was combined with deconvolution and lateral connection, to make the up-sampling operation learnable. Meanwhile, the up-sampling network can restore details by integrating high-level features to make the prediction result smoother.

In addition to traditional semantic segmentation networks, existing networks are also designed according to lane features. Ko et al. [7] proposed the end-to-end Lane Detection Network. LaneNet took advantage of the feature that lane lines had multi-lane instances, considered lane line detection as an instance segmentation problem, added an additional instance segmentation network branch, and divided all lane line pixel regions into different lane line instances. Lee et al. [8] proposed that based on the vanishing point principle and the end-to-end Vanishing Point Guided Network (VPGNet), lane lines and road markers could be processed at the same time, and the global information provided by vanishing points could be used to improve the detection accuracy in complex weather environments. Mei et al. [9] proposed Spatial Convolutional Neural Network for lane structure (SCNN), modified the original convolution model into a slice convolution suitable for predicting the strip structure, so that the information flow between row and column pixels could be transmitted in the convolution network, thus increasing the weight of lane structure in the network. Garnett et al. [10] proposed the end-to-end 3D Multiple Lane Detection Network (3D-LANenet). The front view and the transformed top view provided by the vehicle multi-camera system were used for lane line prediction. The forward stream processed and retained image information. The head-view flow provided translation invariance characteristics and output the final 3D lane line information. Although the above network can effectively detect the lane lines, it only uses the current moment image to detect the lane lines, ignoring the lane line features in the past time series, under some sudden lane occlusion, Angle of view transformation, and the change of light and shade on the ground. Due to the lack of time domain context information, the detection accuracy is not high.

At present, there are few researches on using video stream as network input to realize lane detection. Zhang et al. [11] proposed the method of combining CNN and Recurrent Neural Network (RNN) for lane line detection, which had better performance than that using only CNN. Although the network introduced the RNN to improve the detection and segmentation effect. However, when the same network structure is used for feature extraction for each frame in the temporal sequence, the network will weaken the most critical feature information of the current frame, and the computing resources will be consumed in the multi-frame features with a large amount of redundancy. In addition,

once the network structure is modified, the global features will be greatly affected, which is not conducive to the optimization of network structure.

In order to make up for the lack of time domain information in lane detection network based on single ton image, the detection speed and accuracy are taken into account. This paper proposes a parallel lane detection network based on image sequence. On the basis of the single-ton image segmentation network, a multi-frame network based on RNN is added, which takes the multi-frame images in the past, including the current frame, as the input of the network. The Convolutional long-short Term Memory (ConvLSTM) gate mechanism [12] is used to fuse and extract the target feature information and environment context information in multi-frame images according to the time series.

In order to reduce the new computational burden caused by multi-frame network, the low-resolution input strategy is adopted, and the lightweight network is selected as the skeleton network of multi-frame network, which can effectively reduce the number of parameters and computational complexity of multi-frame network. At the same time, self-attention mechanism is adopted to extract context features. In order to effectively fuse the multi-frame time-domain features extracted by multi-frame network and the global semantic features extracted by single frame network, a feature fusion module with channel connection is designed. So that the single frame feature can effectively fuse all the multi-frame time domain information, so that the fusion feature has the ability to represent both the spatial information and the time domain information. Finally, the up-sampling network is used to output the lane line feature map.

The paper is organized as follows. Section 2 introduces the related works for this paper. In section 3, we introduce the proposed lane parallel detection network. Data enhancement and model training strategy are shown in section 4. Experiments are displayed in section 5. There is a conclusion in section 6.

2. Related Works

From the perspective of how to define lane detection tasks, lane detection methods based on deep learning can be divided into four categories:

1. Category-based methods [13], which usually combine prior information to determine lane location. For example, DeepLane2 overcomes the shortcoming that the position of lane lines cannot be directly obtained by image classification by adding some priori knowledge related to location into the network [14], and directly estimates the position of lane lines by using deep neural networks.
2. The lane detection method based on classification has a simple network structure, but limited application scenarios, and high computational complexity of post-processing. The method based on object detection uses coordinate regression to detect the boundary box position of the lane line. For example, VPGNet1 uses an improved vanishing point location method to propose an end-to-end multi-task network [15], which can identify lane lines in rainy and low-lighting weather conditions, but requires complex post-processing operations such as point sampling, clustering and lane line regression. Reference [16] proposed a lane detection method based on spatiotemporal deep learning, which used time and space constraints to narrow the scope of the search

area and could accurately detect the lane boundary. However, its data flow and network junction architecture were complex, and the preprocessing could not provide effective results when the initial assumptions were not satisfied.

3. Model-based approach. It uses mathematical model to model the lane lines, and fits the lane lines by solving the model parameters. In reference [17], segmented straight line model was used to fit curved lane lines, heuristic search algorithm was used to search lane boundaries in ROI, and the detected lane lines were fitted as continuous smooth curves, which could well approximate curves. However, this method was only applicable to lane lines with small curvature, and the detection effect of lane lines with large curvature was not accurate enough. In reference [18], a hyperbola model was used to fit the lane lines, so that the two hyperbola converged to the same point, which effectively overcomed the problem of discontinuity of the linear and parabolic models at the junction between the straight and the curve. However, its detection effect was poor in bad weather or when there was interference from other vehicles. Each mathematical model has its own advantages and disadvantages, and it is usually not possible to perfectly fit the lane lines through only one model.
4. Method based on image segmentation. It marks the lane lines and background pixels into different classes, and the detection results are in the form of pixel-level classification. Compared with traditional methods, the detection effect is better and the robustness is better through end-to-end learning. Reference [19] proposed an improved segmentation network (SegNet) algorithm, which used convolution and pooling to extract lane features, and used connected domain constraints to achieve accurate segmentation of lane lines in binary images, but its structure was complex and its real-time performance was poor.

LaneNet4 is a lane detection method based on instance segmentation, with two decoders, one for segmentation and one for embedding,LaneNet can cluster each lane in the training phase rather than the post-processing phase. CNN-LSTM [20] is a hybrid depth network structure that combines CNN and RNN. Two layers of LSTM are added between encoder and decoder, and time domain information is integrated. This method can also achieve better performance under vehicle occlusion, shadow and bad weather conditions. However, the combination of CNN and LSTM requires a large amount of computation. The relevant MMA-Net algorithm in this paper belongs to the method based on image segmentation, which uses 4-layer CNN as the encoder to extract the local and global memory features of the lane lines. The Local and GlobalMemory Aggregation Module (LGMA) is used to aggregate the local and global memory features of multiple layers to enhance the features of the target frame. Finally, the results of lane segmentation are predicted by the decoder. Aiming at the problems of low accuracy and high omission rate of MMA-Net algorithm, this paper proposes a lane parallel detection network based on multi-frame network feature fusion model and self-attention mechanism.

3. Proposed Lane Parallel Detection Network

The structure of lane parallel detection network based on image sequence in this paper is shown in Figure 1, which is mainly composed of two network modules: multi-frame network and single-frame network. Multi-frame network is used to extract time-domain fea-

tures from multi-frame temporal images. The single frame network is based on the coding-decoding model, which is used to extract the global semantic features of the current moment image, and output the final result through the fusion module and the up-sampling module. Because of the parallel structure design, single-frame network and multi-frame network are endowed with different information flows, so that multi-frame network can learn multi-frame time domain characteristics. Single frame network can learn the spatial semantic features of a single frame image, connect these features, and express the lane information more comprehensively. In addition, the relatively independent parallel network structure is also convenient for structural tuning.

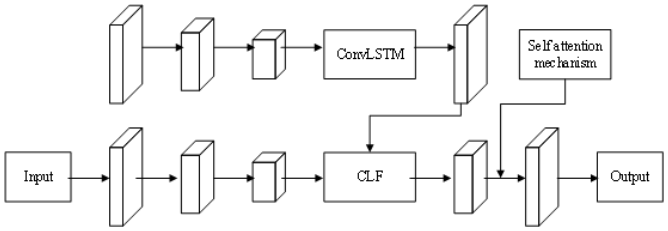


Fig. 1. Proposed network

The multi-frame network in this paper uses a lightweight skeleton network, which takes the multi-frame images in the past, including the current frame, as the network input. Since the introduction of multi-frame network will bring great computational complexity, the low resolution input strategy is adopted. By reducing the input resolution of the multi-frame image, the network computation load is obviously reduced and the network computation time is shortened. After extracting the features of each image output by skeleton network in different time domain, how to effectively fuse the features extracted by each image becomes the main problem. In this paper, the ConvLSTM network module is used to process these features, and the Long short-term memory network (LSTM) [21,22] is used to process the key target feature information in the multi-frame feature sequence while forgetting the unimportant feature information, so as to obtain the final multi-frame time-domain feature.

The single-frame network in this paper is more complex than the multi-frame network. This is because in reality, compared with the image frames at other moments in the past, the image frames at the current moment contain more accurate semantic information, and high-complexity network structures often have stronger generalization learning ability and representation ability for semantic information [23]. Therefore, single-frame network adopts more complex network structure and deeper network level to learn higher level abstract features, and integrates feature information of different sizes through pyramid module structure to obtain single-frame features.

A ConvLSTM fusion module (CLF) is designed to fuse the feature information of single-frame network and multi-frame network. The single frame feature can be fused with the complete multi-frame feature in time domain by means of channel connection. Thus, it makes up for the lack of single-frame features in time domain information, and single-frame features occupy more feature dimensions in the fused features. Ensure that

the single frame feature can be used as the dominant feature in the fusion feature, and the fusion feature can output the final fusion feature map through the up-sampling network.

3.1. Multiple Frame Network

In order to reduce the computational burden of multi-frame network due to the processing of continuous multi-frame images, this paper adopts multi-resolution strategy, which takes low-resolution continuous images as the input of multi-frame network. The high resolution single frame image is used as the input of the single frame network. The following describes the influence of the image input resolution on the network computation in neural networks when the resolution increases gradually. The classic Residual Network with 50 Layers (ResNet50) [24] was used as the test object, and three conventional resolution images were selected as the input. The image input resolution is linearly increased by multiples based on 320x180. The influence of image input resolution on network computation amount and frame number is shown in Table 1.

Table 1. Influence of image input resolution on network computation amount and frame number

Resolution	Network computation amount	Frame/s
320×180	392.3	151.1
480×270	758.6	123.6
640×360	1336.7	91.9

As can be seen from Table 1, the increase of image input resolution is not equivalent to the linear growth of network computation amount, but increases in accordance with the exponential trend. At the same time, it is found that the more the number of channels in the network layer, the greater the increase of the computation. Therefore, by reducing the resolution of the image input, the computation amount of the network can be effectively reduced, and the saved computation amount can be exchanged for the application of more complex and deeper network structure, so as to make up for the loss of image details caused by the resolution. When focusing on semantic segmentation problems such as lane lines, reducing the resolution can also dilute some unimportant background details and highlight the difference between lane lines and background. Therefore, it is reasonable and desirable to reduce the resolution to some extent.

In this paper, a multi-frame network is parallel to a single-frame network. In order to reduce the computational burden caused by the introduction of multi-frame network and achieve better network performance, the second edition of efficient spatial pyramid is selected after analyzing the existing lightweight neural networks network (ESPNetV2) serves as a skeleton network for multi-frame networks. Compared with Image Cascade Network (ICNet) [25], Efficient Residual Factorized Convolutional Network (ERFNet) [26] and the Second Edition of Mobile Vision Convolutional Network [27], the number of floating point operations per second of ESPNetV2 decreases to $[1/9-1/12]$, and the accuracy decreased by only 2%-4%. ESPNetV2 network further optimizes the convolution mode of ESPNet, reduces the number of training parameters of the network by point-group convolution and void depth-separable convolution , while maintaining the original

network structure similar to spatial pyramid [28], and proposes the highly efficient spatial pyramid module (EESP).

The EESP module is mainly based on the block convolution principle and spatial pyramid theory. Firstly, the dimension of the input features is reduced by Group Convolution (GConV), and then the Depthwise Separable Convolution (DSConV) operation is performed by using Convolution of different scales to verify low-dimensional features. Finally, the features are concatenated by element summation. The specific structure of the module is shown in Figure 2.

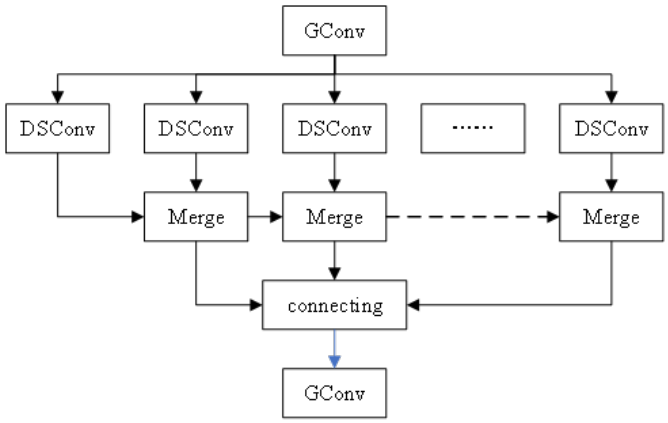


Fig. 2. Structure of EESP module

The EESP module replaces conventional convolution by grouping convolution and depth-separable convolution respectively, which can effectively reduce the computational cost of convolution operation. The features are convolved by expanding convolution kernels with different scales in Hierarchical Feature Fusion (HFF) module. The local feature information and global feature information under different receptive fields are fused by summing element by element. The information receiving field of the whole network is expanded, so as to effectively improve the detection effect. At the same time, the overall module network structure of EESP is similar to the spatial pyramid structure, so that the network has stronger semantic information and enriches the receptive field.

The ESPNetV2 skeleton network is constructed based on the EESP network module. After initial convolution, the skeleton network structure is divided into four spatial levels. Each space level employs one or more EESP network modules. After analyzing the original network structure, this paper finds that the original skeleton structure applies multiple EESP modules to the same scale in Part 2 and Part 3, and a large number of repeated convolution operations will repeatedly extract feature information, resulting in certain information redundancy. Therefore, in this paper, the number of EESP modules in part 2 and part 3 of the skeleton network is further compressed, and finally two and four EESP modules are adopted, respectively, to further reduce the computational burden of multi-frame network. The skeleton network structure of the final multi-frame network is

shown in Table 2. Each temporal image will pass through the skeleton network to extract features and obtain multi-frame feature sequences.

Table 2. Backbone network structure of multi-frame network

Layer	Number	Output size
Convolution	1	112×112
Strided EESP	1	56×56
Strided EESP	1	28×28
EESP	2	28×28
Strided EESP	1	14×14
EESP	4	14×14
Strided EESP	1	7×7
EESP	3	7×7

When feature sequences are obtained through multi-frame network, how to extract the target feature information that changes in time domain from feature sequences. To effectively ignore the background features that do not change much, that is, how to abstract features selectively is very important for segmentation target. Therefore, this paper studies the characteristics of RNN and introduces LSTM network module to obtain target feature information selectively. As a special form of RNN, LSTM uses three different gate functions to extract long-term features, which are the input gate that controls the addition of new information, the forgetting gate that controls the passage of information, and the output gate that determines the output of information.

In this paper, the gate mechanism of LSTM is used to process multi-frame feature sequence, and the target feature information in multi-frame feature sequence is extracted according to the time domain. At the same time, the unimportant feature information is forgotten. However, LSTM is a fully connected network model, which will lead to extra time consuming. In this paper, the ConvLSTM structure is adopted, and the convolution operation is used to replace the matrix multiplication operation in each gate function, so as to capture the potential spatio-temporal features in the temporal features. Convolution is also more effective and easy to understand the image feature extraction, effectively improve the image feature processing speed and adaptability to feature maps. A regular ConvLSTM memory unit is computed at time t as follows.

$$i_t = \sigma(W_{xi} \times X_t + W_{hi} \times H_{t-1} + W_{co} \times C_{t-1} + b_i). \tag{1}$$

$$f_t = \sigma(W_{xf} \times X_t + W_{hf} \times H_{t-1} + W_{cf} \times C_{t-1} + b_f). \tag{2}$$

$$C_t = f_t \times C_{t-1} + i_t \tanh(W_{xc} \times X_t + W_{hc} \times H_{t-1} + b_c). \tag{3}$$

$$o_t = \sigma(W_{xo} \times X_t + W_{ho} \times H_{t-1} + W_{co} \times C_{t-1} + b_o). \tag{4}$$

$$H_t = o_t \times \tanh(C_t). \tag{5}$$

Where, i_t represents the exit entry at time t. f_t represents the forgetting gate at time t. C_t represents the ConvLSTM cell state at time t. o_t represents the output gate at time t. represents the state of the hidden layer at time t. σ stands for sigmoid function. $tanh$ is the hyperbolic tangent function. $\times$ denotes the convolution operation. X_t represents the input feature at time t. W_{xi} represents the weight matrix of input X_t under the input gate. b_i represents the offset value under the input gate. In general, C_t changes slowly. H_t varies a lot from node to node.

3.2. Single Frame Network

Since the single frame feature at the current moment needs to be the dominant feature, the skeleton network of the single frame network needs to use a more complex network structure to improve the generalization learning ability and representation ability of features. In this paper, classical neural networks such as ResNet, VGG16-BN and GoogleNet are selected as the skeleton network of single-frame network, and the fully connected layer in the network structure is removed.

The number of channels in the output feature map of different skeleton networks is different. In order to keep the output feature map of the single-frame network and the multi-frame network have the same scale and equal ratio of channels, and strengthen the feature information of the single frame feature at different scales, the multi-scale feature enhancement structure [29] is added after the single-frame skeleton network, and the feature information is enhanced at three different scales. After using the enhanced structure, each spatial location can view the local environment in different scale space, further expand the information receiving domain of the whole network, and improve the effect of lane detection.

The ConvLSTM fusion module (CLF) is designed to effectively fuse the single frame features and multi-frame features extracted through parallel network. As shown in Figure 3, CLF first up-samples the multi-frame feature map to restore the feature map size to the same size as the single-frame feature. Then 1x1 convolution kernel is used for single-frame features and multi-frame features to smooth features. Then, multi-frame features and single-frame features are fused by channel connection, in which single-frame features occupy more feature dimensions in the fused features and serve as the dominant features. Finally, the fused features were activated at the site to the reasonable Satisfaction Linear Unit (ReLU) to reduce the interdependence between parameters.

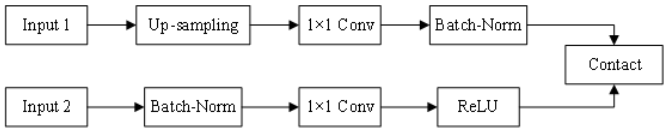


Fig. 3. CLF structure

3.3. Self Attention Mechanism

Self-attention mechanism can effectively capture the relevant information of internal features. The self-attention mechanism is used to automatically learn the context information of the users in the group and the influence of other users, which can solve the problem of dynamic complexity of the members in the group. The structure of the self-attention mechanism is shown in Figure 4. For the analysis of one module, c_1 represents the context information vector representation of the first user u_1. $u_2, \cdots, u_n$ stands for other user vector representation. According to the context information of the user and the embedding vector of the user into a matrix $U_1 = [c_1, u_2, u_3, \cdots, u_n]$, $U_1 \in R_d$. d represents the embedding vector dimension of the user, and other modules learn in the same way. The input to the self-attention mechanism used in this article consists of query, key, and value. The input of query, key and value is composed of the matrix U_1 formed by embedding vectors of user and context information, and the three are equal, and the output is the final weighted $A_{u,i}$. The specific calculation steps are as follows:

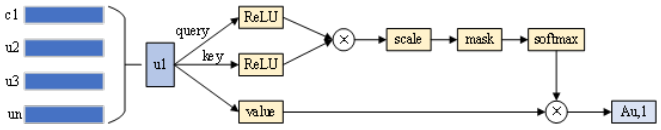


Fig. 4. Structure of self attention mechanism

Project the input query and key into the same space and introduce nonlinear functions to learn Q and K.

$$Q = \sigma(U_1 \cdot W_a). \tag{6}$$

$$K = \sigma(U_1 \cdot W_k). \tag{7}$$

Where σ is the Relu activation function. $W_a = W_k$ is the weight matrix of Query and key respectively, $W_a = W_k \in R_d \times d$.

This paper uses the compressed inner product attention function. The results in (1) are multiplied to get the corresponding weight matrix B_1, and then normalized.

$$B_1 = \sigma(\frac{Q \cdot K^T}{\sqrt{d}}). \tag{8}$$

Where σ is the softmax function. The scale operation uses $\sqrt{d}$ to expand and shrink attention to desaturate the function value to avoid too low gradient. Secondly, through mask operation, the diagonal elements of the matrix are hidden to avoid high matching scores between the same vectors of Query and key.

The weight matrix B_1 can be obtained from Equation (8), and the group matrix represents $A_{u,1}$ by multiplying the weight moment B_1 and value obtained. In this method, the user context information and user information are concatenated into a matrix U_1 and

the final weighted output is learned by the self-attention mechanism to obtain the group representation $A_{u,1}$.

$$A_{u,1} = B_1 \cdot U_1. \tag{9}$$

Where U_1 is the matrix spliced by the context information vector of u_1 and other member user vectors in the group. B_1 is the weight matrix obtained by compressing the inner product attention function. The above is the first module that integrates contextual information to obtain group representation $A_{u,1}$ through self-attention mechanism learning. Other modules learn in the same way, that is, n groups can be obtained, representing $A_{u,2}, \cdots, A_{u,n}$.

3.4. Model Optimization

Bayesian personalized ranking (BPR) establishes a triplet model for users and items. Assuming the user views the positive items, the user must prefer the items he sees to the negative items he does not see. Therefore, the sorted items of positive items are better than the negative items. The BPR optimization objective is to perform the optimal personalized ranking based on the maximum a posteriori estimator, and its formula is as follows:

$$L = argmin \sum_{(g,j,k) \in D_S} - \ln \sigma(X_{u,a}(\theta) - X_{u,b}(\theta) + \lambda ||\theta||^2). \tag{10}$$

Where (g, j, k) belongs to the triplet of set D_S and contains all pairs of positive and negative items of each user. $X_{u,a}$ represents the predicted score of user u for Project a. $X_{u,b}$ represents user's prediction score for project b. σ is the sigmoid function. λ is the regularization parameter. is the model parameter.

This paper takes BPR as the basis of model learning, treats a group as a user, uses BPR algorithm to make recommendations, and then uses matrix factorization method. Therefore, the objective function of this paper is:

$$L = arg \min_{\theta} \sum_{(g,j,k) \in D_S} - \ln \sigma(g_1 \cdot v_a - g_1 \cdot v_b + \lambda ||U||^2 + \lambda ||C||^2). \tag{11}$$

Where g_1 represents the group representation of the mean embedding. U and C are parameters which represent the set of user latent vector and user context vector, respectively. $g_1 \cdot v_a$ represents the group's predicted score on a. $g_1 \cdot v_b$ represents the group's predicted score on b. (g, j, k) belongs to the triplet of set D_S, including all pairs of positive and negative items of each group.

3.5. Up-sampling Network Structure

As a decoding network, the up-sampling network uses the underlying features to recover the target information by up-sampling. The up-sampling network in this paper adopts the same decoding structure as LaneNet. LaneNet consists of four convolutional layers and one up-sampling layer. The up-sampling layer smooths the sampled feature map through the up-sampling algorithm of bilinear interpolation. Finally, the fused feature map is enlarged to the same size as the input feature map, and this feature map was output as the result feature map.

4. Data Enhancement and Model Training Strategy

4.1. Data Enhancement

In this paper, network training is based on TuSimple road data set [30], and the data set is enhanced accordingly. The TuSimple road dataset contains a total of 6570 groups of lane line image sequences. Each sequence contains 20 consecutive frames of images acquired within 1s. All images are based on the foreground view of the car. The original resolution of each frame is 1280×720. For each set of image sequences, only the last 20 frame image marks the true lane line label. In TuSimple dataset, 3626 sets of image sequences are used as the training set and 2944 sets of image sequences are used as the test set, including different time periods during the day, different number of lanes and various traffic conditions. To a certain extent, it is in line with the realistic road situation.

In order to avoid over-fitting and improve the performance of the dataset, data enhancement was performed on the dataset. According to the network structure in this paper, for each group of image sequence, 4 images are sampled from the 20 consecutive frames at different intervals, and a group of training data is composed by combining the label images. The road images collected by the camera at different speeds can be simulated by acquisition at different intervals. In this paper, 1, 2, 3, 4 and 5 frames are respectively used as sampling benchmarks, and this standard is applied to each group of image sequences. The sampling methods of continuous ton images are shown in Table 3.

Table 3. Sampling of continuous frame image

Sampling interval	The sampling frame
1	17,18,19,20
2	14,16,18,20
3	11,14,17,20
4	8,12,16,20
5	5,10,15,20

After the above processing, 5 times of the original data set image sequence data is obtained. In order to further improve the diversity of training data, image horizontal flip is used for data enhancement. Aiming at the ratio of 0.55:0.45 in the training set and test set on the original data set, the more commonly used ratio of 0.8:0.2 is used to construct the training set and test set. Finally, a training set including 51260 groups of image sequences and a test set containing 12820 groups of image sequences were constructed.

In order to test whether the proposed network can effectively cope with complex scenes such as lane occlusion and light shadow, 540 groups of complex image sequences were further selected from the test set as the complex test set for subsequent experiments, including 435 groups of lane occlusion image sequences and 105 groups of light shadow image sequences. In addition, in order to improve the training speed, 640×360 is finally selected as the basic resolution.

4.2. Training Strategy

After the parallel network structure is established in this paper, the design of reasonable training strategy is also a key part of neural network training. In order to effectively extract the features of the lane model through continuous update and learning of network parameters, gradually reduce the error value with the real label, and finally achieve the effect of separating the lane line from the background, this paper formulated the training strategy as follows.

1. Consider the initialization of network parameters. The methods of network parameter initialization mainly include network pre-training parameter initialization, random parameter initialization and fixed value parameter initialization. Because the network in this paper adopts parallel network structure design, the initialization parameter strategy will be different. For the skeleton network of a single frame network, the pre-trained parameters of the network model on the ImageNet dataset are used for parameter initialization. For the convolutional layer in the multi-frame network, the Kaiming normal distribution initialization algorithm is applied to initialize the parameters of the convolutional layer. For the Batch Normalization (BN) layer in the multi-frame network structure, according to the initialization method in reference [31], the weight value and bias value are filled with fixed 1 and fixed 0, respectively
2. In this paper, the network adopts Stochastic Gradient Descent (SGD) as the optimization algorithm, with the initial learning rate=0.03, weight $decay = 5e^{-5}$, and momentum parameter=0.9. The learning rate strategy selects Poly as the way to adjust the learning rate. The logarithmic function makes the learning rate decay according to the number of cycles, which avoids the problem that the learning rate is too large to converge in the later stage of neural network training.
3. Weighted cross-entropy is used as the loss function of the network in this paper, which has considerable effect on image classification and segmentation. According to the idea of classification, the cross drop loss function first makes class prediction for each pixel of the segmented image, then calculates the loss value of a single pixel by comparing the prediction results with the label pixels, and finally calculates the average value of the total loss value of all pixels. Therefore, each pixel can be learned by the network and continuously improve the output result, but such loss function can not be directly used to deal with the lane segmentation problem. This is because in the actual image, the background pixels are far more than the lane pixels, which leads to the category imbalance in the learning process, making the final loss value dominated by the background category, and the lane features are difficult to be learned by the network. With different categories in order to solve this problem, can be obviously improved by means of weighted study effect, loss of background value multiplied on a smaller weight value, can significantly reduce background for the effects of loss value, otherwise, the target class on a large weight value, increase the probability of the target class are studied, the loss function are defined as follows:

$$Loss = \sum_{x \in \Omega} w(x) \ln(p_{l(x)}(x)). \tag{12}$$

Where x represents the target category of segmentation. $w(x)$ represents the weighted value of the current category. $p_{l(x)}$ represents the probability that the network fits the

predicted value l to the true value x which is calculated by KL divergence. If the predicted value is close to the true value, the loss value is small. Otherwise, the loss is greater. In this paper, the W-weighted value of the background class is set as 0.02, and the W-weighted value of the lane line is set as 1.02.

5. Experiments and Result Analysis

The network in this paper is implemented based on Pytorch.1 and runs on 64-bit Ubuntu 18.04 system. In the computer hardware configuration: the processor is Intel 9700K CPU4.50 GHz; GPU is GeForceGTX-1080TI; The video memory is 11GB.

5.1. Evaluation Indexes

In order to comprehensively evaluate the network performance in this paper, the evaluation indexes used in the experiment are as follows.

Accuracy. Measurement of the proportion of the number of correctly classified pixels in the total number of pixels:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}. \tag{13}$$

Where TP represents the number of pixels correctly predicted as road labels and it is defined as the true positive sample. TN denotes the number of pixels correctly predicted as road labels defined as the true negative sample. FN denotes the number of pixels that incorrectly predict a road label as a background label defined as a false negative sample. FP represents the number of pixels that incorrectly predict the background label to be a road label defined as a false positive sample.

The accuracy rate cannot accurately describe the lane detection accuracy in the actual situation. This is because in ordinary road environment, most pixels are background pixels rather than lane pixels. The extreme imbalance of this category leads to a high accuracy in the image with fewer pixels to predict lane lines, even if all of them are predicted as background labels. Therefore, in this paper, F1 is used to measure the comprehensive performance of the network model for the single category of lane. As a model accuracy index, the F1-measure evaluates the comprehensive performance of the network through the weighted average of Precision and Recall. The detailed calculation is as follows:

$$F1 = \frac{2P \cdot R}{P + R}. \tag{14}$$

$$P = \frac{TP}{TP + FP}. \tag{15}$$

$$R = \frac{TP}{TP + FN}. \tag{16}$$

5.2. Skeleton Network Selection

In order to further verify that the network in this paper still plays a role when other single-frame skeleton networks are used, VGG16-BN, GoogleNet, ResNet32 and ResNet50 are selected as the skeleton networks of the single-frame network, and comparative experiments are carried out. The experimental results are shown in Table 4. It can be seen from the table that different skeleton networks have a certain degree of influence on the detection effect. The index values of ResNet50 are the highest, followed by VGG16 and ResNet32, and GoogleNet is the lowest.

Table 4. Experimental result comparison of different backbone networks

Backbone network	Accuracy	P	R	F1	Frame/s
GoogleNet	0.970	0.891	0.913	0.901	33.2
ResNet32	0.973	0.895	0.928	0.910	28.3
VGG16	0.975	0.905	0.929	0.918	22.6
ResNet50	0.984	0.914	0.948	0.928	19.9

Figure 5 shows the subjective detection results of different skeleton networks. As can be seen from the figure, the subjective detection effect of ResNet50 and VGG16-BN is better, and the edge of ResNet50 is finer. GoogleNet detection effect is poor, the edge is rough, and the lane line is broken.

5.3. Network Structure Test

In this paper, the TuSimple dataset is enhanced by randomly dividing the dataset according to the ratio of 0.8:0.2. But there is a definite chance of randomization. In order to eliminate the contingency, the experiment uses the way of 5 fold cross validation to carry out multiple tests. The original data set was divided into 5 groups, each group of subset data was used as a test set, and the remaining 4 groups of subset data were used as training set. Finally, five model results were obtained. The results of these five models on their respective test sets are calculated and the average value is calculated, which is used as the performance index of the network in this paper. In cross-validation, 640×360 is selected as the input image resolution of single-frame network, and 320×180 is selected as the input image resolution of multi-frame network.

The cross-validation results of the network in this paper are shown in Table 5. The experimental data are relatively stable, and the floating range of all parameters is within 0.005, indicating the stability of the network in this paper.

In order to test the impact of the proposed multi-frame network on the performance of the proposed single-frame network, the ablation experiment was further used to comprehensively test the network structure, using only the single-ton network, using only the multi-frame network, and using the proposed network, respectively. The test results are shown in Table 6. As can be seen from Table 6, compared with multi-frame network and single-frame network, the proposed network achieves better results in all parameters,

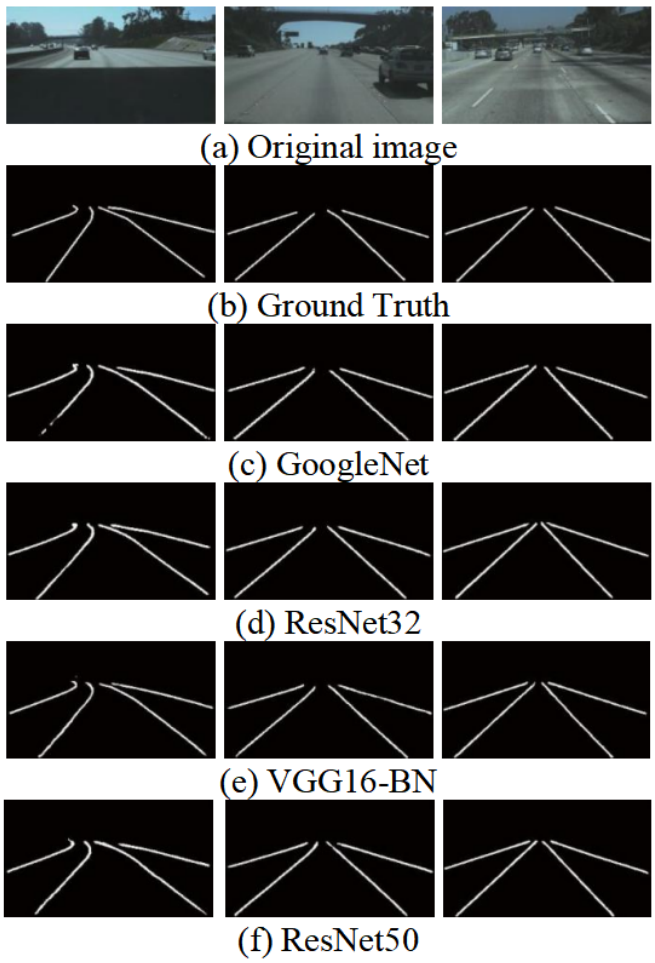


Fig. 5. Subjective results of different backbone networks

Table 5. Cross validation results of the proposed network

Cross validation	Accuracy	P	R	F1
1	0.985	0.919	0.942	0.931
2	0.985	0.921	0.941	0.931
3	0.985	0.918	0.937	0.928
4	0.984	0.913	0.943	0.928
5	0.985	0.916	0.936	0.927
Average	0.985	0.917	0.940	0.929

which objectively verifies the effectiveness of the proposed network structure and indicates that single-ton network can effectively improve network performance after adding multi-ton network.

Table 6. Results of different network structures on test set

Network	Accuracy	P	R	F1	Frame/s
Single frame	0.981	0.885	0.916	0.893	29.1
Multi-frame network	0.976	0.828	0.892	0.857	44.5
Proposed	0.985	0.917	0.942	0.939	22.4

Figure 6 shows the subjective detection results of the three networks under three different scenarios: no lane line is blocked, there is lane line is blocked, and light shadow. As shown in figure 6, in the image with no lane line blocked on the left, the three network structures can effectively detect lane lines. But the details at the edges of this network are smoother. In the image with blocked lane lines in the middle, the multi-frame network and the proposed network can effectively detect the blocked lane lines due to the advantage of image sequence, while the single-frame network has lane line breaks at the occlusion. In the image of light shadow on the right, the lane line is discontinuous in both multi-frame network and single-frame network, and the proposed network can still detect the lane line effectively.

According to the above experimental data, because the multi-frame network adopts lightweight skeleton network, the ability to learn lane features is not strong, resulting in lane discontinuity and poor edge quality. Although the parameters of the single-frame network are better than those of the multi-frame network, it lacks the time domain perception ability of lane features in complex scenes such as lane occlusion and light shadow, and the lane line discontinuity occurs at the occlusion. In this paper, the multi-frame network module is added on the basis of the single-frame network, which not only strengthens the generalization learning ability and representation ability of the single-frame network for complex scenes, but also integrates the time-domain context features given by the multi-frame network, and obtains the blocked lane features according to the RNN features in the case of short-term occlusion.

From the above experimental results, it can be seen that compared with the independent single-frame network and multi-frame network, the proposed network achieves better detection effect in both subjective and objective evaluation.

5.4. Multiple Resolution Selection

A low resolution image input strategy is used in multi-frame networks to reduce network computation. In order to test the influence of different input image resolutions on the performance of the overall network, three different resolutions are selected as the input image resolutions of the multi-frame network while keeping the resolution of the input image of the single-frame network as 640×360. The specific results are shown in Table 7.

As can be seen from Table 7, reducing the input resolution of the multi-frame network will slightly reduce the network detection effect, but effectively reduce the running time

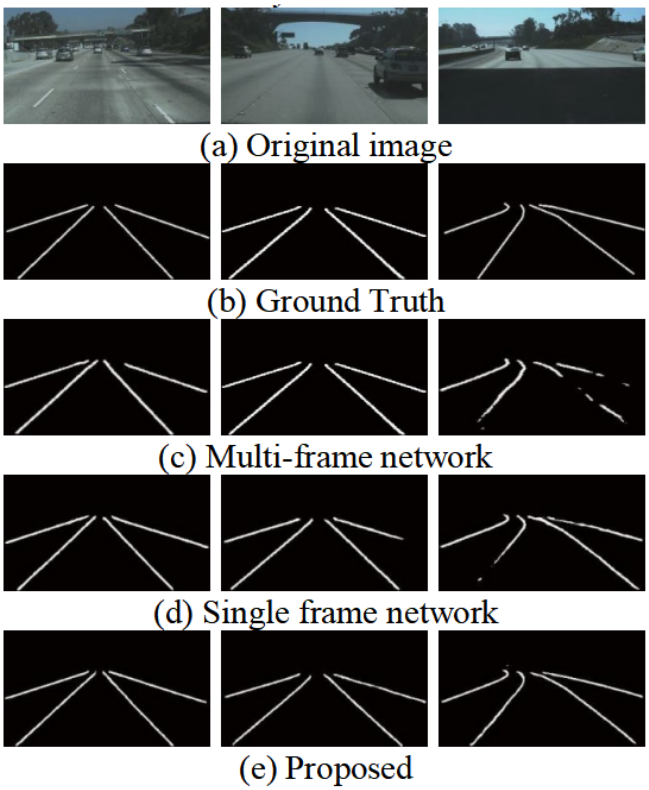


Fig. 6. Subjective results of different network structures

Table 7. Experimental result comparison with different input resolutions

Resolution	Accuracy	P	R	F1	Frame/s
640×360	0.982	0.918	0.941	0.928	21.1
480×270	0.982	0.919	0.938	0.927	22.0
320×180	0.982	0.914	0.937	0.925	22.7

of the multi-frame network and improve the number of network calculations. Reducing the input resolution will make the edge sampling less refined, and the edges of the longer lane lines will be rougher, resulting in a slightly lower overall accuracy.

Figure 7 shows the subjective detection results under different input resolutions. As can be seen from the comparison of the detected images in the figure, the subjective visual gap is not obvious. In view of subjective and objective evaluation, 320×180 is finally selected as the input resolution of multi-frame network.. The experiments in this section also further show that the performance of multi-frame network has a positive influence on the overall parallel network to a certain extent, and optimizing the performance of multi-frame network can also improve the performance of parallel network in this paper.

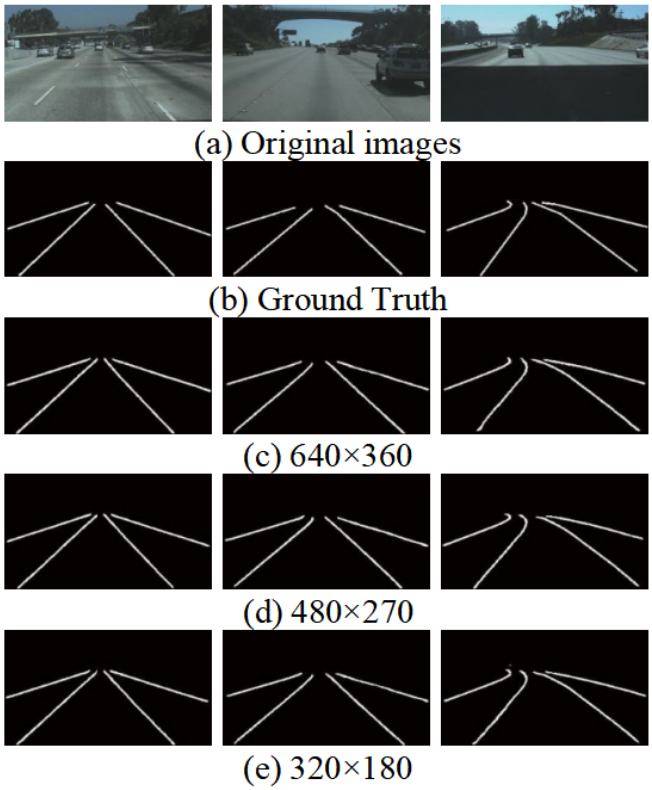


Fig. 7. Subjective results with different input resolutions

5.5. Comparison with Advanced Networks

In order to further evaluate the proposed network, several existing neural networks that also use VGGNet as the skeleton network are selected for performance comparison, including SegNet, UNet, LaneNet, SegNET-CL and UNet-CL. In order to better evaluate the

performance of the proposed network and other neural networks based on single frame detection and multi-frame detection, the performance factors of skeleton network are excluded. The above networks are trained with the same training set, and tested with the same overall test set and complex test set with lane line occlusion respectively. The experimental results are shown in Table 8 and Table 9, respectively.

Table 8. Performance comparison of different networks on TuSimple test set

Network	Accuracy	P	R	F1	Frame/s
LaneNet	0.981	0.875	0.916	0.893	30.1
UNet	0.982	0.888	0.910	0.899	20.0
SegNet	0.982	0.887	0.918	0.901	26.0
SegNet-CL	0.983	0.908	0.916	0.909	25.1
UNet-CL 0.983	0.899	0.927	0.912	19.1	
Proposed	0.985	0.917	0.940	0.928	23.6

Table 9. Performance comparison of different networks on complex test set

Network	Accuracy	P	R	F1
UNet	0.978	0.831	0.853	0.841
UNet-CL	0.980	0.853	0.889	0.865
Proposed	0.982	0.878	0.901	0.887

It can be seen from Table 8 that, compared with the existing neural networks for multiple single-frame detection, the proposed network can improve the accuracy, recall and F1 metric by about 2% when the time complexity is slightly increased.

As can be seen from Table 9, when the difficulty of the dataset increases, the performance index of each network decreases to a certain extent. But the accuracy rate of this network is still high. This further illustrates the advantages of the network in this paper. The reason is that the lane information in the image sequence is relatively continuous and changes little, but the overall background changes are more dynamic and complex. In the face of occlusion, it is difficult to accurately detect lane features in single frame detection. Through multi-frame network, continuous and less variable features such as lane lines have a higher probability to be detected.

Figure. 8 shows the comparison of subjective detection results of each network. Based on the analysis of the actual detection effect, it can be concluded that the network in this paper can still effectively remove the influence of occlusion when the lane line is suddenly blocked in columns 2 to 6 in Figure 8. For complex scenes with varying light and dark scenes such as column 4 and column 6 in Figure 8, the multi-frame features enhance the semantic information of single frame features, so the detection effect is also better.

SegNetConvLSTM and UNet-ConvLSTM, which use the same network structure to extract features for each image, weaken the most critical feature information of the current frame and produce more redundant features. Therefore, excessive redundant features need

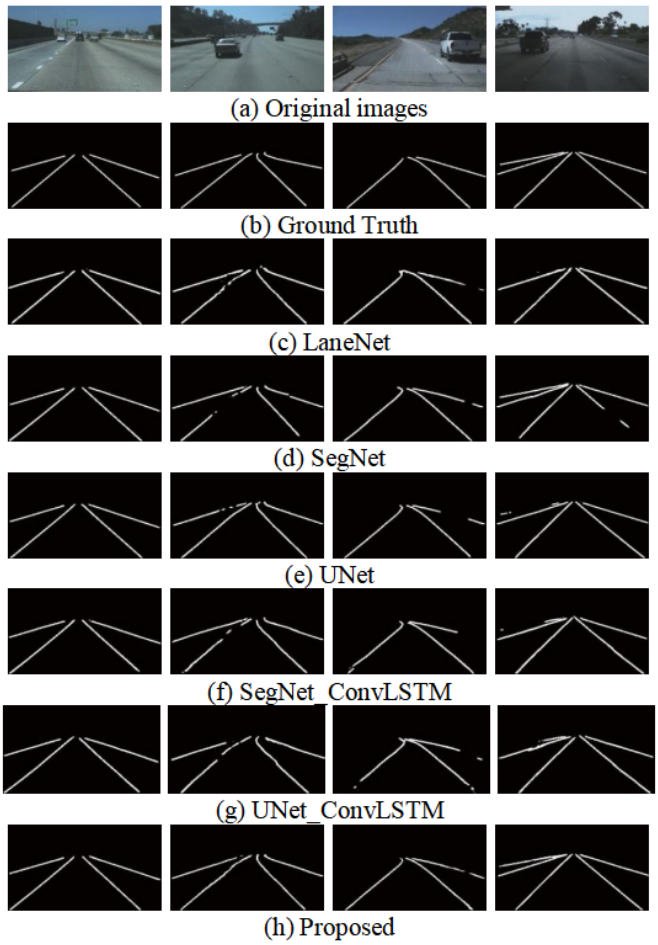


Fig. 8. Subjective result comparison of different networks

to be processed when using RNN to fuse features. This leads to a decrease in accuracy. In this paper, by giving different information flow to single-frame network and multi-frame network, and adopting skeleton network with different complexity, multi-frame network can learn multi-frame time-domain semantic features. Single-frame network learns spatial semantic features of a single frame image, which not only effectively reduces redundant features, but also strengthens the interpretation of the network and extracts lane features more effectively. According to the results in Tables 8 and 9, compared with the lane detection network based on video streaming, the accuracy rate, recall rate and F1 measure of the proposed network on the overall test set are all improved by about 1%. On the complex test set containing only lane line occlusion, the performance improvement of the proposed network is more obvious.

6. Conclusion

This paper proposes a parallel lane detection network based on image sequence. On the basis of the single frame based on the original network parallel rise frame extraction network, using RNN module containing the current frame and the frame more sequences of time-domain characteristics, environmental context feature extraction and fusion, not only effectively make up for the single frame network of temporal context information is missing, also assist improve the lane line semantic information. Since the current frame image is the most important temporal frame, compared with the multi-frame network, the single-frame network adopts the skeleton network with higher complexity to improve the generalization learning performance and representation ability of the current frame features. Experiments show that the introduction of multi-frame network module in the network of this paper has better performance index than using only single-frame network. Due to the low resolution strategy, lightweight network structure and reuse of the feature map of the past frame, the time of adding multi-frame feature module is only increased by about 25%. Although the network in this paper can make up for the missing temporal feature information of the single-frame network to a certain extent, it is still not suitable for the situation that the lane line is blocked for a long time and the road light has many shadows. In the subsequent research, the feature extraction and fusion of multi-frame network and single-frame network and the optimization of long-term occlusion and other complex actual scenes will be further studied, so as to make the spatial feature information of single frame target more complementary to the temporal information of image sequence.

Acknowledgments. This work was supported by Research project of Zhengzhou University of Science and Technology: Vibration Analysis and Control Based on Flexible Joints and Loads of Industrial Robots (2022XJKY05); Key Teaching reform project of Zhengzhou University of Science and Technology, "Research on Cultivating Interdisciplinary Innovative Practice Teaching Based on Robot Technology" (2022JGZD03); 2023 Teacher Development Research Project of Zhengzhou University of Science and Technology, Research on Career Development Dilemma and Implementation Path of Young Teachers in Application-oriented Universities in the New Era (JSFZ-ZXKT2023004); Key research projects of universities in Henan Province, "Development and Research of Intelligent Gauge Detection System of Rail Transit Based on Dynamic Distance Measurement" (24B460027), "Application Research of Rhodes Correlation Failure Model Based on Big Data Acquisition in high-speed Cutting Analysis" (24B460028); "Research on Structure Optimization and vibration Damping Performance of high-frequency needling Equipment for Nonwoven

Fabrics" (23B460011), "Research on Key Technologies of Supercapacitor Module Equalization and Safety Monitoring and early Warning based on Evidence Theory" (23B480003).

The author is grateful for the anonymous review by the review experts.

References

1. A. Dominic Savio, C. Balaji, D. kodandapani, K. Sathyasekar, R. Naryanmoorthi, C. Bharatiraja, and Bhekisipho Twala. DC Microgrid Integrated Electric Vehicle Charging Station Scheduling Optimization [J]. Journal of Applied Science and Engineering. 26(2), 253-260, 2022. https://doi.org/10.6180/jase.202302_26(2).0011
2. Liu S, Liu L, Tang J, et al. Edge computing for autonomous driving: Opportunities and challenges[J]. Proceedings of the IEEE, 2019, 107(8): 1697-1716.
3. Liu L, Lu S, Zhong R, et al. Computing systems for autonomous driving: State of the art and challenges[J]. IEEE Internet of Things Journal, 2020, 8(8): 6469-6486.
4. Tang J, Li S, Liu P. A review of lane detection methods based on deep learning[J]. Pattern Recognition, 2021, 111: 107623.
5. Li Y, Shi T, Zhang Y, et al. Learning deep semantic segmentation network under multiple weakly-supervised constraints for cross-domain remote sensing image semantic segmentation[J]. ISPRS Journal of Photogrammetry and Remote Sensing, 2021, 175: 20-33.
6. Liu L, Cheng J, Quan Q, et al. A survey on U-shaped networks in medical image segmentations[J]. Neurocomputing, 2020, 409: 244-258.
7. Ko Y, Lee Y, Azam S, et al. Key points estimation and point instance segmentation approach for lane detection[J]. IEEE Transactions on Intelligent Transportation Systems, 2021, 23(7): 8949-8958.
8. Lee S, Kim J, Shin Yoon J, et al. Vpgnet: Vanishing point guided network for lane and road marking detection and recognition[C]//Proceedings of the IEEE international conference on computer vision. 2017: 1947-1955.
9. Mei S, Jiang R, Li X, et al. Spatial and spectral joint super-resolution using convolutional neural network[J]. IEEE Transactions on Geoscience and Remote Sensing, 2020, 58(7): 4590-4603.
10. Garnett N, Cohen R, Pe'er T, et al. 3d-lanenet: end-to-end 3d multiple lane detection[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019: 2921-2930.
11. Zhang W, Liu H, Wu X, et al. Lane marking detection and classification with combined deep neural network for driver assistance[J]. Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering, 2019, 233(5): 1259-1268.
12. Sudhakaran S, Lanz O. Learning to detect violent videos using convolutional long short-term memory[C]//2017 14th IEEE international conference on advanced video and signal based surveillance (AVSS). IEEE, 2017: 1-6.
13. Takahashi M, Iino K, Watanabe H, et al. Category-based memory bank design for traffic surveillance in context R-CNN[C]//International Workshop on Advanced Imaging Technology (IWAIT) 2023. SPIE, 2023, 12592: 84-87.
14. Battarra I, Accorsi R, Manzini R, et al. Hybrid model for the design of a deep-lane multisatellite AVS/RS[J]. The International Journal of Advanced Manufacturing Technology, 2022, 121(1-2): 1191-1217.
15. Garg T, Bachu S, Panda D, et al. Multi-Stage Pyramid Parsing Network For Lane Marking Detection[C]//2022 International Conference on INnovations in Intelligent SysTems and Applications (INISTA). IEEE, 2022: 1-6.
16. Dong Y, Patil S, van Arem B, et al. A hybrid spatial-temporal deep learning architecture for lane detection[J]. Computer-Aided Civil and Infrastructure Engineering, 2023, 38(1): 67-86.

17. Modi S, Bhattacharya J, Basak P. Multistep traffic speed prediction: A deep learning based approach using latent space mapping considering spatio-temporal dependencies[J]. Expert Systems with Applications, 2022, 189: 116140.
18. Luo S, Yao J, Hu J, et al. Using deep learning-based defect detection and 3D quantitative assessment for steel deck pavement maintenance[J]. IEEE Transactions on Intelligent Transportation Systems, 2022. doi: 10.1109/TITS.2022.3169164.
19. Ghandorh H, Boulila W, Masood S, et al. Semantic segmentation and edge detection: Approach to road detection in very high resolution satellite images[J]. Remote Sensing, 2022, 14(3): 613.
20. Kortli Y, Gabsi S, Voon L F C L Y, et al. Deep embedded hybrid CNN-LSTM network for lane detection on NVIDIA Jetson Xavier NX[J]. Knowledge-based systems, 2022, 240: 107941.
21. Li P, Laghari A A, Rashid M, et al. A deep multimodal adversarial cycle-consistent network for smart enterprise system[J]. IEEE Transactions on Industrial Informatics, 2022, 19(1): 693-702.
22. Wang L, Yin S, Alyami H, et al. A novel deep learning-based single shot multibox detector model for object detection in optical remote sensing images[J]. Geoscience Data Journal. 2022. https://doi.org/10.1002/gdj3.162.
23. Tsironi E, Barros P, Weber C, et al. An analysis of convolutional long short-term memory recurrent neural networks for gesture recognition[J]. Neurocomputing, 2017, 268: 76-86.
24. Theckedath D, Sedamkar R R. Detecting affect states using VGG16, ResNet50 and SE-ResNet50 networks[J]. SN Computer Science, 2020, 1(2): 1-7.
25. Cui Z, Chang H, Shan S, et al. Deep network cascade for image super-resolution[C]//European Conference on Computer Vision. Springer, Cham, 2014: 49-64.
26. Romera E, Alvarez J M, Bergasa L M, et al. Erfnet: Efficient residual factorized convnet for real-time semantic segmentation[J]. IEEE Transactions on Intelligent Transportation Systems, 2017, 19(1): 263-272.
27. Lee J, Tang H, Park J. Energy efficient canny edge detector for advanced mobile vision applications[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2016, 28(4): 1037-1046.
28. Shoulin Yin, Hang Li, Asif Ali Laghari, et al. A Bagging Strategy-Based Kernel Extreme Learning Machine for Complex Network Intrusion Detection[J]. EAI Endorsed Transactions on Scalable Information Systems. 21(33), e8, 2021. http://dx.doi.org/10.4108/eai.6-10-2021.171247
29. Liang H, Yang J, Shao M. FE-RetinaNet: Small Target Detection with Parallel Multi-Scale Feature Enhancement[J]. Symmetry, 2021, 13(6): 950.
30. Cao J, Song C, Song S, et al. Lane detection algorithm for intelligent vehicles in complex road conditions and dynamic environments[J]. Sensors, 2019, 19(14): 3166.
31. Gao S H, Han Q, Li D, et al. Representative batch normalization with feature calibration[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021: 8669-8679.

Guang Zhu is with the School of Vehicle and Traffic Engineering, Zhengzhou University of Science and Technology, Zhengzhou 450064 China. Research direction: Image processing, data analysis, artificial intelligence.

Yajuan Liu is with the School of Foreign Languages, Zhengzhou University of Science and Technology, Zhengzhou 450064 China. Research direction: English data processing, big data, pattern recognition.

Jiyue Wang is with the School of Vehicle and Traffic Engineering, Zhengzhou University of Science and Technology, Zhengzhou 450064 China. Research direction: Image processing, data analysis, artificial intelligence.

Received: March 14, 2024; Accepted: June 29, 2024.

Computer Science and Information Systems 21(4):1725–1743 https://doi.org/10.2298/CSIS240314055S

Stacked Denoised Auto-encoding Network-based Kernel Principal Component Analysis for Cyber Physical Systems Intrusion Detection in Business Management

Zhihao Song

School of Business Administration, Zhengzhou University of Science and Technology
Zhengzhou 450064 China
zsshanln@163.com

Abstract. At present, the network data under the environment of industrial information physical system is larger and more complex. Traditionally, feature extraction by machine learning is cumbersome and computation-intensive, which is not conducive to anomaly detection of industrial network data. To solve the above problems, this paper proposes a stacked denoised auto-encoding network based on kernel principal component analysis for industrial cyber physical systems intrusion detection. Firstly, a novel kernel principal component analysis method is used to reduce the data feature dimension and obtain a new low-dimension feature data set. Then, a multi-stacked denoised auto-encoding network model is used to classify and identify the data after dimensionality reduction by voting. Experimental results show that the proposed method has better classification performance and detection efficiency by comparing the state-of-the-art intrusion detection methods.

Keywords: industrial cyber physical systems, intrusion detection, stacked denoised auto-encoding network, kernel principal component analysis.

1. Introduction

With the development of industrial control system information technology, more and more industrial control networks are connected with external public networks [1], forming an open network of information sharing, resulting in the increase of global industrial control network attack events. The risk root of industrial control system mainly includes the defects of system protocol technology, the loopholes of industrial software and the congenital deficiency of hardware design. And the industrial network traffic has the characteristics of stability, correlation, self-similarity and the difference between the network protocol technology and the general protocol, so that the traditional network anomaly detection scheme cannot be simply transplanted into the industrial network [2,3]. The methods of industrial intrusion detection are mainly misuse intrusion detection and abnormal intrusion detection. Misuse intrusion detection technology is a kind of model based on known intrusion behavior data, and then matches all the newly accessed data, and determines that the matched data has the intrusion behavior. Therefore, the misuse of intrusion detection technology can accurately describe the characteristics of attack behavior, reduce the false alarm rate of the system, but it can improve the missing alarm rate. The abnormal intrusion detection technology is a model based on the data of normal behavior, and then matches all the new access technology, and determines the data of matching as normal

behavior [4]. This abnormal intrusion detection technology reduces the false alarm rate, but increases the false alarm rate at the same time.

In the application of industrial intrusion detection, attack types emerge one after another. At present, scholars have carried out many researches on invasion attack and security protection. The used algorithms include statistics-based [5], rule inference-based [6] and neural networks-based methods [7]. Shang et al. [8] selected Modbus/TCP function code as the research object, combined with SVM (Support Vector) Machine), he proposed a Modbus/TCP communication function code sequence anomaly detection method based on SVM, which realized the identification of attack behavior and anomaly behavior that firewall and intrusion detection system could not identify. But the study did not involve learning from new data later on. Therefore, the complexity of industrial network environment cannot be satisfied. Wan et al. [9] proposed a Modbus/TCP communication access control method based on function code deep detection based on deep packet resolution technology without changing Modbus/TCP protocol, which not only realized traditional function code and address range detection, but also realized range detection. However, the system used fixed protocol to match, and also had poor learning ability, which could not meet the complexity and expansibility of industrial network. The intrusion detection system designed by Ferrag et al [10] analyzed the telemetry data in the industrial control network and used the REPTree decision tree algorithm to identify the communication between different machines according to the hop number of the data propagating in the network, so as to distinguish the attacker and the engineer machine inside the network. This method could effectively identify the threats within the network. Ocaa et al. [11] converted Modbus RTU/ASCII data in the industrial control network to Modbus/TCP. Then it used the improved Snort software for threat analysis, it could effectively protect against denial of service, response injection, command injection and reconnaissance attack. Finally, the detected data was converted to the original Modbus RTU/ASCII format again. But this method could not effectively solve the attack of fake data injection in industrial network. Tong et al. [12] proposed two OCSVM intrusion detection algorithms to be applied in fault detection, and the Gaussian kernel parameter optimization models were DFN and DTL, respectively. In the application of fault diagnosis, it could effectively detect various types of faults and improve the accuracy. But the false detection rate had also increased significantly. This paper proposes a stacked denoised auto-encoding network based on kernel principal component analysis for industrial cyber physical systems intrusion detection. It shows a better detection efficiency compared with other methods through experiments.

This paper is organized as follows. In section 2, we give the related works. Section 3 detailed shows the proposed intrusion detection method. Experiments and analysis are displayed in section 4. There is a conclusion in section 5.

2. Related Works

In recent years, with the rapid development of automation control and communication, computer and other related technologies, the theory and application of cyber physical systems (CPS) have also been developed rapidly [13]. CPS integrates computer, communication network and control technology efficiently, so that the physical system is equipped with precision control, remote collaboration and autonomy. CPS is applied to industry,

forming industrial cyber physical system systems (ICPS), which is widely used in key fields such as intelligent energy, intelligent mobility and intelligent manufacturing [14]. ICPS requires carrier-level quality of service for bearer networks, and puts forward higher requirements for network security, reliability, controllability and manageability [15].

The network security of communication bridge is particularly important. Intrusion Detection Systems (IDS) is another network security barrier of ICPS after network firewall. It mainly detects internal and external attacks through monitoring and analyzing activities on host or network [16]. Intrusion detection technology can be divided into two types. One is network-based IDS, which captures the communication data packets between the main control center and the site or on-site devices to evaluate whether the data poses threats to the system. According to different traffic characteristics, the detection model is established and the abnormal behavior of the system is detected. The other is application-based IDS. ICPS stores control and measurement data generated by sensors and actuators in a history server. Application-based IDS reads the application data of the history server to detect whether the system is attacked [17]. Since the information source of IDS based on ICPS application can be collected from different remote field devices such as programmable logic controllers and remote terminal units. Therefore, IDS based on ICPS applications can be deployed in a variety of ways, such as on history servers, standalone servers, and off-line servers.

At present, most of the intrusion detection systems of ICPS use the traditional machine learning methods to extract useful information from labeled data sets to train the detection model. However, the high-speed interconnection between devices in today's network environment leads to massive data, which brings new challenges to IDS design. How to realize stable and efficient intrusion detection in the network environment with massive data is the primary problem in the current research. Abbasi et al. [18] proposed a sensor cloud intrusion detection algorithm based on parallel discrete optimization feature extraction and machine learning method in fog computing mode to effectively improve the success rate and miss rate of anomaly detection in view of large-scale high-dimensional data and variable intrusion behavior of sensor cloud. Wook et al. [19] proposed a new method based on partial least squares (PLS) and core vector machine. PLS algorithm was used to extract the principal component of network data to construct feature set, and then CVM was used to complete abnormal intrusion detection and judgment. This method could deal with large-scale data quickly. Hadeel et al. [20] proposed a feature selection pigeon optimizer algorithm for intrusion detection systems. The algorithm performed well in true positive rate (TPR), false rejection rate (FRR), accuracy (Acc) and F-measure. Guo et al. [21] proposed a one-class SVM based on incremental single-class support vector machine. OCSVM algorithm was used to learn the normal behavior communication mode from Modbus/TCP data information, and the current training sample set was reduced by using adjacent classification interval and KKT condition. The reduced samples were trained again to improve the incremental learning speed of intrusion detection system while ensuring higher classification accuracy. Lv et al. [22] proposed a new intrusion detection model based on optimal mixing and functional extreme learning machine. This method used gravity search algorithm and differential evolution algorithm to optimize the parameters of hybrid kernel extreme learning machine, so as to improve the global and local optimization ability in the process of attack prediction, and introduced kernel

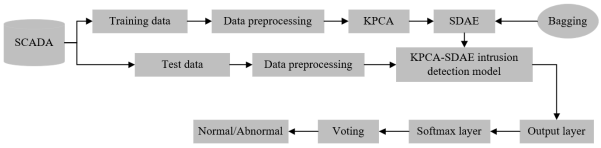


Fig. 1. Framework of ICPS intrusion detection model based on KPCA-SDAE

principal component analysis algorithm to reduce the dimension and feature extraction of intrusion detection data, with high detection accuracy and time saving.

The above methods can improve detection accuracy and other performance from different perspectives, but the following problems need further study. First, the network data in ICPS presents complex nonlinear relations, resulting in low correlation between feature attributes in the data set. Therefore, they are not suitable to use traditional linear dimensionality reduction strategies such as Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) to screen out highly relevant features. Second, network data in industrial scenarios are faced with more severe noise interference than traditional IT networks, which may lead to data redundancy or even some missing attribute values. The above methods cannot effectively solve these problems. Therefore, this paper studies a deep learning hybrid strategy detection model to enhance the anti-jamming ability of the system. The improved KPCA is introduced as the data dimension reduction model for the system to enhance the robustness of the acquired data. Integrated learning method is used to integrate stacked denoised auto-encoding network with different structures, and softmax classifier is used to classify and identify the data after dimensionality reduction, so as to reduce the dependence of system performance on the parameter structure setting of a single deep belief network.

3. ICPS Intrusion Detection Model Based on KPCA-SDAE

KPCA structure and stacked denoised auto-encoding (SDAE) network model are used to construct ICPS intrusion detection framework, namely, KPCA-SDAE, as shown in Figure 1.

In the training stage, the original communication traffic data obtained from SCADA system is preprocessed, and then imported into KPCA model for feature extraction and data dimension reduction. Then, it adds labels to the data after dimensionality reduction as training samples of SDAE. Then Bagging technology is used to divide the training data. The KPCA-SDAE model is constructed by using the data of different groups to train multiple SDAE networks and fix the parameters of SDAE model. The parameters of KPCA-SDAE model are adjusted by supervised learning algorithm.

In the test phase, the preprocessed data is fed into the KPCA-SDAE model. After the output layer, the binary classification or multiple classification is realized by Softmax classification, and finally the Voting mechanism is adopted to determine the final predicted value. The performance of the intrusion detection algorithm is evaluated by ACC and FN. The main components of ICPS intrusion detection model based on KPCA-SDAE are KPCA data dimension reduction model and ensemble classifier construction,

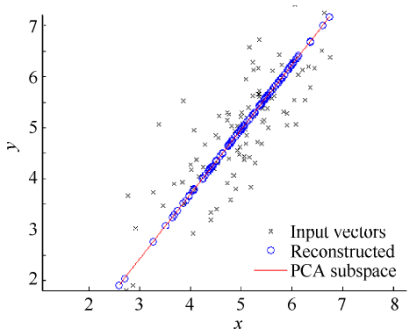


Fig. 2. PCA process

in which KPCA data dimension reduction model is mainly used to eliminate data redundancy and conduct data dimension reduction processing, and ensemble classifier construction is mainly used to classify and identify processed data.

3.1. Dimension Reduction Optimization Based on KPCA

Basic theory of principal component analysis (PCA).

Assuming that m samples have been selected and each sample contains d variables (d-dimensions) [23], the sample matrix can be expressed as:

$$x_{d\times m} = [x^1, \cdots, x^j, \cdots, x^m]. \tag{1}$$

The $j-th$ sample is $x^j = [x_1^j, x_2^j, \cdots, x_d^j]^T$. The basic principal component analysis is how to project sample points onto a hyper-plane, and make the projection of sample points as separate as possible on this hyper-plane, so as to find the dominant direction. Let the projection of our sample point x^j onto our new space be $U^T x^j$. U^T is the projection coefficient matrix. In order to keep the projection of the sample points as separate as possible, the variance of the sample points after projection should be maximized, as shown in Figure 2.

The variance of sample points after projection is $\sum_{j=1}^{m} U^T x^j (x^j)^T U$, so the projection matrix must satisfy:

$$\max_U tr(U^T X X^T U), s.t. U^T U = I. \tag{2}$$

Formula (2) can be obtained by Lagrange multiplier method.

$$X X^T u^i = \lambda_i u^i (i = 1, 2, \cdots, d). \tag{3}$$

The eigenvalue decomposition of Equation (3) is carried out, and the obtained eigenvalues are sorted from large to small $\lambda_1 > \lambda_2 > \cdots > \lambda_d$. Then it takes the eigenvectors corresponding to the first d' eigenvalues to form the reconstruction space. d' value can be selected according to the minimum dimension established by the following formula of reconstruction threshold t [12].

$$\frac{\sum_{i=1}^{d'} \lambda_i}{\sum_{i=1}^{d} \lambda_i} \geq 1. \tag{4}$$

KPCA.

It is obvious that PCA method directly discards feature vectors after in the process of spatial reconstruction, ignoring a lot of information. And from high dimensional space to low dimensional space, it is a linear mapping, so it has certain limitations. Kernel principal component analysis [24] in nonlinear dimensionality reduction is one of the effective ways to solve this problem.

To introduce the kernel method, the following transformation is performed on Equation (3).

$$(\sum_{i=1}^{m} z_i z_i^T) u^i = \lambda_i u^i. \tag{5}$$

Where z_i is the image of sample point x^i in higher dimensional feature space. It can be known from Equation (5).

$$u^j = \frac{1}{\lambda_i} (\sum_{i=1}^{m} z_i z_i^T) u^j = \sum_{i=1}^{m} z_i \alpha_i^j. \tag{6}$$

Where $\alpha_i^j = \frac{z_i^T u^j}{\lambda_j}$. Suppose $z_i = \varphi(x^i)$, so equation (5) is transformed into:

$$(\sum_{i=1}^{m} \varphi(x^i) \varphi(x^i)^T) u^j = \lambda_j u^j. \tag{7}$$

Equation (6) is transformed into:

$$u^j = \sum_{i=1}^{m} \varphi(x^i) \alpha_i^j. \tag{8}$$

By introducing the kernel function,

$$\kappa(x^i, x^j) = \varphi(x^i)^T \varphi(x^j). \tag{9}$$

After substituting Equation (8) and Equation (9) into Equation (7), we can get,

$$K \alpha^j = \lambda_j \alpha^j. \tag{10}$$

Where $\alpha^j = [\alpha_1^j, \alpha_2^j, \cdots, \alpha_m^j]^2$. The kernel matrix K is:

$$K = \begin{bmatrix} \kappa(x^1, x^1) & \cdots & \kappa(x^1, x^j) & \cdots & \kappa(x^1, x^m) \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ \kappa(x^m, x^1) & \cdots & \kappa(x^m, x^j) & \cdots & \kappa(x^m, x^n) \end{bmatrix}$$

It is obvious that Equation (10) is the eigenvalue solution problem. After obtaining the eigenvalue and eigenvector of Equation (10), the following optimization design space can be reconstructed.

Reconstructing and optimizing the design space based on KPCA.

According to the above theory, for any sample x, after its projection, the j-dimension coordinate in the new design space is:

$$z_j = \sum_{i=1}^{m} \alpha_i^j \kappa(x^i, x). \tag{11}$$

The following kernel function is used in this paper.

$$\kappa(x^i, x^j) = (x^i)^T x^j. \tag{12}$$

Reconstruct the design space and substitute Equation (11) into Equation (10), we can get,

$$z_j = (\sum_{i=1}^{m} u_i^j (x^i)^T) x. \tag{13}$$

Therefore, the relationship between any sample x and the projected coordinate can be written in matrix form, i.e,

$$Z = Ax. \tag{14}$$

Where $Z = [z_1, z_2, \cdots, z_d]^T$. The coefficient matrix A is:

$$A = \begin{bmatrix} \sum_{i=1}^{m} \alpha_i^1 x_1^i & \cdots & \sum_{i=1}^{m} \alpha_i^1 x_d^i \\ \vdots & \cdots & \vdots \\ \sum_{i=1}^{m} \alpha_i^d x_1^i & \cdots & \sum_{i=1}^{m} \alpha_i^d x_d^i \end{bmatrix}$$

Multiply both sides of equation (14) by A^{-1}, we can get the following formula,

$$x = A^{-1} Z. \tag{15}$$

It is obvious that equation (14) establishes the mapping relationship between the original variable and the projected variable. Equation (15) gives the mapping relationship between projection variables and original variables, which is the key to optimize the design space after dimensionality reduction. The two relations can be used to reconstruct the optimization space and search for optimization in the reconstructed optimization space. Then the optimization results are re-mapped back to the original optimization space.

The dimension reduction optimization algorithm based on linear kernel principal component analysis is shown in **Algorithm 1**.

3.2. SDAE Network

Auto-encoding (AE) network [25] belongs to the non-recurrent neural network in feed-forward neural network, which is a 3-layer unsupervised learning network, including encoder and decoder, as shown in Figure 3.

The encoder transforms the input sample $x_i (i = 1, 2, \cdots, n)$, where n is the sample number, into a lower-dimensional space through the activation function $h(\cdot)$.

Algorithm 1 Linear kernel principal component analysis.

1: **Input.** Get sample matrix $X_{d\times m} = [x^1, \cdots, x^j, \cdots, x^m]$. And it is normalized and decentralized to form standardized data.
2: **Output.** Optimized solution.
3: Calculate the kernel matrix K corresponding to the sample matrix using the above standardized data. The eigenvalues and eigenvectors of the matrix are obtained by eigenvalue decomposition.
4: Order the eigenvalues corresponding to the above determined eigenvectors from large to small, and determine the reconstruction threshold.
5: Use the sorted feature vectors in (3) to calculate the coefficient matrices A and A^{-1}.
6: The original design space was reduced in dimension, and the design variables after dimension reduction were optimized.
7: End

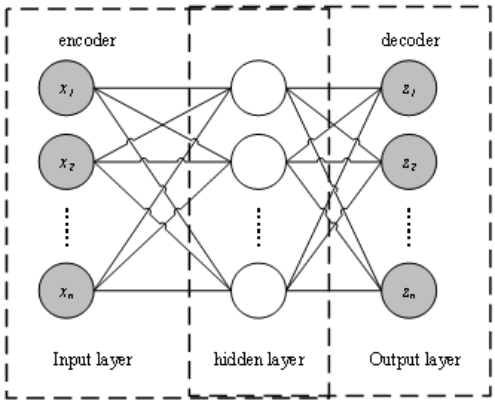


Fig. 3. AE network structure

$$y = h(xw + b). \tag{16}$$

Where y is the output matrix of the hidden layer. x is the input sample matrix, $x = [x_1, x_2, \cdots, x_n]$. w is the weight matrix of the encoder. b is the bias matrix of the encoder.

The decoder inverts the signal y from the low-dimensional space to the high-dimensional space through the activation function $f(\cdot)$, and then reconstructs the input sample.

$$z = f(x) = f(h(xw' + b')). \tag{17}$$

Where z is the reconstructed sample matrix, $z = [z_1, z_2, \cdots, z_n]$. z_i is the input sample of reconstruction. w' is the weight matrix of decoder. b' is the bias matrix of the decoder.

In practical application, the training samples are often mixed with noise, resulting in the deviation of features acquired through AE network learning. Denoised auto-encoding (DAE) network adds noise to AE network and extracts more robust features by learning noisy data [26]. DAE network structure is shown in Figure 4. Conditional distribution C_L is introduced, and the probability of damaged samples is given to obtain the input sample

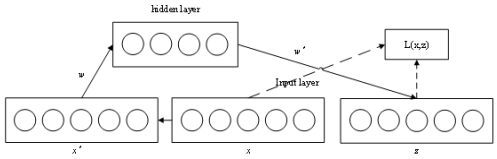


Fig. 4. DAE network

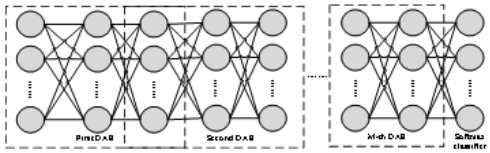


Fig. 5. DAE network

matrix x' after adding noise. The loss function $L(x, z)$ is used to calculate the reconstruction error, and the effective features of the input samples are extracted by minimizing the reconstruction error.

In order to extract deep features of input samples, DAE is stacked layer by layer with deep network structure to form SDAE network, as shown in Figure 5. The input of the hidden layer of the next DAE network is the output of the hidden layer of the previous DAE network, and the network uses unsupervised learning to complete the stacking process. SDAE network, as an unsupervised learning network, can extract deep features of input samples and carry out feature coding, but cannot classify samples. Therefore, the Softmax classifier [27] is added to the last layer of the SDAE network to make the entire network as a supervised learning network.

3.3. Classifier Construction Based on a Novel Bagging Algorithm

DBN (deep belief network) structure is introduced to classify and identify the KPCA model data after dimensionality reduction. Although DBN has strong ability of feature extraction and data mining, it requires artificial selection for the optimal structure (including DBN network layer number and node number) and the optimal parameters (including vector, momentum, etc.). Even the optimal structure of DBN is not fixed. Therefore, the optimization of DBN structure and parameters is a time-consuming process. Therefore, ensemble DBN (EDBN) uses parallel integration of multiple DBN structure learning models to avoid spending a lot of time on DBN structure selection and parameter adjustment, which can effectively reduce the dependence of detection model on the optimal parameters of a single DBN model.

DBN is composed of a series of Restricted Boltzmann machines (RBM) and reverse transmission stacked neural networks. The training steps of DBN model are as follows:

1) Low-level DBN training.

– Initialize model parameters.

Let the initial value of the weight in parameter $\theta_1 = (W_1, a, b_1)$ of the model be $W_1 \in (0, 1)$. The initial value of both visible layer bias a and hidden layer bias b_1 is set to 0.

– DBN contrast divergence algorithm.
First, compute the hidden layer output forward. Let $V = (v_1^{c_1}, v_2^{c_1}, \cdots, v_k^{c_1})$ be the $c_1 - th$ updated input value in the visible layer. $w_{i,1j}$ is the connection weight between the $i - th$ neuron in visible layer V and the $j - th$ neuron in the forward hidden layer $H1$, where,

$$h_{1j} = \sum_{i=1}^{k} v_i^{c_1} w_{i,1j} + b_{1j}. \tag{18}$$

Neuron h_{1j} of the hidden layer is activated at the $c_1 - th$ update, that is, the output value of the hidden layer is,

$$P(h_{1j}^{c_1} = 1 | v_i^{c_1}) = S(h_{1j}^{c_1}). \tag{19}$$

Second, reverse reconstruct the visibility layer output. The output of the hidden layer reconstructs the input of the visible layer. At the $c_1 - th$ update, the input value of neuron $v_i^{c_1}$ reconstructed in the visible layer is:

$$v_{i'}^{c_1} = \sum_{j=1}^{l_1} S(h_{1j}^{c_1}) + a_i. \tag{20}$$

Similarly, the probability that neuron v_i of the reconstructed visible layer is activated at the $c_1 - th$ update is the output value of the reconstructed visible layer.

$$P(v_{i'}^{c_1} = 1 | h_{1j}^{c_1}) = S(v_{i'}^{c_1}). \tag{21}$$

Third, reconstruct the output to the front hidden layer. The hidden layer continues to be reconstructed from the output of the reconstructed visible layer. The probability that neuron h_{1j} of the hidden layer is activated again, namely, the output value of the forward hidden layer after reconstruction is,

$$P(h_{1j}^{c_1} = 1 | v_{i'}^{c_1}) = S(h_{1j}^{c_1}). \tag{22}$$

Where $h_{1j'}^{c_1} = \sum_{i=1}^{l_1} S(v_{i'}^{c_1}) w_{i,1j} + b_{1j}$ represents the input after the reconstruction of the forward hidden layer at the $c_1 - th$ update.
Fourth, update model parameters. After the visible layer and hidden layer are reconstructed, the updated model parameter θ_1 can be obtained by the following formulas:

$$w_{i,1j} \leftarrow w_{i,1j} + \lambda \times (S(h_{1j}^{c_1}) \times v_i^{c_1} - S(h_{1j'}^{c_1}) \times v_{i'}^{c_1}). \tag{23}$$

$$a_i \leftarrow a_i + \lambda \times (v_i^{c_1} - v_{i'}^{c_1}). \tag{24}$$

$$b_{1j} \leftarrow b_{1j} + \lambda \times (S(h_{1j}^{c_1}) - S(h_{1j'}^{c_1})). \tag{25}$$

Where λ represents the learning rate.
When $c_1 < T$, repeat step (2) with updated model parameters as initial values. If $c_1 = T$, perform Step (3).

– Stop DBN training. Output the underlying DBN model parameters, namely $\theta_1 = (W_1, a, b_1)$.

2) DBN training.

– DBN pre-training. Suppose that DBN is stacked with three RBMs, and the training process of each RBM is the same as step (1). The output of the RBM hidden layer of the upper layer serves as the input of the visible layer of the RBM of the next layer. When the model parameter $\theta_3 = (w_3, b_2, b_3)$ of the last layer RBM is output, the DBN pre-training is finished.
– (2)Fine-tuning of DBN model. DBN fine-tuning is a supervised training process. In the fine-tuning process, BP algorithm is used to reverse update the parameters of the model, and the cross entropy error function is used as the optimization objective function. The objective function can be calculated as follows:

$$J = \frac{1}{m}\sum_{k=1}^{m}(-\sum_{i=1}^{r_3}(o_i^k \ln \hat{o}_i^k)). \tag{26}$$

Where $\hat{o}_i^k$ and o_i^k respectively represent the output value (predicted value) and real label value of the $i-th$ neuron corresponding to the $k-th$ sample in the DBN output layer. By using the partial derivatives of the cross entropy error function to the weight terms and bias values of each network layer, the weight terms and bias terms of each network layer can be dynamically updated. When the J value converges or reaches the set number of generation selection, the weight items and offset values of each layer are fixed to complete the fine tuning of the system.
EDBN model adopts Bagging algorithm to integrate DBN structure. Assuming that the size of the training set is N, Bagging algorithm selects m sub-training sets with size N (including repeated samples) by self-sampling method. Then, m data sets are used to train m DBNSs with different hyperparameter settings. And m DBN base classifier models can be obtained. Finally, the predictive label value is obtained by majority voting mechanism. The pseudo-codes of Bagging algorithm are described in **Algorithm 2**.

4. Experiments and Results Analysis

In order to verify the feasibility of SDAE-KPCA intrusion detection algorithm, real data obtained from SCADA system is selected as the test data set [28]. Each instance data in the dataset contains 26 feature attributes and one tag attribute. There are eight types of labels: Normal, NMRI (naive malicious response injection attack), CMRI (malicious response injection attack), MSCI (malicious status command injection attack), MPCI (malicious parameter command injection attack), MFCI (malicious function code injection attack), DOS (denial of service attack), and Reco (reconnaissance attack).

The experiment is carried out under Inter Core i6 CPU, 32GB RAM, GTX1060 GPU environment and Windows10 operating system. Python3.6 is used for simulation experiments. The algorithm is implemented by deep learning framework Tensorflow. Accuracy (ACC) and false negative (FN) are used to measure the detection accuracy of IDS, while

Algorithm 2 Bagging algorithm.

Input: Training set $S = (x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N)$, where $x_k \in R^l$, number of classifiers m.
Output: The final classifier H_c.
for $t = 1, 2, \cdots, m$ do
$S_t =$ sample with Bootstrap(S,N)
$h_t = L(S_t)$
end for
sample with Bootstrap(S,N)
$I = \phi$
for $t = 1, 2, \cdots, N$ do
$r = Random - Iterger(1, N)$
add $S[r]$ to I
return I
end for

training time (TRT) and test time (TET) are used to measure the complexity of the algorithm. FN refers to the probability of abnormal traffic being misjudged as normal traffic in the sample. Considering that in the intrusion detection system, the threat of abnormal traffic being misjudged as normal traffic is greater than that of normal traffic being misjudged as abnormal traffic, FN is selected as one of the detection accuracy indexes.

The dataset contains 80000 data sets , and the distribution of sample data is shown in Table 1. There are only 780 samples from MSCI and 575 from MFSI. According to the ratio of 8:1:1, the sample is divided into three parts: training set, verification set and test set. The training set is used to train the IDS model and simulate the data samples. The validation set is used to adjust the hyperparameters of IDS model. The test set is used to test the accuracy of IDS model to identify network attacks and the generalization ability of detection model. Common parameters include weight values and bias values of each network layer. The hyperparameters include network depth (DAE and DBN layers), sparse index, and the setting of generation selection times, etc.

Table 1. Distribution of sample data

Normal	Anomaly	Anomaly	Anomaly	Anomaly	Anomaly	Anomaly	Anomaly
Type	CMRI	NMRI	MSCI	MPCI	MFCI	DOS	Reco
44407	15466	2763	780	7367	575	1837	6805

After pre-processing and min-max normalization of the original data, 64000 groups of data are transferred into the DAE model in batches after removing label values for pre-training. In order to make the acquired features better represent the original data features, the error between the output value of the hidden layer and the real value of the last layer of the regular de-noising autocoding network in DAE is fine-tuned. Then, the data set of DAE dimension reduction operation is collected by self-help sampling method to obtain the data of m group with the size of 80000. Then, m groups of DBN models are trained, and the optimal DBN models are obtained by changing the DBN network depth, generation

times and other super parameters. Bagging ensemble method is used to integrate m DBN models, and the final prediction value is determined by voting mechanism.

Before the final testing of the SDAE-KPCA model, a process of parameter tuning is required. Since there is no automatic method for parameter tuning, empirical parameter tuning is used to adjust parameter settings to achieve the best performance of the model. The initial parameters of DAE model are as follows: network depth=2, learning rate=0.02, L1 regularization coefficient =0.01, sparse penalty factor=0.15 and sparse parameter=0.05, the number of pre-training rounds=100 and the number of fine-tuning rounds=50. The initial parameters of EDBN are set as follows: the number of integrated DBN=3, the network depth of each DBN=3, the learning rate=0.001, the number of training rounds=100, and the number of fine tuning rounds is 50.

In order to test the model's performance in identifying abnormal traffic, the model is tested in detecting binary samples, that is, Normal and Abnormal, and in the multi-classification samples,that is, NMRI, CMRI, MSCI, MPCI, and MFCI, DOS and Reco. The model is compared and analyzed, including the performance of different model parameters and network structure design, as well as the comparison with the current mainstream intrusion detection algorithms.

4.1. Classification Network Parameters Determination

We take the SDAE network depth=2, the learning rate=0.02, the L regularization coefficient =0.01, the sparsity penalty factor $\beta = 0.10$, the sparsity parameter $\rho = 0.05$, the number of pre-training rounds=100, and the number of fine-tuning rounds=50 as the standards. Only one parameter is changed in each group of experiments, and the experimental results are shown in Tables 2-7.

1. Network structure setting.
   Under the premise of fixed learning rate, L regularization coefficient and other parameters, the training error obtained by changing only the depth of network structure and the number of neurons in hidden layer is shown in Table 2 and Table 3. As shown in Table 2, when the network depth is 2 and the hidden layer neuron is set to 30-10, the mean square error (ERR) between the network output value and the real value is the smallest.

Table 2. Error values with different neuron structures

No.	Neuron structure	ERR
1	30-10	0.001657
2	30-20	0.002894
3	30-30	0.002944
4	40-10	0.005368
5	40-20	0.005813
6	40-30	0.005995

2. Parameter setting.

Table 3. Error values with different training periods

Training period	30-10	30-20	30-30	40-10	40-20	40-30
0	0.13	0.13	0.13	0.15	0.20	0.21
2	0.06	0.07	0.12	0.13	0.14	0.18
4	0.06	0.07	0.12	0.13	0.14	0.15
6	0.06	0.07	0.12	0.13	0.13	0.14
8	0.06	0.02	0.12	0.13	0.13	0.14
10	0.01	0.02	0.06	0.10	0.12	0.13
20	0.01	0.02	0.05	0.07	0.12	0.13
40	0.01	0.02	0.05	0.07	0.09	0.09
60	0.01	0.02	0.05	0.07	0.09	0.09
80	0.01	0.02	0.05	0.07	0.08	0.09
100	0.01	0.02	0.05	0.07	0.08	0.09

When SDAE network depth is fixed as 2, network structure is 30-10, learning rate and other parameters are preset values, the training error obtained by changing only the number of pre-training rounds (PTR) or fine-tuning rounds (FTR) is shown in Table 4. It can be seen from Table 3 that when the number of PTR and FTR values are set as 100 and 100, the ERR of the network is the smallest.

Table 4. Error values with different PTR and FTR

No.	PTR	FTR	ERR
1	100	50	0.002874
2	100	100	0.001963
3	100	25	0.003647
4	200	50	0.002971
5	50	50	0.003923
6	200	200	0.002685

Under the condition that the number of PTR and FTR of SDAE is fixed as 100 and 100, the network depth is 3 (50-50-20), the learning rate and other parameters are set as default values, table 5 shows the worth training error of only changing the coefficient penalty factor β or sparse parameter ρ. It can be seen from Table 5 that when β and ρ are set to 0.10 and 0.10, the error value is minimum.

Under the condition that the number of pre-training and fine-tuning rounds of SDAE is fixed as 100 and 100, $\beta = 0.10$ and $\rho = 0.10$, network depth=2, and learning rate is set as the default value, the training error obtained by only changing the L1 regularization coefficient is shown in Table 5. According to Table 6, when L1 regularization coefficient is 0.010, ERR reaches the local minimum value. When the regularization coefficient is too small, it is easy to cause under-fit phenomenon and the error will increase. When the regularization coefficient is too large, it is easy to cause over-fit phenomenon, and the data learning time will increase accordingly.

In SDAE, the number of pre-training and fine-tuning rounds are 100 and 100, $\beta = 0.10$, $\rho = 0.10$, L1 regularization coefficient=0.010, and network depth=2. The initial

Table 5. Error values with different β and ρ

No.	β	ρ	ERR
1	0.10	0.05	0.001943
2	0.20	0.05	0.003029
3	0.25	0.05	0.003141
4	0.10	0.10	0.001878
5	0.20	0.10	0.003104
6	0.25	0.10	0.003829

Table 6. Error values with different L1 regularization coefficients

No.	L1	ERR
1	0.005	0.002447
2	0.010	0.001877
3	0.015	0.003105
4	0.020	0.003218
5	0.025	0.004115
6	0.030	0.005934

learning rate is 0.01, and the results obtained by changing this initial value are shown in Table 7. As can be seen from the table, dynamic changing learning rate will reduce the influence of learning rate on data fitting accuracy. 0.01 is selected as the final SDAE learning rate.

Table 7. Error values with different learning rates

No.	L1	ERR
1	0.01	0.001775
2	0.02	0.001865
3	0.03	0.002467
4	0.04	0.002536
5	0.05	0.002917
6	0.06	0.003348

4.2. Classification Performance Analysis

The parameters obtained above are used in the classification experiments. 64000 training samples are obtained after dimensionality reduction. Self-sampling method is used to extract three groups of data sets, and the data extraction results are shown in Table 8.

Data sets D1, D2 and D2 are used to train SDAE with three different parameter settings respectively. The training results are shown in Table 9. Finally, the three groups of SDAE structure settings are shown in Table 10.

Table 8. Error values with different learning rates

Training set			
Data set	Normal	Anomaly	Total
D1	27000	23000	
D2	26000	24000	50000
D3	27000	23000	
Result			
Data set	Normal	Anomaly	Total
D1	7800	6200	
D2	7500	6500	14000
D3	7300	6700	

Table 9. Accuracy with different training periods

Method	0	10	20	30	40	50	60	70	80	90	100
SDAE1	0.905	0.925	0.926	0.935	0.935	0.935	0.935	0.935	0.935	0.935	0.935
SDAE2	0.898	0.926	0.927	0.935	0.935	0.935	0.935	0.935	0.935	0.935	0.935
SDAE3	0.888	0.926	0.927	0.935	0.935	0.935	0.935	0.935	0.935	0.935	0.935

4.3. Comparison Experiments

The SDAE-KPCA algorithm is compared with other algorithms (including PCA-BP, DBN, MDCO [29], RFSS [30] and IDSGAN [31]) using the same data sets. The dichotomous comparison results of detection models are shown in Table 11. As can be seen from Table 11, compared with PCA-BP, DBN, MDCO, RFSS and IDSGAN model, the accuracy of proposed method increases by 5.12%, 3.90%, 2.32%, 1.85% and 1.26%, the FN decreases by 1.45%, 1.33%, 1.17%, 0.77% and 0.73%, respectively. The time complexity of the proposed algorithm is slightly higher than that of the comparison algorithms, and the TRT training time is higher than that of other three detection methods. However, the test time TET is similar to the comparison algorithm for the test set with a small amount of data. The comparison results of multiple classifications are shown in Table 12. It can be seen from Table 12 that SDAE-KPCA has significantly improved accuracy compared with other algorithms in the case of multiple classifications, and the error leakage rate is relatively low. However, due to the small number of samples of MSCI and MFCI attacks, the detection effect of these two types of attacks is relatively poor, but compared with other comparison algorithms, the detection performance is significantly improved.

Table 10. Three groups of SDAE structure parameter settings

Class	Neuron structure	Learning rate	Discard rate	ACC/%
SDAE1	100-100	0.001	0.2	94.79
SDAE2	100-50	0.002	0.1	94.82
SDAE3	50-100	0.001	0.1	94.83

Table 11. Results of dichotomies with different methods

Method	ACC/%	FN/%	TRT/s	TET/s
PCA-BP	92.67	2.61	37.4	8.8
DBN	93.89	2.49	40.9	9.1
MDCO	95.47	2.33	46.1	9.4
RFSS	96.21	1.93	47.8	10.8
IDSGAN	96.53	1.89	46.3	11.3
SDAE-KPCA	97.79	1.16	59.9	11.9

Table 12. Results of dichotomies with different methods

Type	MDCO	MDCO	RFSS	RFSS	IDSGAN	IDSGAN	SDAE-KPCA	SDAE-KPCA
Type	ACC	FN	ACC	FN	ACC	FN	ACC	FN
Normal	0.959	0.0224	0.953	0.0209	0.961	0.0197	0.983	0.0101
NMRI	0.927	0.00247	0.917	0.0174	0.934	0.0170	0.925	0.0091
CMRI	0.948	0.0204	0.939	0.0219	0.951	0.0207	0.975	0.0183
MSCI	0.776	0.0323	0.794	0.0375	0.903	0.0331	0.914	0.0181
MPCI	0.954	0.0247	0.955	0.0247	0.958	0.0228	0.968	0.0154
DOS	0.987	0.0178	0.981	0.0155	0.986	0.0129	0.999	0.0014
Reco	0.968	0.0135	0.982	0.0168	0.979	0.0154	0.999	0.0092

5. Conclusion

This paper studies an integrated intrusion detection model based on deep auto-encoding network. KPCA is used to reduce data redundancy. Integrating SDAE overcomes the difficulty of DAE structure selection and significantly reduces the dependence of model on DAE parameter setting. Compared with other methods, the proposed algorithm has a better performance in dealing with large-scale industrial network data, which significantly improves the robustness and anti-interference ability of intrusion detection system, while reducing the dimension of data.

References

1. A. Al-Abassi, H. Karimipour, A. Dehghantanha, et al., "An ensemble deep learning-based cyber-attack detection in industrial control system," *IEEE Access*, vol. 8, pp. 83965–83973, 2020.
2. X. Yan, Y. Xu, X. Xing, et al., "Trustworthy network anomaly detection based on an adaptive learning rate and momentum in IIoT," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 6182–6192, 2020.
3. F. Souhe, A. Boum, P. Ele, et al., "Fault detection, classification and location in power distribution smart grid using smart meters data," *Journal of Applied Science and Engineering*, vol. 26, no. 1, pp. 23–34, 2022.
4. M. Novaes, L. Carvalho, J. Lloret, et al., "Long short-term memory and fuzzy logic for anomaly detection and mitigation in software-defined network environment," *IEEE Access*, vol. 8, pp. 83765–83781, 2020.
5. S. Anton, L. Ahrens, D. Fraunholz, et al., "Time is of the essence: Machine learning-based intrusion detection in industrial time series data," *2018 IEEE International Conference on Data Mining Workshops (ICDMW). IEEE*, pp. 1–6, 2018.

6. J. Liu, W. Zhang, T. Ma, et al., "Toward security monitoring of industrial cyber-physical systems via hierarchically distributed intrusion detection," *Expert Systems with Applications*, vol. 158, 2020.
7. S. Latif, Z. Idrees, Z. Zou, et al., "DRaNN: A deep random neural network model for intrusion detection in industrial IoT," *2020 International Conference on UK-China Emerging Technologies (UCET). IEEE*, pp. 1–4, 2020.
8. W. Shang, P. Zeng, M. Wan, et al., "Intrusion detection algorithm based on OCSVM in industrial control system," *Security and Communication Networks*, vol. 9, no. 10, pp. 1040–1049, 2016.
9. M. Wan, J. Li, H. Luo, et al., "Dynamic Adjusting ABC-SVM Anomaly Detection Based on Weighted Function Code Correlation," *International Conference on Machine Learning for Cyber Security. Springer, Cham*, pp. 1–11, 2020.
10. M. Ferrag, L. Maglaras, A. Ahmim, et al., "Rdtids: Rules and decision tree-based intrusion detection system for internet-of-things networks," *Future internet*, vol. 12, no. 3, pp. 44, 2020.
11. W. Ocaña, C. Alex, G. Ricardo, et al., "Control and monitoring of electrical variables of a level process using Modbus RTU-TCP/IP industrial communication," *Indian Journal of Science and Technology*, vol. 11, no. 32, 2018.
12. W. Tong, B. Liu, Z. Li, et al., "Intrusion Detection Method of Industrial Control System Based on RIPCA-OCSVM," *2019 3rd International Conference on Electronic Information Technology and Computer Engineering (EITCE). IEEE*, pp. 1148–1154, 2019.
13. H. Habibzadeh, B. Nussbaum, F. Anjomshoa, et al., "A survey on cybersecurity, data privacy, and policy issues in cyber-physical system deployments in smart cities," *Sustainable Cities and Society*, vol. 50, 2019.
14. H. Li, S. Yin, J. Liu, et al., "Novel gaussian approximate filter method for stochastic non-linear system," *International Journal of Innovative Computing, Information and Control*, vol. 13, no. 1, pp. 201–218, 2017.
15. J. Lee, M. Azamfar, J. Singh, "A blockchain enabled Cyber-Physical System architecture for Industry 4.0 manufacturing systems," *Manufacturing letters*, vol. 20, pp. 34–39, 2019.
16. K. Rijswijk, L. Klerkx, M. Bacco, et al., "Digital transformation of agriculture and rural areas: A socio-cyber-physical system framework to support responsibilisation," *Journal of Rural Studies*, vol. 85, pp. 79–90, 2021.
17. J. Leng, H. Zhang, D. Yan, et al., "Digital twin-driven manufacturing cyber-physical system for parallel controlling of smart workshop," *Journal of ambient intelligence and humanized computing*, vol. 10, no. 3, pp. 1155–1166, 2019.
18. J. Abbasi, F. Bashir, K. Qureshi, et al., "Deep learning-based feature extraction and optimizing pattern matching for intrusion detection using finite state machine," *Computers & Electrical Engineering*, vol. 92, 2021.
19. M. Wook, N. Hasbullah, N. Zainudin, et al., "Exploring big data traits and data quality dimensions for big data analytics application using partial least squares structural equation modelling," *Journal of Big Data*, vol. 8, no. 1, pp. 1–15, 2021.
20. A. Hadeel, S. Ahmad, S. Eddin, "A feature selection algorithm for intrusion detection system based on Pigeon Inspired Optimizer," *Expert Systems with Applications*, 2020.
21. Y. Guo, Z. Zhang, F. Tang, "Feature selection with kernelized multi-class support vector machine," *Pattern Recognition*, vol. 117, 2021.
22. L. Lv, W. Wang, Z. Zhang, et al., "A novel intrusion detection system based on an optimal hybrid kernel extreme learning machine," *Knowledge-based systems*, vol. 195, 2020.
23. S. Kanwal, M. TAHIR, H. RAZZAQ, "Principal component analysis and assessment of Brassica napus l. accessions for salt tolerance using stress tolerance indices," *Pak. J. Bot*, vol. 53, no. 1, pp. 113–118, 2021.
24. W. Lee, G. Mendis, M. Triebe, et al., "Monitoring of a machining process using kernel principal component analysis and kernel density estimation," *Journal of Intelligent Manufacturing*, vol. 31, no. 5, pp. 1175–1189, 2020.

25. W. Ma, L. Peng, "Image Quality Transfer with Auto-Encoding Applied to dMRI Super-Resolution," *2021 4th International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE). IEEE*, pp. 828–831, 2021.
26. G. Liu, Y. Fan, J. Zhang, et al., Deep flight track clustering based on spatial-temporal distance and denoising auto-encoding, *Expert Systems with Applications*, vol. 198, 2022.
27. N. Udoh and M. Ekpenyong, "A Knowledge-Based Framework for Cost Implication Modeling of Mechanically Repairable Systems with Imperfect Preventive Maintenance and Replacement Schedule," *Journal of Applied Science and Engineering*, vol. 26, no. 2, pp. 221–234, 2023. https://doi.org/10.6180/jase.202302_26(2).0008
28. J. Ling, Z. Zhu, Y. Luo, et al., "An intrusion detection method for industrial control systems based on bidirectional simple recurrent unit," *Computers & Electrical Engineering*, vol. 91, 2021.
29. W. Liang, K. Li, J. Long, et al., "An industrial network intrusion detection algorithm based on multifeature data clustering optimization model," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 2063–2071, 2019.
30. X. Li, W. Chen, Q. Zhang, et al., "Building auto-encoder intrusion detection system based on random forest feature selection," *Computers & Security*, vol. 95, 2020.
31. Z. Lin, Y. Shi, Z. Xue, "Idsgan: Generative adversarial networks for attack generation against intrusion detection," *Pacific-Asia Conference on Knowledge Discovery and Data Mining. Springer, Cham*, pp. 79–91, 2022.

Zhihao Song is with the School of Business Administration, Zhengzhou University of Science and Technology Zhengzhou 451460 China. Research direction: Business management, Image processing, data analysis, artificial intelligence.

Received: March 14, 2024; Accepted: June 29, 2024.

Computer Science and Information Systems 21(4):1745–1764 https://doi.org/10.2298/CSIS240314056C

A Novel Image Enhancement Method Using Retinex-based Illumination Map Weighted Guided Filtering

Su Chen[1,⋆] and Dahai Li[2]

[1] Department of Mechanical and Electrical Engineering, Henan Vocational College of Water Conservancy and Environment
Zhengzhou 450002, China
chensssu@163.com

[2] School of Electronics and Electrical Engineering, Zhengzhou University of Science and Technology
Zhengzhou 450064, China
ldadahai@163.com

Abstract. Halo artifact, edge detail loss and noise amplification are the main problems in low illumination image enhancement, an image enhancement algorithm combining Retinex and illumination map weighted guided filtering is proposed. The traditional defogging physical models only enhance the images based on dark channels prior, resulting in different depths of field in local areas, and it can lead to some problems such as image overexposure and halo artifacts. To solve this problem, the method of combining light and dark channels is adopted to calculate the atmospheric light value and transmittance. For the problem that edge information is easily lost, the illumination gradient domain weighted guided filtering is utilized to improve the thinning transmittance. Experimental results with the proposed method have obvious improvement in denoising, halo elimination, brightness adjustment and edge preservation in the low-illumination image under different conditions.

Keywords: Image enhancement, Retinex, weighted guided filtering, dark channel.

1. Introduction

In the era of the information technology explosion, computer vision has played an increasingly important role in various fields. However, in the case where the light such as night or weak light is insufficient, the low-illumination type image has low contrast, low brightness, and detail information [1,2]. The imaging quality is seriously decreased, and the visual effect is poor, which is subsequently processed for the image and it has brought great challenges to a large extent to reduce computer vision applications [3-5]. Therefore, effective low illumination image enhancement has very important value in transportation and application prospects [6,7].

To perfect the demerits of low illumination images, the image enhancement algorithm must improve the image contrast, enhance the brightness, and highlight the details in the image. The current low illumination image enhancement methods mainly have the following categories.

⋆ Corresponding Author

1. Based on the histogram equalization algorithm, the histogram of distribution unevenness is evenly distributed by mapping changes, and the image contrast is improved. Based on this, scholars put forward a lot of improved algorithms such as contrast restricted adaptive histogram equilibrium, platform histogram equalization and adaptive platform histogram equalization [8,9], etc. When these algorithms heighten the low-illumination image overall, there is a problem that the high frequency grayscale affects low frequency grayscale or local enhancement.
2. Retinex-based algorithm, the reflection component characterized by the image is captured by removing the illumination component in the raw image. Zhuang et al. [10] combined Sigmod-MSR with original images for data loss issues. Kong et al. [11] presented a Retinex image enhancement approach based on Gauss-based bilateral filtering. Al-Hashim et al. [12] proposed a new way based on directional full variation Retinex for noise and artifacts. These algorithms can adjust the grayscale distribution of the image, but cannot improve the image detail, and easily produce color bias and color distortion.
3. The image-enhancement algorithms based on deep learning. Through a huge database training model, the input test image is processed, and a wide range of applications have been obtained in recent years. Liu et al. [13] combined attention mechanisms with convolutional neural networks (CNN) to remove brightness components in Retinex theory. Li et al. [14] designed a convolution with encoding and procrastics neural networks, constructing a generation and counteracting model to quickly complete image enhancement. These algorithms require a large number of data sets to train and test model, and the data set of the current low illumination images is limited. The synthesized low-illumination images and actual images are still different.
4. The degenerative physical model is built upon the atmospheric scattering model. In 2020, Liu et al., [15] found the similarity of low illumination images and fog days, and brought indirectly to low illumination image enhancement by combining the prior theoretical theory of dark channels (DCP). Because it had no sufficient constraint, there was an excessive or too small problem in the estimation of the transmittance and atmospheric light value, and the enhancement results often had a flare artifact. In response to these problems, Feng et al. [16] proposed a transmitting transmission rate in rapidly oriented filtering. Yu et al [17] proposed a binding super-resolution neural network with guided filter repelling fog. Mei et al. [18] showed a method of combining a brightness communication diagram with dark channels. Guo et al. [19] combined dark channels for super-poxes to correct atmospheric light value and transmittance. These modified algorithms have improved the estimation of atmospheric light values and transmittance to a certain extent, but there are still some problems such as indefinite edge, flare artifact and image exposure.

Image enhancement methods based on histogram equalization focus on adjusting the histogram distribution of input images to display more visual information. The histogram represents the distribution of pixel intensity in the image. When the overall image is dark, the histogram distribution can be equalized to adjust the contrast of the image. Lee et al. [20] proposed a contrast enhancement algorithm based on hierarchical difference of two-dimensional histograms to enhance image contrast by amplifying gray difference between adjacent pixels. Lin et al. [21] put forward a contrast enhancement algorithm that carried out color channel stretching, histogram averaging and remapping in sequence. By

stretching color channels, color information in the scene could be better restored, histogram averaging was re-equalized to enhance image content, and histogram remapping was used to reduce artifacts that often occurred in equalization. In order to better recover image content, Sujee et al. [22] combined the idea of multi-resolution with histogram equalization, used pyramid technology to decompose low-illumination images, and enhanced histogram matching for images with different resolutions to extract information from images to the maximum extent. Although the low-illumination image enhancement algorithm based on histogram equalization can enhance the contrast by simply stretching the dynamic range of the image gray scale, it does not consider the influence of ambient light, so this kind of algorithm will have the problem of over-enhancement or under-enhancement.

Retinex theory is a color constant theory, the main idea is that the color perceived by the human eye is determined by the joint action of the light reflected by the surface of the target object and the ambient illumination. According to Retinex theory, low illumination image is the combination of normal illumination image and environmental illumination, and the image of normal illumination can be obtained by removing the influence of environmental illumination in the image. How to separate the ambient illumination from the low-illumination image is a pathological problem, which is usually solved by some prior and constraint conditions. Chen et al. [23] proposed a multi-scale Retinex enhancement algorithm with color restoration, which combined dynamic range compression, color consistency and human visual stability to restore image brightness and contrast. Meng et al. [24] proposed an effective low-light image enhancement algorithm. First, the low-light image was inverted to make it visually close to the de-fogging image, then the de-fogging algorithm was used to process the image, and the enhanced image was inverted again. Yadav et al. [25] used the illuminance estimation algorithm based on morphological approximation to decompose the image into reflection component and illuminance component, and then carried out global brightness enhancement and local contrast enhancement operations on the illuminance component respectively. Finally, multi-scale fusion was used to obtain the illuminance image and obtain the image of normal illuminance. Zhou et al. [26] built the initial illuminance map by looking for the maximum value of image pixels in the three color channels of red, green and blue, and then used a predetermined structure to refine the initial illuminance map to obtain the final illuminance map and further achieved image illumination enhancement. Inspired by human vision, Li et al. [27] proposed a multi-stage multi-exposure image fusion framework. In the first stage, exposure was adjusted by simulating human eyes to generate multi-exposure image sets. The second stage simulated the human brain mechanism to fuse the multi-exposure images to obtain the final enhanced image. The key of low illumination image enhancement method based on Retinex model is the accuracy of illumination image estimation, which depends on the precise definition of prior constraints. However, a prior condition is difficult to apply to image enhancement of various real scenes.

Convolutional neural networks have strong feature learning ability, and have become the main tool to solve computer vision task problems, and are also widely used in low-illumination image enhancement tasks. Lore et al. [28] first proposed an image enhancement framework based on deep learning. By designing a depth autoencoder to identify the features of low-illumination images, image brightness could be adaptive enhanced, image enhancement and noise reduction could be realized at the same time. Li et al. [29]

proposed a decomposition network based on Retinex model. Firstly, the input image was decomposed into illuminance map and reflection map, then denoised and enhanced the illuminance map and reflection map respectively, and finally reconstructed two components to obtain the normal illuminance image. Wang et al. [30] regarded low-illumination image enhancement as a residual learning problem, that is, the image after brightness enhancement was obtained by estimating the residual difference between low-illumination image and normal illumination image. Xu et al. [31] proposed a brightness enhancement network based on frequency decomposition, which first restored the content information of the low frequency layer under the condition of suppressing noise, and then enhanced the details of the high frequency layer of the image. Zhang et al. [32] proposed a brightness enhancement network based on Retinex model decomposition, which designed a mapping function to flexibly adjust the brightness according to the different needs of users. The network also included an adjustment module, which could effectively remove the visual defects amplified by enhancing the brightness of darker areas.

In response to the above-mentioned enhancement algorithms and to restore the true color of the image, to solve the problem of image exposure, flare artifact and edge loss, etc., and to improve the visual effect of the image, we propose a dual-channel combining light gradient domain weighted guided filtering low illuminance image enhancement method. Images are acquired by some robot equipment. First, the bright channel is introduced, and the atmospheric light value is obtained by using a threshold method, and then the transmittance is obtained by the dual channel image pixel level. After Retinex decomposition, the optical illumination map is used as the guide map, and the multi-scale gradient domain weighted guide filter operation to refine the transmittance. The denoising optimization is performed using BM3D. Finally, it restores the image according to the dehazing physical model.

2. Related Works

This enhancement algorithm is based on the atmospheric scattering debris model with a priority theory of dark channels, which is the light map obtained by Retinex decomposing in image filtering. It obtains a final reinforcing image after comprehensive denoising optimization.

2.1. Atmospheric Scattering Physical Model

The fog degradation model is based on a classic atmosphere scattering model. The given input image I to be removed from fog is:

$$I(x) = A\rho(x)exp[-\beta d(x)] + A(1 - exp[-\beta d(x)]). \tag{1}$$

Where $\rho(x)$ is density. $d(x)$ is the scene depth. β is the scattering coefficient of the optical wavelength. x is the pixel point position of the image. A is a global atmospheric value. The first term on the right of formula (1) is the direct attenuation term, and the second term is the atmospheric light component. Let the transmission medium transmittance be $t(x) = exp[-\beta d(x)]$, the well-exposed natural images be $J(x) = A\rho(x)$, so the atmospheric scattering physical model is simplified as:

$$I(x) = J(x)t(x) + A[1 - t(x)]. \tag{2}$$

In reference [33], an improved atmospheric scattering model (IASM) was proposed to overcome the inherent limitation of the traditional atmospheric scattering model. Based on the IASM, a fast single image dehazing algorithm was also presented. In this algorithm, by constructing a linear model between the transmission and the haze aware density feature, the transmission map could be directly estimated through a linear operation on three components: luminance, saturation and gradient.

2.2. Dark Channel Prior and Bright Channel Prior

He et al. [34] obtained the dark channel prior theory through the statistics of 5000 outdoor fog-free images. In most non-sky local region images, there is at least one color channel with a very small intensity value approaching to zero. It is described as:

$$J_{dark}(x) = \min_{c \in R,G,B} \min_{x \in \Omega(x)} [J_c(x)] \to 0. \tag{3}$$

Where J_{dark} is the dark primary color channel of the image. J_c is the intensity of red (R), green (G) and blue (B) color channels of image J. Ω is the local area centered on pixel point x with radius r.

The basic idea of bright channel prior is similar to dark channel prior. In most naturally sharp images, there is at least one color channel with a large intensity value approaching 255. It is described as:

$$J_{bright}(x) = \min_{c \in R,G,B} \max_{x \in \Omega(x)} [J_c(x)] \to 255. \tag{4}$$

Where J_{bright} is the bright primary color channel of the image.

Reference [35] proposed a dual channel prior-based method for nighttime low illumination image enhancement with a single image, which built upon two existing image priors: dark channel prior and bright channel prior. It utilized the bright channel prior to get an initial transmission estimate and then used the dark channel as a complementary channel to correct potentially erroneous transmission estimates attained from the bright channel prior.

2.3. Retinex Theory

Retinex theory is also known as the retinal cortex theory [36]. According to this theory, the observed color information of an object is determined by the reflective nature of the object and the intensity of the surrounding light. The reflection property determines the intrinsic property of the image and the illumination part determines the dynamic range of the image pixel. Its mathematical model is:

$$I(x, y) = R(x, y) \times L(x, y). \tag{5}$$

Where $I(x, y)$ is the observed image. $R(x, y)$ is a reflection map. $L(x, y)$ is an illuminated map. This model can be referred to as a multiplication model, and only $R(x, y)$ can be decomposed from $I(x, y)$ to obtain a representation of the essence of the image.

In reference [37], different networks were first combined to construct a multi-branch module for features extraction, and used the module and Retinex theory to extract the reflection map of the image. Then an attention mechanism was introduced into the multi-branch construction to balance the feature weight of each branch, and get the final result by the reconstruction module. The Retinex theory was used to calculate the L1 loss and the gradient loss for the intermediate feature map of the entire model to train our framework.

3. Proposed Low-illuminance Enhancement Algorithm

The new algorithm first inverts the low-illumination image I, and then defogs the fog map. The light and dark channels of the image are obtained, and the initial atmospheric light A_d and A_b obtained by the two channels are combined to calculate the final atmospheric light value A. Through the dark-bright channel coefficients t_d and t_b, combining with the dark-bright transmittance, the estimation of the initial transmittance t is obtained. Illumination gradient domain weighted guided filtering (IGDWGIF) is used to refine and smooth the initial transmittance. Finally, the atmospheric scattering model is used to restore the original image. Then the obtained defogging image is inverted. The BM3D is used for denoising, and the final enhanced image J is obtained. The flow chart is shown in figure 1.

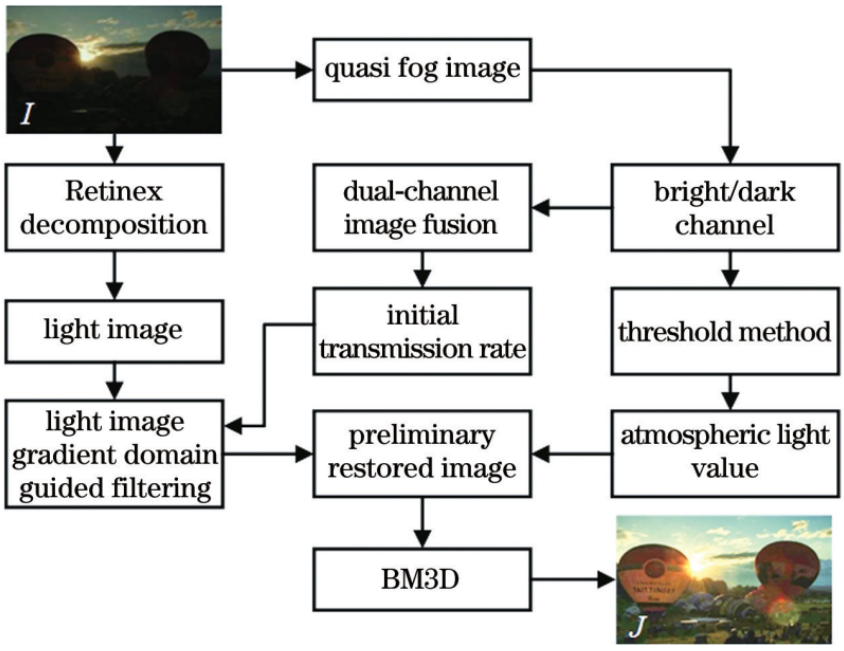


Fig. 1. The flow chart of the proposed low illumination enhancement algorithm

3.1. Threshold Method for Obtaining Atmospheric Light Value

For the calculation of global atmospheric light value A, if the brightest pixel of dark channel is directly used as atmospheric light value, there will be the problem of over-saturation. The atmospheric light value should be the radiation value at the depth of field infinity and the point should be in the background region of the image. Compared with the dark channel, the atmospheric light-dark density after merging with the bright channel is closer to the atmospheric light-dark density of the image to be enhanced. Therefore, the mean value of the pixel of the bright channel of the image to be enhanced can be used as the atmospheric light value. But some images completely contain sky or white objects, the results can be skewed.

To solve this problem, a threshold method is utilized to estimate atmospheric light value. Firstly, the light and shade profile of the image is judged, and the difference I_{dif} between the bright channel I_b and dark channel I_d is calculated:

$$I_{dif} = I_b(x) - I_d(x). \tag{6}$$

For various types of low illumination images, the control variable method is used to conduct several enhancement experiments, and the threshold value a is finally determined to be 0.38. If $I_{dif} > a$, it indicates that there is a large difference between the images of the two channels. Suspected points of atmospheric light with high pixel value are in the bright channel image. In order to avoid high values, the mean value of bright channel is used to estimate the atmospheric light value. If $I_{dif} \leq a$, it indicates that the suspected point of atmospheric light is on the dark object, and the atmospheric light value a obtained only by using the bright channel prior is not reliable. For these unreliable pixels, A_{cor} is obtained by combining dark channel prior images for correction, i.e,

$$A_{cor} = A_d \times A_b. \tag{7}$$

In summary, the atmospheric light value is:

$$A = \begin{cases} A_{bmean}, I_{dif} > a \\ A_{cor}, I_{dif} \leq a \end{cases} \tag{8}$$

Where A_{bmean} is a mean value of bright channel.

Figure 2 shows the enhanced effect diagrams of atmospheric light values by different methods. It can be seen that the enhanced image obtained by the new method in this paper is clearer, and the addition of bright channels makes the details in the dark more obvious. The restoration at the lamp source is more natural, but there are still issues of halo and unclear edges, which will be solved by improving the transmittance.

3.2. Two-channel Estimated Transmittance

Dark channel prior estimation of dark transmittance According to equation (3), the image to be defogged after filtering and inverting is defogged. Minimum filtering is performed on both ends of equation (3) when atmospheric light value A is known. It can obtain the following:



Fig. 2. Enhanced images of atmospheric light value via different methods. (a) Original image; (b) Reference [21]; (c) Proposed method

$$\min_{c\in R,G,B}\min_{x\in\Omega(x)} = t(x)\times\min_{c\in R,G,B}\min_{x\in\Omega(x)}[J_c(x)] + A_c[1-t(x)]. \tag{9}$$

Where A_c is the atmospheric light value of any color channel. I_c is any color channel of the image.

According to the dark channel prior theory, the dark channel of fog-free images tends to zero, so the first term on the right of equation (9) is 0. Meanwhile, in order to make the image look more natural and avoid excessive defogging, the defogging coefficient $m = 0.98$ is added to obtain the dark transmittance.

$$t_d(x) = 1 - m\times[\min_{c\in R,G,B}\min_{c\in R,G,B}[\frac{I_c(x)}{A_c}]]. \tag{10}$$

Bright channel prior estimation of bright transmittance

$$\max_{c\in R,G,B}\max_{x\in\Omega(x)}[I_c(x)] = t(x)\times\min_{c\in R,G,B}\min_{x\in\Omega(x)}[J_c(x)] + A_c[1-t(x)]. \tag{11}$$

According to the bright channel prior theory, the bright channel of fog-free images tends to 255. After image normalization, the pixel value of bright channel approaches to 1. The first term on the right of formula (11) is t, and the defogging coefficient m is also added, so the transmittance is:

$$t_b(x) = m\times\frac{\max_{c\in R,G,B}\max_{c\in\Omega(x)} - A_c}{1-A_c}. \tag{12}$$

Dual channel prior estimation of transmittance Although the initial transmittance estimated by dark channel prior has achieved good results, not all images have dark channel. For some low illumination images with dark channel prior not tending to 0 (for example, when there is a large bright area), other auxiliary conditions are required. In addition, using only dark channel windows will lead to the coexistence of different depth-of-field in the same local area, which will lead to the wrong classification of pixel points at the junction of light and dark or at the junction of far and near scenes, resulting in the deviation of the estimation of transmittance. Therefore, the depth of field problem of the same local area can be corrected by adding bright channel prior and combining the local windows of both regions, so as to solve the halo artifact problem caused by transmittance estimation deviation. In this paper, the dark and bright channel coefficients α and γ are used

to fuse the double channel, and then the transmittance is calculated. The enhancement effect can be better controlled by using certain linear relation. The specific relationship and constraint conditions are:

$$\begin{cases} t(x) = \alpha t_d(x) + \gamma t_b(x) \\ \alpha + \gamma = 1 \end{cases} \tag{13}$$

The transmittance will change as the values of α and γ change. Experiments are carried out on Figure 2 under different α and γ values. The experimental results are shown in Figure 3. It can be seen that the larger dark channel coefficient denotes the brighter transmittance image, the larger bright channel coefficient α and the darker transmittance image. When only dark channel is selected ($\alpha = 1$, $\gamma = 0$), there will be overexposure. When only the bright channel is selected ($\alpha = 10$, $\gamma = 1$), there will be a halo at the junction of the light, shade and the overall darkness. In order to make the overall light transparent and avoid overexposed, the effect is better when the dark channel coefficient is greater than the bright channel coefficient. Through a large number of low-illumination image enhancement experiments, the dark channel coefficient α is 0.75 and the bright channel coefficient γ is 0.25.



Fig. 3. Transmittance images under different α and γ. (a) $\alpha = 1$, $\gamma = 0$; (b) $\alpha = 0.75$, $\gamma = 0.25$; (c) $\alpha = 0.25$, $\gamma = 0.75$; (d) $\alpha = 0$, $\gamma = 1$

3.3. Illumination Graph Multi-scale Gradient Weighted Guided Filtering (IGDWGIF)

Low illuminance image has a serious loss of detail, so it has higher requirements in detail preservation and edge preservation [38]. Guided filtering [39] (GIF) can not only smooth the image, but also has a better effect on edge detail preservation than bilateral filtering and other filters, with lower computational cost. On this basis, a series of improved algorithms are derived. Among them, multi-scale gradient-guided filtering (GDGIF) introduces clear edge constraints and adds image gradient information, resulting in clearer processing effects. The guiding image of the guiding filter is important in the filtering process. The guiding image has a variety of choices, which can be the image itself or other images. At present, the image enhancement field uses a wide range of grayscale images to be enhanced, dark channel images, and RGB images to be enhanced. Good guided images make filtering results more structural.

Formula (1) is deformed so that $t(x) = exp[-\beta d(x)]$, and it can obtain:

$$1 - \frac{I(x)}{A} = t(x)[1 - \rho(x)]. \tag{14}$$

Defining $M(x) = 1 - \frac{I(x)}{A}$, $N(x) = 1 - \rho(x)$, it can obtain:

$$M(x) = t(x)N(x). \tag{15}$$

Where $N(x)$ is the inverse albedo. Formula (15) is defined as the multiplication model, which simply separates $t(x)$ and $N(x)$ from $M(x)$. This model is similar to Retinex algorithm. Retinex uses the filtering to decompose the light and reflection images. Here, illumination graph L is continuous and smooth in space. The transmittance $t(x)$ is a continuous function of depth and also remains smooth in space. They have similar properties. R is a reflection graph, which has the characteristics of large local variation and is similar to the inverse albedo $N(x)$. Based on this, this paper proposes a new guided filter, that is, illumination map multi-scale gradient weighted guided filter. Figure 4 is the edge information extraction images with different guidance graphs. Table 1 shows the similarity comparison between the original image and different guided images.

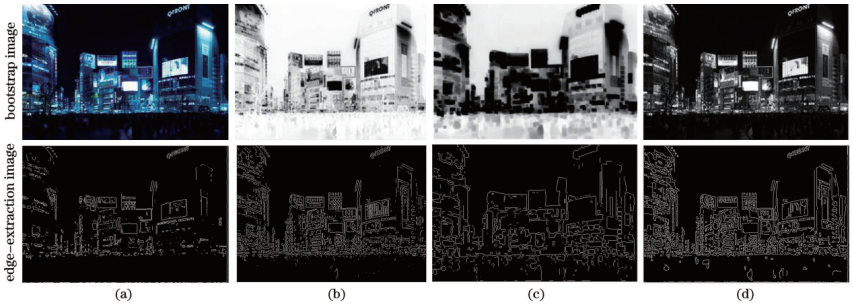


Fig. 4. Edge extraction images with different guided images (a) Original image (b) Gray-level inversion image (c) Dark-channel image (d) Illumination image

Table 1. Similarity between different guided images and original images

Guided image	Similarity
Gray-level inversion	0.2061
Illumination image	0.5677
Dark-channel	0.4052

In Figure 4, when the guided image adopts the image itself, filtering becomes an operation that only preserves edge and loses part of edge information. The gray scale inverse image is the pseudo fog image of low illuminance image under dark channel prior algorithm. Compared with the original image, the pixel value is higher, and the value is more than 200. In the chroma channel processing of the subsequent resulting image, the available edge details and brightness information are deviated, which makes the enhanced image overexposed and affects the color of the final enhanced image. When the guided image is used as a dark channel image, the pixel value is also high or low,

and the filtered image is blocky, so the details need to be improved. As a guided image, brightness image is better than gray image and dark channel image in both brightness and edge extraction. Compared with the original image, the brightness information of the image is richer and more consistent with the color characteristics of the object itself. It avoids too high brightness to affect the reduction effect. It enhances the detail information of dark area to some extent and is beneficial to image restoration.

From $t(x) = exp[-\beta d(x)]$, we can know that the transmittance t is a continuous function of depth of field d and concentration β. Dark channel prior is valid only when $t \to 1$. That is, when β is large, the prior condition is satisfied with $d \to 0$, otherwise, it is not. In order to make the vision also meet the prior conditions, the value of concentration β can be reduced and the method of two-time selective filtering can be adopted. That is, the concentration β is gradually reduced so that the prospect also satisfies the conditions of the prior theory. In summary, combining the properties of multi-scale gradient weighted guided filtering and brightness image, IGDWGIF enables the filtered image to better restore the low-illuminance image in terms of gradient information, texture information and color information.

3.4. Image Denoising Optimization Based on BM3D

Low illumination images generally have the problem of excessive noise, which has a great influence on image enhancement. Therefore, it is necessary to optimize the image denoising [40-43]. Through many experiments, if the denoising stage is carried out before enhancement, the original dark image will become more blurred and the image details will also be affected. Therefore, the optimization step of denoising is executed at the end of the algorithm, so as to achieve the purpose of optimizing and enhancing the image.

Most of the noise in low illuminance images are Gaussian noise generated during transmission and Poisson noise generated under low illuminance. Among them, the latter accounts for the majority. Therefore, block-matching and 3D filtering (BM3D) [44] is adopted for denoising. The BM3D mainly includes basic estimation and final denoising. The similarity block with the least difference [45] in the image is found and integrated into a three-dimensional matrix, which is transformed into a two-dimensional matrix to obtain the basic estimation. The denoising image is obtained by weighted average of the repeated values after filtering. When BM3D searches for reference pixel blocks, the step size is 4 pixel, and the complexity is reduced to 1/16 of that when the step size is 1, effectively shortening the computing time.

Here we conduct image denoising experiments with two methods BM3D and TWSC [46] to show the robust of BM3D. Result is shown in figure 5. The Gaussian noise is set to 35. In Figure 5, we find that at such a high noise level, the observed image is heavily degraded and many significant details are lost. SRR-A can effectively remove noise, but it causes the restored image to be over-smoothed and many significant details are lost. The image restored by BM3D has the most details and best overall appearance.

4. Experiments and Analysis

Here, 500 low-illuminance images are collected in this paper, including low-illuminance images at night. In this paper, three different enhancement models are selected to compare with the proposed method including NCTV [47], RFA [48] and FFTR [49].



(a) (b) (c)

Fig. 5. Experimental results with BM3D and TWSC. (a) Noisy image;(b) TWSC; (c) BM3D

4.1. Low Illumination Images with a Large Sky Area and a Wide Variation in Depth of Field

Low illumination image enhancement [50] with a large sky area and large variation of depth of field has always been one of the difficulties in research. The enhancement results of these images are compared and analyzed from two aspects: subjective evaluation and objective evaluation. The results are shown in Figure 6.

As can be seen from Figure 6(b1)-(b4), the enhancement results of the NCTV have halo artifacts. The white object is particularly obvious around, the contour of the restored object is not clear enough, and the overall image is dark. As shown in Figure 6(b2) and (b3), obvious halo can be seen, and the objects on the left side of the building in Figure 6(b1) and the hot air balloon on the ground in Figure 6(b4) are blurred with serious loss of details. As can be seen from Figure 6(cl)-(c4), the image brightness and contrast of the RFA are significantly improved compared with the NCTV algorithm. However, there is a certain over-exposure phenomenon, which is reflected in the sky region in Figure 6(c2) and Figure 6(c3). In Figure 6(c3), the boundary between the cloud and the sky is not obvious, and the circular corridor information of the object is lost. From Figure 6(d1)-(d4), it can be seen that the overall enhancement effect of the FFTR is relatively natural, and both brightness and contrast are improved. However, there is a certain color deviation phenomenon, especially in the building part of 6(d3) and the sky area of Figure 6(d4), the color distortion is obvious. The proposed algorithm gives more accurate estimations for atmospheric light value and transmittance. Therefore, it has a good effect in enhancing brightness and contrast and restoring sky region, as shown in Figure 6(e1)-(e4). The edge of the object is clear, the details are rich, there is no color distortion and halo phenomenon, the overall enhancement effect is more natural and clear. The visual effect is obviously better than other algorithms.

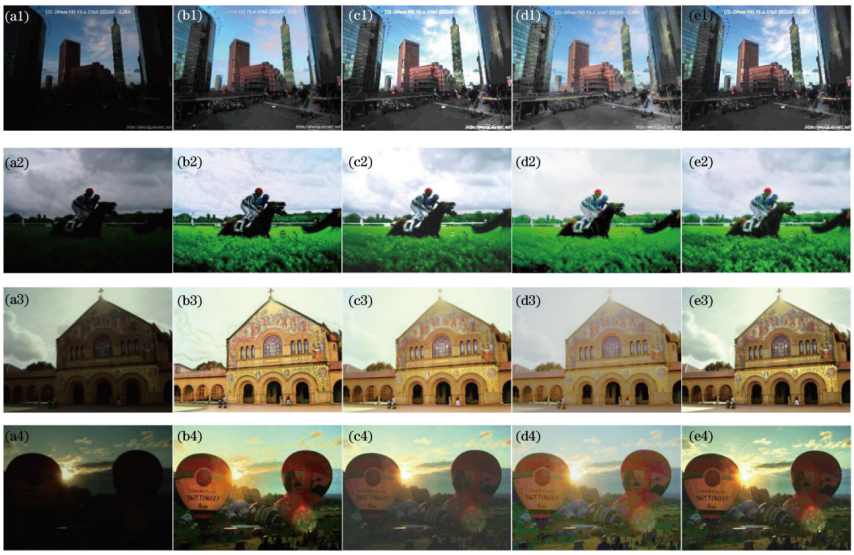


Fig. 6. Enhancement results of low-illumination images with large sky areas and obvious change in depth of field for different methods. (a1)-(a4) Original images; (b1)-(b4) NCTV method; (c1)-(c4) RFA method; (d1)-(d4) FFTR method; (e1)-(e4) proposed method

4.2. Low Illuminance Images with Light Source and Uneven Light

In the process of low illumination image enhancement with light source and non-uniform light, halo artifacts often appear. The enhancement results of such images are also compared, and the experimental results are shown in Figure 7.

As can be seen from Figure 7(b1)-(b3), no matter in indoor or outdoor light source, the enhancement results of the NCTV all have obvious halo phenomenon and the overall image is dark. In Figure 7(b1) and Figure 7(b3), the halo phenomenon can be clearly seen at the light source. In Figure 7(b2), the restoration of some small objects in the book is not clear enough, and a lot of image information is lost. The RFA algorithm is better than the NCTV in detail enhancement and halo artifact removal. However, obvious overexposure still exists in the light source and white objects, as reflected in the light in Figure 7(c1) and white objects in Figure 7(c2). The FFTR algorithm has good halo removal effect and dark area detail reduction effect in light source or white object, but there is color skew phenomenon and the overall color is lighter. As shown in Figure 7(d2), the restoration of dark area background is more complete with more details. However, the enhanced results of Figure 7(d1)-(d3) all have different degrees of color distortion. By combining dual channel and illumination guided filtering, the overall effect of the proposed algorithm is better than the previous algorithms in image restoration. As shown in Figure 7(e1) and Figure 7(e3), the enhancement at the light source is more natural. The image detail restoration in Figure 7(e2) is richer and more comprehensive, and the texture details of the image are clearer and the visual effect is better.



Fig. 7. Enhancement results of low-illumination images with light source and inhomogeneous light images with different methods. (a1)-(a3) Original images; (b1) (b3) NCTV method; (c1)-(c3) RFA method; (d1)-(d3) FFTR method; (e1)-(e3) proposed method

In order to compare the effect of low illumination image enhancement algorithm objectively and fairly, the evaluation indexes of image information entropy, average gradient and structure similarity are selected for comparative analysis.

Image information entropy represents the richness of image information. The higher entropy of image information denotes the image containing more information, and the richer image. It is represented as:

$$H = -\sum_{x=0}^{255} p_x lbp_x. \tag{16}$$

Where p_x is the occurrence probability of pixel point x. As shown in Table 2 (figure 8 is the visible result), the algorithm in this paper has a better overall effect and can make the enhanced image contain richer information. Moreover, the enhancement effect is natural and clear, and the color fidelity is better, which is consistent with subjective evaluation.

Table 2. Information entropy comparison

Original image	NCTV	RFA	FFTR	Proposed
Figure 6(a1)	7.3627	7.7676	7.6044	7.9873
Figure 6(a2)	6.8835	7.1563	7.2614	7.5858
Figure 6(a3)	7.6608	7.6252	7.2637	7.9068
Figure 6(a4)	6.9669	7.6201	7.2637	7.6299
Figure 7(a1)	6.4774	7.0699	7.1874	7.2673
Figure 7(a2)	5.6373	6.3463	6.3542	6.8835
Figure 7(a3)	6.1473	6.9075	6.8736	7.1347

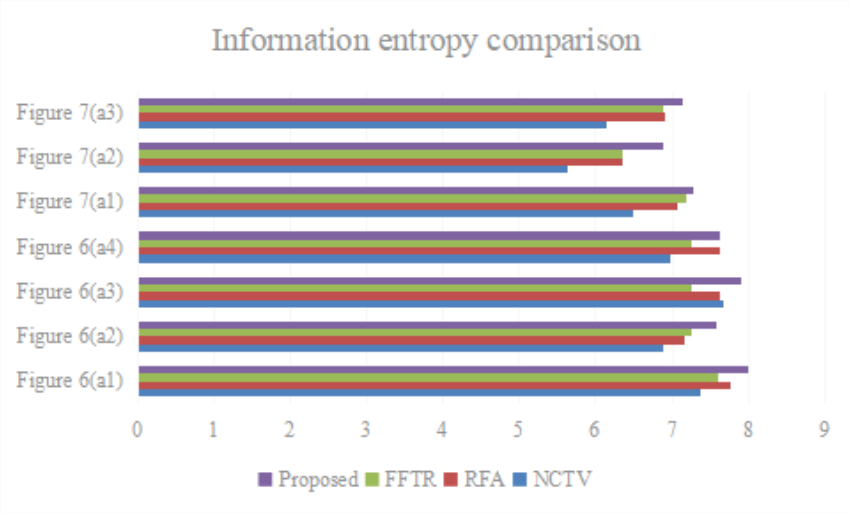


Fig. 8. Visualized result of Table 2.

The average gradient represents the detail information of the image. The larger average gradient denotes the clearer contour of the image and the richer detail information. It is represented as:

$$A = \frac{\sum_{i=1}^{w-1}\sum_{j=1}^{n-1}\sqrt{\frac{(J_{i,j}-J_{i+1,j})^2+(J_{i,j}-J_{i,j+1})^2}{2}}}{(w-1)(n-1)}. \tag{17}$$

Where $J_{i,j}$ is the pixel value of the $i-th$ row and $j-th$ column of the image matrix. w and n are the number of rows and columns of the image matrix respectively. As shown in Table 3 (figure 9 is the visible result), the average image gradient of the proposed algorithm is slightly lower than that of RFA in Figure 6(a1), and the overall average gradient performance is good. It can well maintain edges and restore details, enhance the image hierarchies, which is consistent with subjective evaluation.

Table 3. Average gradient comparison

Original image	NCTV	RFA	FFTR	Proposed
Figure 6(a1)	9.0718	11.9962	9.8472	10.1511
Figure 6(a2)	6.9855	6.5608	5.1720	9.0468
Figure 6(a3)	4.3488	5.4321	4.1092	6.3789
Figure 6(a4)	2.3238	2.2982	2.9927	3.2041
Figure 7(a1)	5.3746	6.9806	6.3413	7.1136
Figure 7(a2)	3.3274	5.8399	4.4367	5.9964
Figure 7(a3)	7.1267	7.9746	7.3422	8.3473

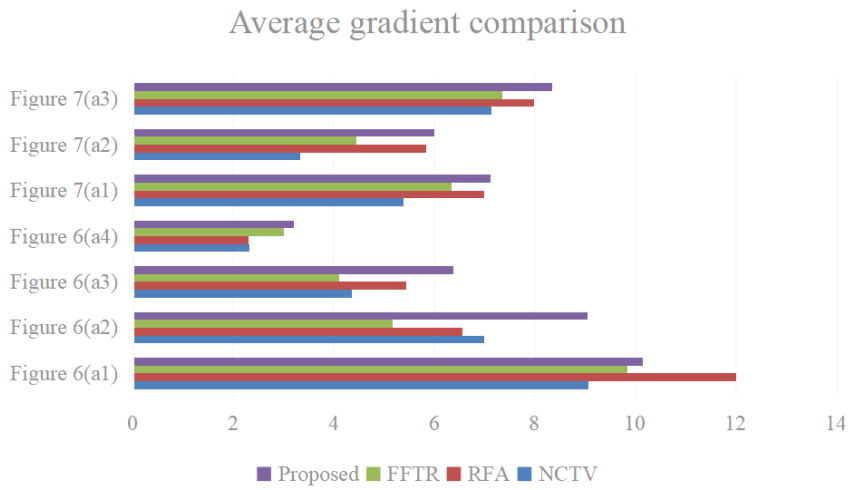


Fig. 9. Visualized result of table 3.

Structural similarity represents the similarity of content structure before and after image restoration. The value is close to 1, the quality of enhanced image is better. It is represented as:

$$S(X,Y) = \frac{(2\mu_X\mu_Y)(\sigma_{XY} + C_2)}{(\mu_X^2\mu_Y^2 + C_1)(\sigma_X^2 + \sigma_Y^2 + C_2)}. \tag{18}$$

Where $S(X,Y)$ is structural similarity. μ_X and μ_Y are the mean values of images X and Y respectively. σ_X and σ_Y are the standard deviations of image X and Y respectively. σ_{XY} is the covariance of images X and Y. C_1 and C_2 are constants. As shown in Table 4 (figure 10 is the visible result), the structure similarity of the proposed algorithm is closest to 1, and the image texture is clear without redundant halo artifacts and noise amplification.

Table 4. Structural similarity comparison

Original image	NCTV	RFA	FFTR	Proposed
Figure 6(a1)	0.6314	0.7046	0.6099	0.7344
Figure 6(a2)	0.5143	0.6322	0.5321	0.6945
Figure 6(a3)	0.7097	0.7636	0.6625	0.8069
Figure 6(a4)	0.6354	0.8093	0.7312	0.8144
Figure 7(a1)	0.6833	0.6919	0.5281	0.7221
Figure 7(a2)	0.6707	0.6228	0.5871	0.6974
Figure 7(a3)	0.5314	0.6372	0.4145	0.6738

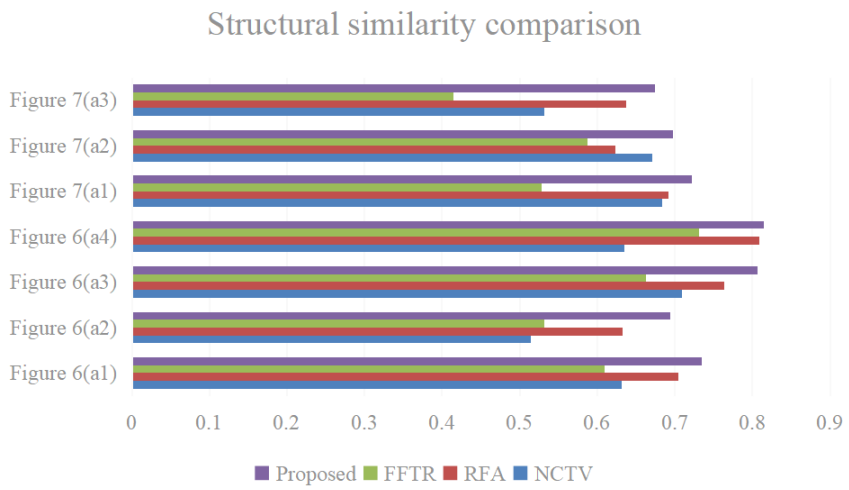


Fig. 10. Visualized result of table 4.

5. Conclusion

Aiming at the problems of halo artifact, brightness oversaturation, edge detail loss and noise amplification in the process of low illumination image enhancement by defogging physical model, an image enhancement method combining dual-channel prior and illumination gradient domain weighted guided filtering was proposed. The atmospheric light value is obtained by introducing broad value method of bright channel image, and the light channel priors are fused with dark channel priors to obtain the initial transmittance. The transmittance is refined by weighted guided filtering in illumination gradient domain. BM3D optimization denoising method is used to enhance the image, which solves the problem that dark channel prior is not suitable for the large bright areas and the same local area has different depth of field, making the estimation of atmospheric light value and transmittance more accurate. Results show that the new algorithm has obvious improvement in halo elimination, brightness adjustment, edge preservation and noise removal. Although many new methods have been proposed, in the future, we will use the most advanced deep learning methods to enhance images, in order to apply them to practical engineering applications.

References

1. He L, Long W, Liu S, et al. A night low-illumination image enhancement model based on small probability area filtering and lossless mapping enhancement[J]. IET Image Processing, 2021, 15(13): 3221-3238.
2. Che Aminudin M F, Suandi S A. Video surveillance image enhancement via a convolutional neural network and stacked denoising autoencoder[J]. Neural Computing and Applications, 2022: 1-17.
3. Wang X. Crowd density estimation based On multi-scale information fusion And matching network in scenic spots[J]. Journal of Applied Science and Engineering, 2022, 26(6): 863-873.

4. Wang J, Yang Y, Chen Y, et al. (2021) LighterGAN: An Illumination Enhancement Method for Urban UAV Imagery[J]. Remote Sensing, 13(7): 1371.
5. Shen W. (2022) A Novel Conditional Generative Adversarial Network Based On Graph Attention Network For Moving Image Denoising[J]. Journal of Applied Science and Engineering, 26(6): 831-841.
6. L. Bai, W. Zhang, X. Pan and C. Zhao, (2020) "Underwater Image Enhancement Based on Global and Local Equalization of Histogram and Dual-Image Multi-Scale Fusion," in IEEE Access, vol. 8, pp. 128973-128990, doi: 10.1109/ACCESS.2020.3009161.
7. Peng Y, Wang W, Tang Z, et al. (2022) Non-uniform illumination image enhancement for surface damage detection of wind turbine blades[J]. Mechanical Systems and Signal Processing, 170: 108797.
8. N. H. Saad, N. A. M. Isa and H. M. Saleh, (2021) "Nonlinear Exposure Intensity Based Modification Histogram Equalization for Non-Uniform Illumination Image Enhancement," in IEEE Access, vol. 9, pp. 93033-93061, doi: 10.1109/ACCESS.2021.3092643.
9. Liu S, Long W, He L, et al. Retinex-based fast algorithm for low-light image enhancement[J]. Entropy, 2021, 23(6): 746.
10. Zhuang P, Li C, Wu J. (2021) Bayesian retinex underwater image enhancement[J]. Engineering Applications of Artificial Intelligence, 101(1):104171.
11. X. -Y. Kong, L. Liu and Y. -S. Qian, (2021) "Low-Light Image Enhancement via Poisson Noise Aware Retinex Model," in IEEE Signal Processing Letters, vol. 28, pp. 1540-1544, doi: 10.1109/LSP.2021.3096160.
12. Al-Hashim M A, Al-Ameen Z. (2020) Retinex-Based Multiphase Algorithm for Low-Light Image Enhancement[J]. Traitement du Signal, 37(5):733-743.
13. Liu K, Tian Y. (2020) Research and analysis of deep learning image enhancement algorithm based on fractional differential[J]. Chaos, Solitons & Fractals, 131.
14. Li G, Yang Y, Qu X, et al. (2021) A deep learning based image enhancement approach for autonomous driving at night[J]. Knowledge-Based Systems, 213: 106617.
15. X Liu, Gao Z, Chen B M. (2020) IPMGAN: Integrating physical model and generative adversarial network for underwater image enhancement[J]. Neurocomputing, 453(3).
16. Feng X, J Li, Hua Z. (2020) Low-light image enhancement algorithm based on an atmospheric physical model[J]. Multimedia Tools and Applications, 79(3).
17. S. Yu and H. Zhu, (2019) "Low-Illumination Image Enhancement Algorithm Based on a Physical Lighting Model," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 29, no. 1, pp. 28-37, doi: 10.1109/TCSVT.2017.2763180.
18. Y. Mei and Y. Ning, (2019) "Multilayer Fusion and Chunk-Based Transmittance Estimation for Natural Hazy Image Enhancement," in IEEE Access, vol. 7, pp. 118269-118277, doi: 10.1109/ACCESS.2019.2937111.
19. Guo J, Li C, Guo C, et al. (2017) Research progress of underwater image enhancement and restoration methods[J]. Journal of Image and Graphics, 22(3), 273-287.
20. Lee C, Lee C, Kim C S. Contrast enhancement based on layered difference representation of 2D histograms[J]. IEEE transactions on image processing, 2013, 22(12): 5372-5384.
21. Lin S C F, Wong C Y, Rahman M A, et al. Image enhancement using the averaging histogram equalization (AVHEQ) approach for contrast improvement and brightness preservation[J]. Computers & Electrical Engineering, 2015, 46: 356-370.
22. Sujee R, Padmavathi S. Image enhancement through pyramid histogram matching[C]//2017 International Conference on Computer Communication and Informatics (ICCCI). IEEE, 2017: 1-5.
23. Chen B, Zhu L, Zhu H, et al. Gap-Closing Matters: Perceptual Quality Evaluation and Optimization of Low-Light Image Enhancement[J]. IEEE Transactions on Multimedia, 2023.
24. Meng X, Huang J, Li Z, et al. DedustGAN: Unpaired learning for image dedusting based on Retinex with GANs[J]. Expert Systems with Applications, 2024, 243: 122844.

25. Yadav G, Yadav D K, Mouli P C. Fusion-based backlit image enhancement and analysis of results using contrast measure and SSIM[M]//Digital Image Enhancement and Reconstruction. Academic Press, 2023: 235-251.
26. Zhou J, Liu Q, Jiang Q, et al. Underwater camera: Improving visual perception via adaptive dark pixel prior and color correction[J]. International Journal of Computer Vision, 2023: 1-19.
27. Li H, Wu X J. CrossFuse: A novel cross attention mechanism based infrared and visible image fusion approach[J]. Information Fusion, 2024, 103: 102147.
28. Lore K G, Akintayo A, Sarkar S. LLNet: A deep autoencoder approach to natural low-light image enhancement[J]. Pattern Recognition, 2017, 61: 650-662.
29. Li J, Hao S, Li T, et al. RDMA: low-light image enhancement based on retinex decomposition and multi-scale adjustment[J]. International Journal of Machine Learning and Cybernetics, 2023: 1-17.
30. Wang L W, Liu Z S, Siu W C, et al. Lightening network for low-light image enhancement[J]. IEEE Transactions on Image Processing, 2020, 29: 7984-7996.
31. Xu K, Yang X, Yin B, et al. Learning to restore low-light images via decomposition-and-enhancement[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 2281-2290.
32. Zhang Y, Zhang J, Guo X. Kindling the darkness: A practical low-light image enhancer[C]//Proceedings of the 27th ACM international conference on multimedia. 2019: 1632-1640.
33. Ju M, Gu Z, Zhang D. (2017) Single image haze removal based on the improved atmospheric scattering model[J]. Neurocomputing, 260: 180-191.
34. K. He, J. Sun and X. Tang, (2011) "Single Image Haze Removal Using Dark Channel Prior," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 33, no. 12, pp. 2341-2353, doi: 10.1109/TPAMI.2010.168.
35. Shi Z, Guo B, Zhao M, et al. (2018) Nighttime low illumination image enhancement with single image using bright/dark channel prior[J]. EURASIP Journal on Image and Video Processing, 2018(1): 1-15.
36. Jung C, Tian S, Jiao L. (2013) Eye detection under varying illumination using the retinex theory[J]. Neurocomputing, 113(aug.3):130-137.
37. Li M, Zhou D, Nie R, et al. (2021) AMBCR: Low-light image enhancement via attention guided multi-branch construction and Retinex theory[J]. IET Image Processing, 15(9): 2020-2038.
38. Wang, H., Su, Z., Liang, S. (2019). Structure-Preserving Guided Image Filtering. Intelligence Science and Big Data Engineering. Visual Data Engineering. IScIDE 2019. Lecture Notes in Computer Science, vol. 11935. Springer, Cham.
39. Wang L, Shoulin Y, Alyami H, et al. A novel deep learning-based single shot multibox detector model for object detection in optical remote sensing images[J]. Geoscience Data Journal, 2022. https://doi.org/10.1002/gdj3.162.
40. Feng D, Li T, Li G, et al. (2022) Reverse time migration of GPR data based on accurate velocity estimation and artifacts removal using total variation de-noising[J]. Journal of Applied Geophysics, 198:104563-.
41. Wei D. (2021) Retinex-Based Fast Algorithm for Low-Light Image Enhancement[J]. Entropy, 23.
42. Pan X, Li C, Pan Z, et al. (2022) Low-Light Image Enhancement Method Based on Retinex Theory by Improving Illumination Map[J]. Applied Sciences, 12(10): 5257.
43. Lu B, Pang Z, Gu Y, et al. (2022) Channel splitting attention network for low-light image enhancement[J]. IET Image Processing, 16.
44. K. Dabov, A. Foi, V. Katkovnik and K. Egiazarian, (2007) "Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering," in IEEE Transactions on Image Processing, vol. 16, no. 8, pp. 2080-2095, doi: 10.1109/TIP.2007.901238.
45. Zhang J, Wei Z. (2021) Image Enhancement Algorithm for Low Illumination LCD Screen[J]. Frontiers in Science and Engineering, 1(1): 1-8.

46. Xu J, Zhang L, Zhang D. (2018) A trilateral weighted sparse coding scheme for real-world image denoising[C]//Proceedings of the European conference on computer vision (ECCV). 20-36.
47. Kang M, Jung M. Simultaneous image enhancement and restoration with non-convex total variation[J]. Journal of Scientific Computing, 2021, 87(3): 83.
48. Xu Y, Sun B. (2022) A Novel Variational Model for Detail-Preserving Low-Illumination Image Enhancement[J]. Signal Processing, 195:108468-.
49. Kaur K, Jindal N, Singh K. Fractional Fourier Transform based Riesz fractional derivative approach for edge detection and its application in image enhancement[J]. Signal Processing, 2021, 180: 107852.
50. Yin S, Li H, Sun Y, et al. Data Visualization Analysis Based on Explainable Artificial Intelligence: A Survey[J]. IJLAI Transactions on Science and Engineering, 2024, 2(2): 13-20.

Su Chen female, Han nationality, born in Zhengzhou, Henan Province, Master, Lecturer, Mechanical and Electrical Engineering Department, Henan Vocational College of Water Conservancy and Environment, research interests: automatic control, image processing.

Dahai Li male, Han nationality, born in Zhengzhou, Henan Province, Master, Associate Professor, School of Electronics and Electrical Engineering, Zhengzhou University of Science and Technology, research interests: embedded technology development, image processing.

Received: March 14, 2024; Accepted: June 29, 2024.

Computer Science and Information Systems 21(4):1765–1781 https://doi.org/10.2298/CSIS240314057T

Underwater Image Denoising Based on Curved Wave Filtering and Two-dimensional Variational Mode Decomposition

Lin Teng[1], Yulong Qiao[1,⋆], and Shoulin Yin[1,2]

[1] School of Information and Communication Engineering, Harbin Engineering University
145 Nantong Street, Nangang District, Harbin City and 150001, China
qiaoyulong@hrbeu.edu.cn
[2] Software College, Shenyang Normal University
Shenyang, 110034 China
yslin@hit.edu.cn

Abstract. Underwater image denoising technology is of great significance in underwater operation. Underwater operations (such as offshore oil drilling, undersea tunnels, pipeline construction, underwater archaeology, biological research, and lifesaving) require stable and clear underwater images to aid analysis. Due to the scattering and absorption of light by water bodies, obtaining high-quality underwater images is a challenging task. Underwater images are prone to low contrast, low resolution and edge distortion. Therefore, it is difficult to accurately separate the effective signal when removing the underwater image noise, which leads to the image contrast reduction. Also the edge contour is not clear, and the detail loss is serious. Therefore, we propose a novel underwater image denoising method based on curved wave filter and two-dimensional variational mode decomposition. Firstly, the noisy image is decomposed by two-dimensional variational mode decomposition, and a series of modal components with different center frequencies are obtained. The effective modal components are selected by correlation coefficient and structural similarity. And the effective modal components are processed by the curve-wave filter. Finally, the filtered modal components are reconstructed to remove the noise in the image. The experimental results show that, compared with other state-of-the-art methods, the proposed method has clearer denoising results, less mean square error, and better denoising effect.

Keywords: Underwater image denoising, curved wave filter, two-dimensional variational mode decomposition, image reconstruction.

1. Introduction

With the rapid development of science and technology, human beings need to further develop Marine resources and understand the seabed organisms and available resources, which will be conducive to the study of Marine resources. As an auxiliary means of underwater vision, clear underwater images are the most intuitive way to understand the underwater world [1-3]. Although there is a high-quality imaging system for remotely operated underwater vehicles, due to the dispersion of water molecules, microorganisms in water,

⋆ Corresponding Author

and the quality of sensing elements in the camera, the underwater images collected will contain a lot of complex noise, which greatly reduces the visual quality of underwater images. It seriously affects the application of machine vision and late underwater images, and brings great challenges to the field of computer vision [4,5].

As a kind of water resources, Marine resources play a very important role in the development of human society and the earth's energy cycle [6]. Different from the terrestrial environment, the complexity, variability and dynamics of the Marine environment bring serious challenges to underwater exploration [7]. Underwater image denoising technology is an early means of ocean exploration, and plays an important role in seabed topography mapping, underwater target object recognition and detection, and biological population monitoring [8,9]. Due to the presence of scatterers in the ocean and the reverberation effect caused by the undulating seafloor and sea surface, the image has serious speckled noise, especially in the shallow sea area [10]. Speckled noise and interfering signals lead to degraded image quality, which adversely affects subsequent image processing. Therefore, the study of underwater image denoising method has always been the focus of research [11].

Classical image denoising methods include mean filter [12], median filter [13], wavelet transform [14], etc. Although the above denoising methods are simple in operation and have strong adaptability, there are unavoidable problems. Although mean filter is fast in calculation, it will damage the high frequency region of the image in the process of denoising, resulting in the loss of image details. Median filtering replaces the central pixel value of the window with the middle value after sliding window sorting, which can remove the noise in the high frequency part of the image. However, for the part of the image with relatively dense noise, the denoising effect of median filtering will be weakened, and the image will become blurred with the increase of sliding window. These methods need to adjust the filter suitable for the noise characteristics according to the local noise, which is faced with many difficulties. In the wavelet transform, the selection of the threshold will affect the denoising effect of the image. In addition, different noise images need to set different thresholds.

Different threshold functions in the curve-wave threshold denoising framework [15] produce different denoising effects. The soft threshold and hard threshold functions described in references [16,17] had been widely used, but they had different degrees of defects. The discontinuity of the curve wave hard threshold function at the threshold point caused obvious visual distortion such as ringing and pseudo-Gibbs effect in de-noised images. There is a constant deviation between the curve coefficients before and after the soft threshold function processing, which leads to the blur of the edge of the image and the decrease of the visual quality of the de-noised image. Therefore, the construction of an efficient threshold function is the key to improve the de-noising performance of the curve-wave threshold de-noising method [18]. In addition, the curve-wave threshold denoising method can only ensure that the noise whose absolute value is less than the threshold value is removed, and there is no unified standard for the selection of threshold values. Surround effect will occur, resulting in poor image visual quality [19]. In view of these inherent performance defects of the wavelet threshold denoising method, researchers will use Partial Differential Equation (PDE)[20] based on information fusion to improve the denoising effect of the wavelet threshold method [21].

In 2014, Dragomiretskiy et al. [22] proposed a completely non-recursive variational mode decomposition model (VMD), which could adaptively decompose the signal into a set of band-limited inherent mode functions and obtain the optimal value of the inherent mode function, thus solving problems such as the sensitivity of empirical mode decomposition to noise and sampling. In order to facilitate the processing of two-Dimensional signals, Dragomiretskiy et al. [23] proposed a two-dimensional variational mode decomposition model (2D-VMD) based on the VMD algorithm. The model was a natural two-dimensional extension in the context of image segmentation and orientation, and was a non-recursive, fully adaptive variational method that could decompose an image into a series of sub-modal components with different center frequencies. Many scholars have applied the 2D-VMD model to different areas of image processing. Zhang et al. [24] proposed a denoising method for medical ultrasound images based on two-dimensional variational mode decomposition combined with fast non-local mean, which could retain edge information well while denoising, and had a good denoising effect in high noise variance. Meriem et al. [25] used two-dimensional variational mode decomposition to reduce the random noise of fringe patterns and improve contour and residual images, thereby controlling and reducing the existence of defect flatness on the surface. Although some denoising algorithms based on two-dimensional variational mode decomposition have been proposed successively, no more effective denoising methods have been proposed according to the characteristics of underwater images.

Compared with other filtering methods, curve-wave filtering has a better effect on preserving image edge information and details in the process of denoising. Therefore, an underwater image denoising method combining wave-wave and two-dimensional variational mode decomposition is proposed in this paper. Firstly, the noisy image is decomposed into a series of modal components with different center frequencies by two-dimensional variational mode decomposition. Then, the correlation coefficient (CC) and structural similarity (SSIM) are used to select the effective modal components, and the effective modal components are processed by curvewave filtering. Finally, the filtered modal components are reconstructed to remove the noise in the image.

This paper has the following organization structure. In section 2, we introduce the Fast Discrete Curvelet Transform (FDCT). Section 3 detailed shows the proposed underwater image denoising. Experiments are conducted in section 4. There is a conclusion in section 5.

2. Fast Discrete Curvelet Transform (FDCT)

The Curvelet Transform (CT) [26,27] is decomposed on all possible scales using the Multi-scale Ridgelet Transform, the implementation of CT can be summarized as follows. I is the given dimension $N \times N$ two-dimensional digital image signal, and its signal is decomposed by $J - th$ layer binary wavelet, it can obtain:

$$I = C_j + \sum_{j=1}^{J} D_j \tag{1}$$

Where C_j is the low frequency component on the coarsest scale. $D_{J\,J=1,2,\cdots,J}$ is the high frequency. The size value of $j = 1, 2, \cdots, J$ represents the thickness of the scale. The CT transformation process is shown in Figure 1.

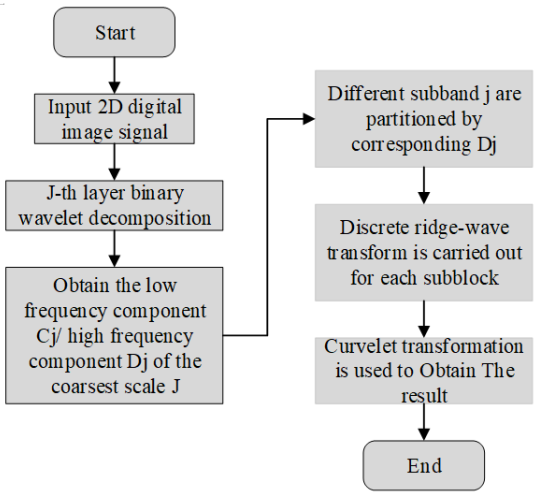


Fig. 1. The process of CT

Table 1. Approximation order under different transformations

Class	Approximation order
Fourier	$\|\|g - \bar{g}_m\|\|_2^2 = o(m^{-0.5})$
Wavelet	$\|\|g - \bar{g}_m\|\|_2^2 = o(m^{-1})$
Curvelet	$\|\|g - \bar{g}_m\|\|_2^2 = o(m^{-2})$

CT is a non-adaptive multi-scale geometric analysis method. Different from the adaptive method, CT does not need to know the geometric features of the image in priori, but does directly decompose the signal on a set of fixed frames or basis functions, and makes full use of the information of the function itself. It can use different kinds of function transformation for different parts of the image, so that the corresponding function types can achieve the optimal approximation, and provide a powerful "sparse" description of the image. Obviously, compared with the curve described by wavelet transform, CT breaks through the limitation of Govitch anisotropy of wavelet "sparse" expression signal [28]. The following theorem gives an optimal explanation of the Curvelet transformation.

Let $f \in W_2^2(R^2)$ and $g(x) = f(x)_{x_2 \leq \tau(x_1)}$. If the curve τ satisfies the second derivative, then the error order of the M term of the function g CT can be achieved by nonlinear approximation $Q_M^2(g)$ is:

$$||g - Q_M^C||_2^2 \leq CM^2(logM)^{0.5} \tag{2}$$

For second-order differentiable functions, it can be seen from $(logM)^{0.5}$ term that CT has almost reached the optimal approximation order, and its optimal approximation order is $O(M^{-2})$. The nonlinear wavelet transform approximation error decays at M^{-1} order. Table 1 shows the approximation order of function g under three different transformations (Fourier, Wavelet, Curvelet).

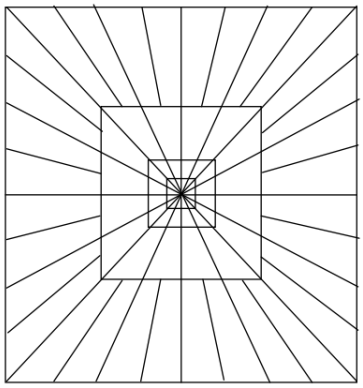


Fig. 2. Scale and Angle segmentation of discrete Curvelet transform

In order to realize the Curvelet transformation of two-dimensional discrete images without changing the construction frame of Curvelet, it is necessary to replace the ring wedge window function U_j with the square window function in two-dimensional Cartesian coordinate system, and use the center square region to describe the frequency domain segmentation of the discrete Curvelet transformation [29], as shown in Figure 2.

In a two-dimensional Cartesian coordinate system, we set the local window to $\tilde{W}_j(\varpi) = \tilde{V}_j(\varpi)U_j(\varpi)$, where $j \geq 0$, and,

$$\tilde{V}_j(\varpi) = \sqrt{\varphi_{j+1}^2 - \varphi_j^2} \tag{3}$$

$$U_j(\varpi) = U(2^{j/2}\varpi_2/\varpi_1) \tag{4}$$

Where, φ is the inner product of two one-dimensional low-pass Windows.

$$\varpi_j(\varphi_1, \varphi_2) = \phi(2^{-j}\varphi_1)\phi(2^{-j}\varphi_2) \tag{5}$$

With the frequency domain segmentation diagram above, it can be seen that the intervals of the curved wave transform segmentation are the same, and each interval has the same slope $\tan\theta_1$, and $\tan\theta_1 = l \cdot 2^{-l/2}$, where $l = -2^{-j/2}, \cdots, -2^{-j/2} - 1$, then,

$$\tilde{W}_{j,l}(\varpi) = V_j(\varpi)U_j(\Gamma_{\theta_l}\varpi) \tag{6}$$

Where, the shear moment Γ_{θ_l} is:

$$\begin{pmatrix} 1 & 0 \\ -\tan\theta & 1 \end{pmatrix} \tag{7}$$

Then a discrete Curvelet can be defined as:

$$\bar{\Re}_{j,l,k}(x) = 2^{3j/4}\bar{\Re}_j(\Gamma_{\theta_l}^T(x - \Gamma_{\theta_l}^{-T}b)) \tag{8}$$

$$b = (k_1 \cdot 2^{-j/2}, k_1 \cdot 2^{-j/2}) \tag{9}$$

The definition of the Curvelet transform is given by the following formula:

$$c(j,l,k) = \int \hat{g}(\varpi)\tilde{W}_j(T_{\theta_l}^{-1})e^{i(T_{\theta_l}^{-1}b,\varpi)}d\varpi \tag{10}$$

In this paper, a simple and fast Unequally Spaced Fast Fourier Transform (USFFT) in CT transform is used for image filtering. The process of USFFT is shown in **Algorithm 1**.

Algorithm 1 USFFT

1: Step 1: Transform the function $g[t_1, t_2] \in L^2(R)$ by 2DFFT Fourier transform to obtain the Fourier sampling set $\hat{g}[n_1, n_2]$, where $-n/2 \leq n_1, n_2 \leq n/2$;
2: Step 2: In the frequency domain, re-sampling $\hat{g}[n_1, n_2]$ on all scale and angular direction parameters (a, b) by interpolation method, get $\hat{g}[n_1, n_2 - n_1 \tan\theta_b]$, where $(n_1, n_2) \in P_a$;
3: Step 3: Multiply $\hat{g}[n_1, n_2 - n_1 \tan\theta_b]$ with the fitting parabolic window $\tilde{U}[n_1, n_2]$ and localize to get $\hat{g}[n_1, n_2] = \hat{g}[n_1, n_2 - n_1 \tan\theta_b] \times \tilde{U}[n_1, n_2]$. Among them, the fitting window size satisfies $width = length^2$;
4: Step 4: Transform the localization $\hat{g}[n_1, n_2]$ into a 2DIFFT inverse Fourier transform to obtain the Curvelet transformation coefficient $c^D(a, b, k)$;

3. Proposed Underwater Image Denoising

The variational mode decomposition (VMD) algorithm can decompose the signal into a set of component modes with specific direction and oscillation characteristics. These intrinsic mode functions can accurately reconstruct a given input signal while limiting each mode to an online estimated center frequency. Based on the successful application of VMD algorithm in one-dimensional signal decomposition, researchers naturally extended VMD algorithm on two-dimensional signals and proposed two-dimensional variational mode decomposition, which was more suitable for two-dimensional image decomposition. Compared with variational mode decomposition, two-dimensional variational mode decomposition minimizes the bandwidth of the sub-signal while maintaining data fidelity [30].

In one-dimensional analytic signals, the signal is analyzed through a single side spectrum, so the negative frequency needs to be set to zero. Generalizing a one-dimensional analytic signal to a two-dimensional analytic signal requires effectively setting to zero half of the plane of the frequency domain, which corresponds to a vector denoted w_k. Therefore, the two-dimensional analytic signal is defined as follows in the frequency domain:

$$D_{AS,k}(w) = \begin{cases} 2u_k(w) & (w, w_k) > 0 \\ u_k(w) & (w, w_k) = 0 \\ 0 & (w, w_k) < 0 \end{cases} \tag{11}$$

$$\begin{aligned} u_{AS,k}(x) = u_k(x) * (\delta(< x, w_k >) + \frac{j}{\pi < x, w_k >}) \\ \times \delta(< x, w_{k,\perp} >) \end{aligned} \tag{12}$$

Here, $*$ denotes convolution, transformations are separable.

The analysis signal w_k is computed linearly along the reference direction and is processed independently, so the definition is one-dimensional in nature, but has the desired two-dimensional Fourier properties.

The function of 2D-VMD is determined according to the components of generalized variational mode decomposition. The formula for the minimization of the function is as follows:

$$\begin{aligned} &\min_{u_k w_k} \sum_k \alpha_k ||\nabla[u_{AS,k}(x) e^{-j<w_k,x>}]||_2^2 \\ &s.t. \forall x : \sum_k u_k(x) = f(x) \end{aligned} \tag{13}$$

The constraints are reconstructed by quadratic punishment and augmenting Lagrange function, and optimized by multiplication operator alternate direction method (ADMM) [14].

As with the one-dimensional analytic signal, the u_k of the two-dimensional analytic signal is optimized:

$$\begin{aligned} u_k^{n+1} = &\underset{u_k}{argmin}\, \alpha_k ||\nabla[u_{AS,k}(x) e^{-j<w_k,x>}]||_2^2 \\ &+ \underset{u_k}{argmin}\, ||f(x) - \sum_i u_i(x) + \lambda(x)/2||_2^2 \end{aligned} \tag{14}$$

The following Wiener filter result will be produced:

$$\begin{aligned} u_k^{n+1}(w) = &(f(w) - \sum_{i \neq k} u_i(w) + \lambda(w)/2) \\ &\times \frac{1}{1 + 2\alpha_k |w - w_k|^2} \end{aligned} \tag{15}$$

Where $\forall w \in \Omega_k : \Omega_k = w|(w, w_k) \geq 0$. The center frequency w_k is slightly simpler to optimize, and its update goal is:

$$w_k^{n+1} = \underset{w_k}{argmin}\, \alpha_k ||\nabla[u_{AS,k}(x) e^{-j<w_k,x>}]||_2^2 \tag{16}$$

The first moment of the power spectrum $|u_k(w)|^2$ on the half plane Ω_k model is:

$$w_k^{n+1} = \frac{\int_{\Omega_k} w |u_k(w)|^2 dw}{\int_{\Omega_k} |u_k(w)|^2 dw} \tag{17}$$

According to equations (14) and (16), K intrinsic modal components of the two-dimensional analytic signal are obtained.

Before the inverse Fourier transformation results are obtained, data needs to be constantly updated in the frequency domain. The specific process of the 2D-VMD can be described as in **Algorithm 2**:

Therefore, the flowchart of proposed underwater image denoising is shown in figure 3. The denoising algorithm proposed in this paper is shown in **Algorithm 3**.

Algorithm 2 2D-VMD

Step 1: Initializing the parameters w_k^0, u_k^0, λ^0, and $n = 0$;

Step 2: Set the iteration $n = n + 1$ and update u_k, w_k. The u_k and w_k are updated as follows:

$$w_k^{n+1} = \frac{\int_{\Re^2} w|u_{AS,k}^{n+1}(w)|^2 dw}{\int_{\Re^2} |u_{AS,k}^{n+1}(w)|^2 dw} \tag{18}$$

$$u_k^{n+1}(x) = \Re F^{-1} u_{AS,k}^{n+1}(w) \tag{19}$$

Step 3: Update multiplication operator λ as follows:

$$\lambda^{n+1}(w) = \lambda^n(w) + \tau(f(w) - \sum_k u_k^{n+1}(w)) \tag{20}$$

τ indicates the step update coefficient.

Step 4: Determine whether the component satisfies the constraint condition $\frac{\sum_k ||u_k^{n+1} - u_k^n||_2^2}{||u_k^n||_2^2} < K_\epsilon$. If satisfy it, stop iteration; If not, go back to Step 2) and continue iterating;

Algorithm 3 Proposed denoising algorithm

Step 1: A series of IMF images with different center frequency modal components are obtained by using 2D-VMD algorithm;

Step 2: The correlation coefficient and structural similarity are used to screen the series of modal components, and the modal components are divided into effective IMF image components and ineffective IMF image components;

Step 3: The image component of the effective IMF is filtered by the curvilinear filter, and the effective modal component after filtering is obtained;

Step 4: The effective modal components after filtering are reconstructed and the denoised images are obtained;

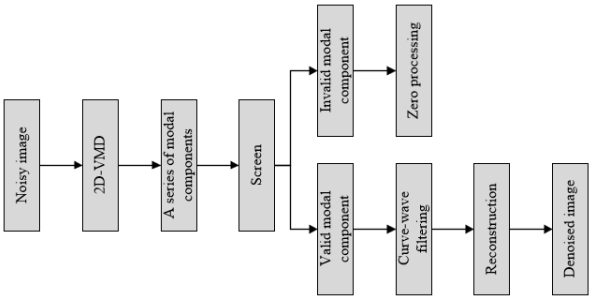


Fig. 3. The whole process of denoising.

4. Experiments and Analysis

Some underwater images are randomly selected from the dataset UIEB [31] as experimental images, as shown in Figure 4. The peak signal-to-noise ratio (PSNR) [32] and structural similarity (SSIM) [33] are used as measurement indexes. The experimental environment is Windows11, CPU i510400F, and memory 16GB, 3th Gen Intel(R) Core(TM) i5-13400F 2.50 GHz. At the same time, we compare with the reference TVM [34], WTM [35] and MSRG [36].

PSNR, as an objective criterion for image evaluation, mainly calculates the degree of distortion between the processed image and the real image. The larger value denotes that the denoised image is more similar to the real underwater image. The calculation of MSE and PSNR is shown in equation (21) and equation (22):

$$MSE(x,y) = \frac{1}{M \times N} \sum_{i=1}^{H} \sum_{j=1}^{W} (W(i,j) - Y(i,j))^2 \tag{21}$$

$$PSNR(x,y) = 10 \times lg(\frac{M_{MAX}^2}{MSE(x,y)}) \tag{22}$$

Where MAX is the maximum value of color image pixel. $MSE(x,y)$ is the mean square error between the real underwater image and the noised image. (i,j) is the pixel value at $(ij) - th$ position in the image, $i,j = 1,2,\cdots,N$.

As a full-reference image quality evaluation index, SSIM measures image similarity from three aspects: brightness, contrast and structure. SSIM is calculated as shown in equation (23):

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma y^2 + c_2)} \tag{23}$$

Where μ_x and μ_y are the average values of x and y, respectively. σ_x and σ_y are the standard deviations of x and y, respectively. σ_{xy} is the covariance of x and y. c_1 and c_2 are positive constants. The larger SSIM value denotes the smaller the gap between the processed underwater image and the undistorted image, and the better denoising effect.

We first add three different Gaussian noises to the original images, i.e. Gaussian noise=0.05,0.4,0.8. The results are shown in figure 5.

For the results obtained by the four denoising methods, the PSNR and SSIM are calculated respectively, and the calculation results are shown in Table 2.

Table 2. Objective index calculation results

Method	PSNR	SSIM
TVM	20.1613	0.4719
WTM	21.9935	0.5147
MSRG	23.9431	0.5627
Proposed	20.2637	0.5907

From table 2, we can see that the PSRN and SSIM values of proposed method are 20.2367 and 0.5907 respectively. Its results are much higher than MSRG, WTM and TVM. Through our proposed method, different index results under different Gaussian noise are obtained as shown in tables 3-6. Meanwhile, the visualizations are also presented as shown in figures 6-9.

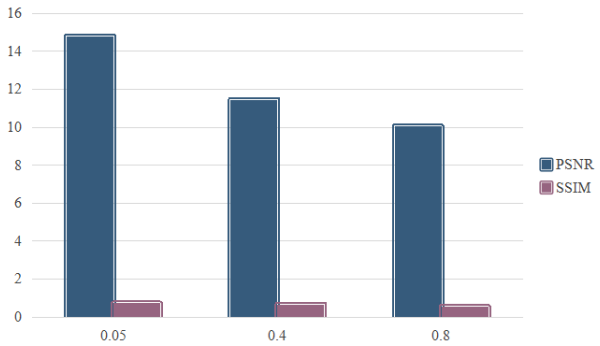


Fig. 4. PSNR and SSIM for image1

Figures 10-13 are the underwater image denoising results under noise with 0.05. Figures 14-17 are the underwater image denoising results under noise with 0.4.

From the above experimental results, it can be seen that, on the whole, the proposed method in this paper has a good effect on noise reduction. No matter it is the submarine tourists in Figure 10 or the fish in Figure 11, there are obvious clear signs.

5. Conclusions

In this paper, the underwater image denoising method is studied, and an image denoising method based on curved wave filtering and 2D-VMD is proposed. Aiming at the problem that the edge of underwater image denoising is not clear and detail loss is serious, the 2D-VMD algorithm is used to achieve effective image decomposition, and the correlation

Table 3. Results of image2

Gaussian noise	PSNR	SSIM
0.05	13.8217	0.6693
0.4	12.7385	0.5895
0.8	9.4398	0.5467

Table 4. Results of image4

Gaussian noise	PSNR	SSIM
0.05	13.8172	0.8217
0.4	12.2525	0.7835
0.8	10.1654	0.6315

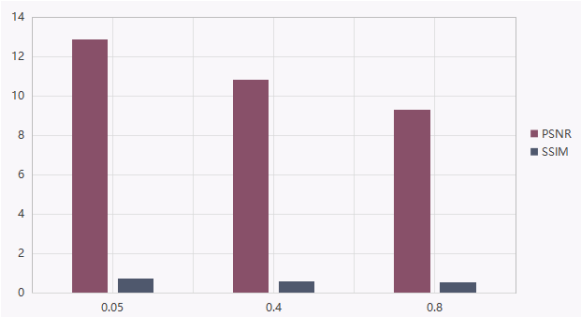


Fig. 5. PSNR and SSIM for image3

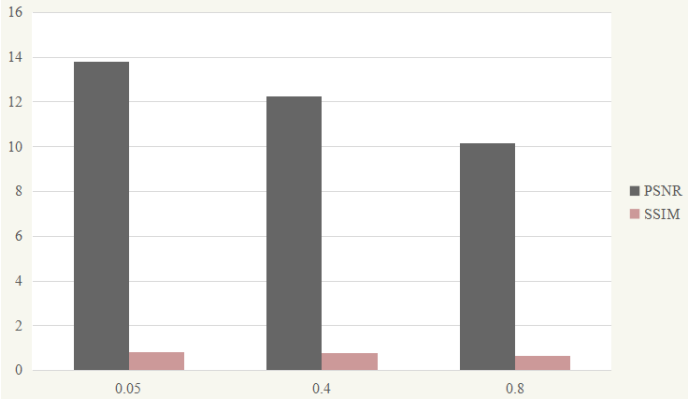


Fig. 6. PSNR and SSIM for image4

coefficient and structural similarity are introduced to further screen the effective modal components and then the curve-wave filtering is used to obtain the denoised images with higher structural similarity, brightness, contrast, edge restoration degree and detail preservation. Effectively improve the image quality. Through experiments, the minimum peak

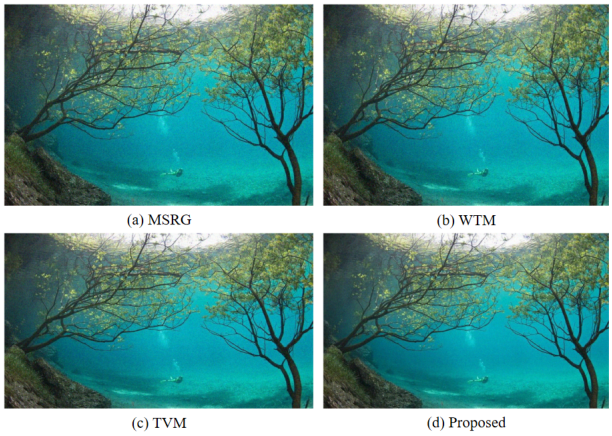


Fig. 7. Denoising comparison results for image1 under noise=0.05

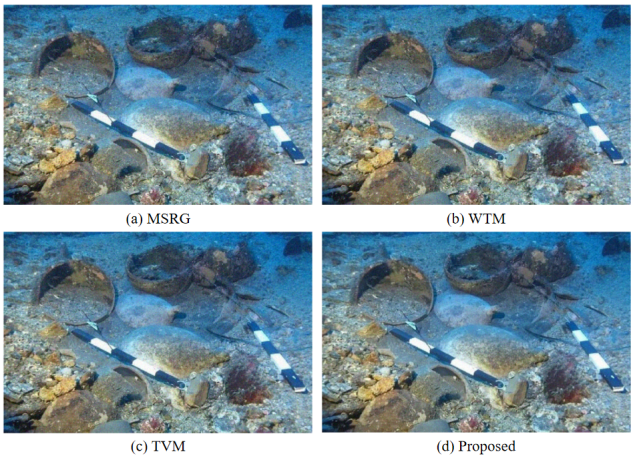


Fig. 8. Denoising comparison results for image3 under noise=0.05

signal-to-noise ratio (PSNR) which can be processed by the algorithm is determined. The denoising of low quality images can satisfy most of the complex conditions and provide help for the subsequent image processing tasks.

Acknowledgments. Authors are grateful for the anonymous review by the review experts.

References

1. Brondi S, Neresini F, Sciandra A. The social representation of nanotechnologies and its relationships with those of science and technology: Making familiar the unfamiliar between enthusiasm and caution[J]. Journal of Risk Research, 2022, 25(1): 113-137.

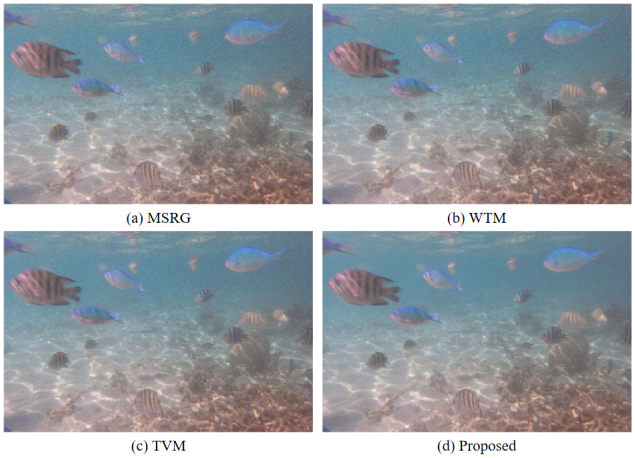


Fig. 9. Denoising comparison results for image4 under noise=0.05

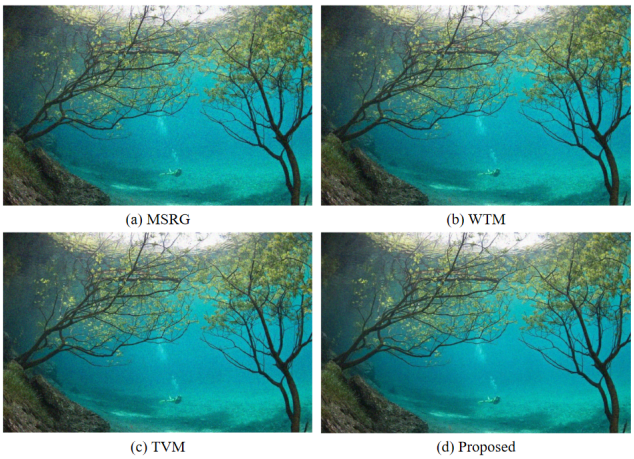


Fig. 10. Denoising comparison results for image1 under noise=0.4

2. He X, Ping Q, Hu W. Does digital technology promote the sustainable development of the marine equipment manufacturing industry in China?[J]. Marine policy, 2022(Feb.): 136. DOI:10.1016/j.marpol.2021.104868.
3. Wang M, Chu Y, Qiang L, et al. Rapid, amplification-free and high-throughput SARS-CoV-2 RNA detection via a reduced-graphene-oxide based fluorescence assay[J]. Sensors & Diagnostics, 2022, 1(2): 262-269.
4. Sun X, Liu L, Dong J. Underwater image enhancement with encoding-decoding deep CNN networks[C]//2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI).IEEE, 2017. DOI:10.1109/UIC-ATC.2017.8397462.

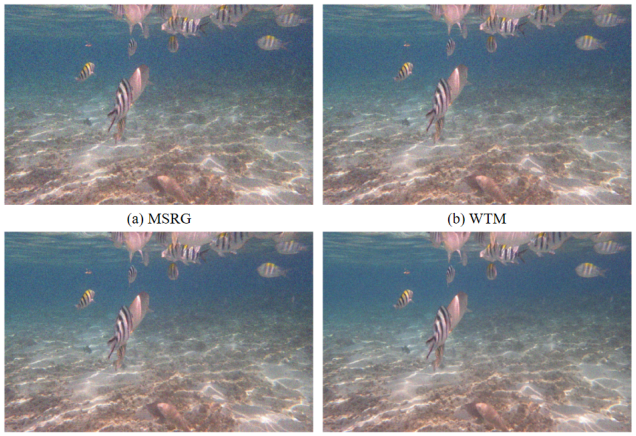


Fig. 11. Denoising comparison results for image2 under noise=0.4

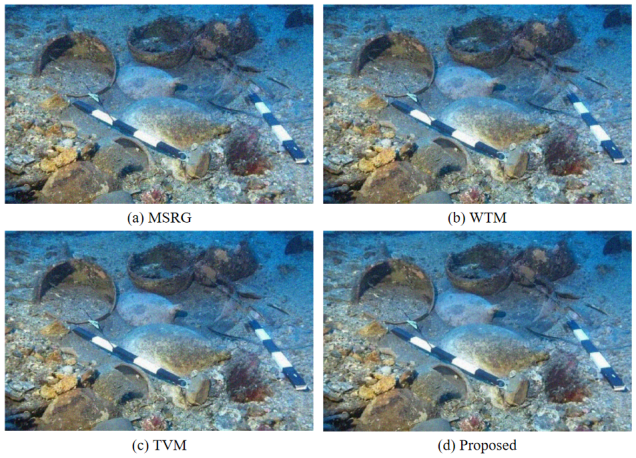


Fig. 12. Denoising comparison results for image3 under noise=0.4

5. Tang Z, Li J, Huang J, et al. Multi-scale convolution underwater image restoration network [J]. Machine Vision and Applications. vol. 33, no. 85 (2022). https://doi.org/10.1007/s00138-022-01337-3
6. Wang S, Li W, Xing L. A review on marine economics and management: How to exploit the ocean well[J]. Water, 2022, 14(17): 2626.
7. Sun Y, Yuan C, Yan X. Theoretical model research on IV characteristics of solar cell under the marine environment[C]//2015 International Conference on Transportation Information and Safety (ICTIS). IEEE, 2015: 877-882.
8. Durden J M, Schoening T, Althaus F, et al. Perspectives in visual imaging for marine biology and ecology: from acquisition to understanding[M]//Oceanography and Marine Biology. CRC Press, 2016: 9-80.

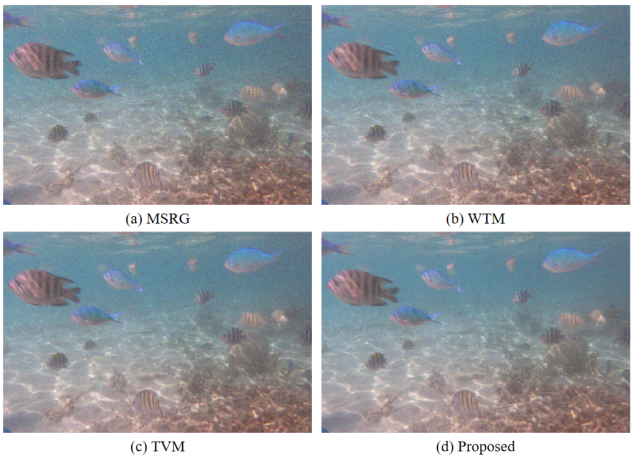


Fig. 13. Denoising comparison results for image4 under noise=0.4

9. Yuan X, Guo L, Luo C, et al. A survey of target detection and recognition methods in underwater turbid areas[J]. Applied Sciences, 2022, 12(10): 4898.
10. L. Teng et al., "FLPK-BiSeNet: Federated Learning Based on Priori Knowledge and Bilateral Segmentation Network for Image Edge Extraction," in IEEE Transactions on Network and Service Management, 2023. doi: 10.1109/TNSM.2023.3273991.
11. Kocak D M, Caimi F M. The current art of underwater imaging - with a glimpse of the past and vision of the future[J]. Marine Technology Society Journal, 2005, 39(3): 5-26.
12. Thanh D N H, Engi?no?lu S. An iterative mean filter for image denoising[J]. IEEE Access, 2019, 7: 167847-167859.
13. Kumar N, Dahiya A K, Kumar K. Modified median filter for image denoising[J]. International Journal of Advanced Science and Technology, 2020, 29(4s): 1495-1502.
14. Teng Lin, Hang Li and Shoulin Yin. Modified Pyramid Dual Tree Direction Filter-based Image De-noising via Curvature Scale and Non-local mean multi-Grade remnant multi-Grade Remnant Filter [J]. International Journal of Communication Systems. vol. 31, no. 16,?November 10, pp. e.3486.1-e.3486.12, 2018.
15. Guo G. Real-time medical image denoising and information hiding model based on deep wavelet multiscale autonomous unmanned analysis[J]. Soft Computing, 2023, 27(7): 4263-4278.
16. Wang L, Xiao D, Hou W S, et al. A nonlocal enhanced Low-Rank tensor approximation framework for 3D Magnetic Resonance image denoising[J]. Biomedical Signal Processing and Control, 2022, 72: 103302.
17. Onur T $\ddot{O}$. Improved image denoising using wavelet edge detection based on Otsu's thresholding[J]. Acta Polytechnica Hungarica, 2022, 19(2): 79-92.
18. F. Wang, B. Yang, Y. Wang and M. Wang. Learning From Noisy Data: An Unsupervised Random Denoising Method for Seismic Data Using Model-Based Deep Learning[J]. IEEE Transactions on Geoscience and Remote Sensing, vol. 60, pp. 1-14, 2022, Art no. 5913314, doi: 10.1109/TGRS.2022.3165037.
19. Mu Q, Wang X, Wei Y, et al. Low and non-uniform illumination color image enhancement using weighted guided image filtering[J]. Computational Visual Media, 2021, 7: 529-546.
20. Y. Sun, S. Yin and H. Li. A New Wavelet Threshold Function Based on Gaussian Kernel Function for Image De-noising[J]. Journal of Information Hiding and Multimedia Signal Processing, Vol. 10, No. 1, pp. 91-101, January 2019.

21. Feng L, Li J, Li C, et al. A Blind Source Separation Method Using Denoising Strategy Based on ICEEMDAN and Improved Wavelet Threshold[J]. Mathematical Problems in Engineering, 2022, 2022.
22. K. Dragomiretskiy and D. Zosso. Variational Mode Decomposition[J]. IEEE Transactions on Signal Processing, vol. 62, no. 3, pp. 531-544, Feb.1, 2014, doi: 10.1109/TSP.2013.2288675.
23. Zosso D, Dragomiretskiy K, Bertozzi, A.L. et al. Two-Dimensional Compact Variational Mode Decomposition[J]. Journal of Mathematical Imaging and Vision, 58, 294-320 (2017). https://doi.org/10.1007/s10851-017-0710-z
24. Zhang X, Gu H, Zhou L, et al. Improved dual-domain filtering and threshold function denoising method for ultrasound images based on non-subsampled contourlet transform[J]. Journal of Medical Imaging and Health Informatics, 2017, 7(7): 1624-1628.
25. Messagier M, Meguellati S, Mahgoun H. Fringe pattern denoising using two-dimensional variational mode decomposition (2D-VMD) method for inspection of flatness of reduced surfaces[J]. Experimental Techniques, 2022: 1-15.
26. Khaldi A, Kafi M R, Moad M S. Wrapping based curvelet transform approach for ECG watermarking in telemedicine application[J]. Biomedical Signal Processing and Control, 2022, 75: 103540.
27. Nagaraja Kumar N, Jayachandra Prasad T, Prasad K S. An intelligent multimodal medical image fusion model based on improved fast discrete curvelet transform and Type-2 fuzzy entropy[J]. International Journal of Fuzzy Systems, 2023, 25(1): 96-117.
28. You Q, Trzasko J D, Lowerison M R, et al. Curvelet transform-based sparsity promoting algorithm for fast ultrasound localization microscopy[J]. IEEE transactions on medical imaging, 2022, 41(9): 2385-2398.
29. Cywińska M, Trusiak M, Zuo C, et al. Enhancing single-shot fringe pattern phase demodulation using advanced variational image decomposition[J]. Journal of Optics, 2019, 21(4): 045702.
30. Li Y, He D, Li Y. Underwater visual image denoising based on total variation method[C]//Conference on Infrared, Millimeter, Terahertz Waves and Applications (IMT2022). SPIE, 2023, 12565: 709-714.
31. Li C, Guo C, Ren W, et al. An underwater image enhancement benchmark dataset and beyond[J]. IEEE Transactions on Image Processing, 2019, 29: 4376-4389.
32. C. Li, S. Lian, J. Niu, C. Wang and X. Zhou, "Research on Underwater Image Denoising Based on Wavelet Threshold Method," 2022 7th International Conference on Intelligent Computing and Signal Processing (ICSP), Xi'an, China, 2022, pp. 1941-1947, doi: 10.1109/ICSP54964.2022.9778825.
33. Zhou J, Yao J, Zhang W, et al. Multi-scale retinex-based adaptive gray-scale transformation method for underwater image enhancement[J]. Multimedia Tools and Applications, 2022: 1-21.
34. Li Y, He D, Li Y. Underwater visual image denoising based on total variation method[C]//Conference on Infrared, Millimeter, Terahertz Waves and Applications (IMT2022). SPIE, 2023, 12565: 709-714.
35. Li C, Lian S, Niu J, et al. Research on Underwater Image Denoising Based on Wavelet Threshold Method[C]//2022 7th International Conference on Intelligent Computing and Signal Processing (ICSP). IEEE, 2022: 1941-1947.
36. Zhou J, Yao J, Zhang W, et al. Multi-scale retinex-based adaptive gray-scale transformation method for underwater image enhancement[J]. Multimedia Tools and Applications, 2022: 1-21.

Yulong Qiao was born in 1978. He received the Doctoral degree in engineering from the Harbin Institute of Technology in 2006. In recent years, he has published more than 50 papers in domestic and foreign journals and important conferences, among which more than

20 papers were indexed by SCI, and more than 30 were indexed by EI. His research interests include signal representation and analysis, statistical image processing, image/video processing and applications, and texture analysis and applications. He has won two science and technology awards of Heilongjiang Province. He is a Senior Member of the Chinese Institute of Electronics and IEEE Signal Processing Branch.

Lin Teng received the M.A. degree from Shenyang Normal University, Shenyang, China, in 2020. She is currently pursuing the Ph.D. degree with the College of Information and Communication Engineering, Harbin Engineering University, Harbin. Her research interests are image processing and semantic segmentation.

Shoulin Yin received the M.A. degree from Shenyang Normal University, Shenyang, China, in 2015. He is currently pursuing the Ph.D. degree with the College of Information and Communication Engineering, Harbin Engineering University, Harbin. His research interests are remote sensing image processing and object detection.

Received: March 14, 2024; Accepted: June 29, 2024.

Computer Science and Information Systems 21(4):1783–1800 https://doi.org/10.2298/CSIS240418062J

A Novel Deep Fully Convolutional Encoder-Decoder Network and Similarity Analysis for English Education Text Event Clustering Analysis

Zhenping Jing

School of Foreign Languages, Zhengzhou University of Science and Technology
Zhengzhou 450064 China
jingzpz@163.com

Abstract. Education event clustering for social media aims to achieve short text clustering according to event characteristics in online social networks. Traditional text event clustering has the problem of poor classification results and large computation. Therefore, we propose a novel deep fully convolutional encoder-decoder network and similarity analysis for English education text event clustering analysis in online social networks. At the encoder end, the features of text events are extracted step by step through the convolution operation of the convolution layer. The background noise is suppressed layer by layer while the target feature representation is obtained. The decoder end and the encoder end are symmetrical in structure. In the decoder end, the high-level feature representation obtained by the encoder end is deconvolved and up-sampled to recover the target event layer by layer. Based on the linear model, text similarity is calculated and incremental clustering is performed. In order to verify the effectiveness of the English education text event analysis method based on the proposed approach, it is compared with other advanced methods. Experiments show that the performance of the proposed method is better than that of the benchmark model.

Keywords: Online social networks, Text event clustering, Deep fully convolutional encoder-decoder network, Similarity analysis, Linear model.

1. Introduction

In today's network era, with the development of the mobile Internet, information interaction has become unprecedentedly simple and fast. QQ, WeChat, Weibo, Douyin, Kuaishou and other social media have widely entered people's lives and changed people's living habits. Studies have shown that social media is more sensitive to new events than traditional media [1-3]. Therefore, it is of great significance to conduct data analysis on educational texts in social media. Among them, event clustering is an important step of event detection in social media education [4,5].

Event clustering aims to cluster text according to different features of text event features. Social media is mostly short text, and the text content is diverse and random, including more disturbing words. The traditional unsupervised clustering model is difficult to accurately extract the event features of social text, and the accuracy of event clustering results is low. Xu et al. [6] used a supervised deep neural network model to cluster social texts, which enhanced the clustering effect. However, faced with massive social

texts, this method required a lot of text annotation work. Costache et al. [7] proposed a Logit model to estimate weights instead of using weights determined entirely by distance. The nearest neighbor was automatically selected using a selection process such as lasso or enhancement. Then, based on the concepts of evaluation and selection, the predictor space was expanded. Balogun et al. [8] proposed an adaptive ASMC (Adaptive Selection of Misprediction Classifiers) method, which dynamically selected a classifier based on the characteristics of the class to better predict the error tendency of a class from a set of machine learning classifiers. An empirical study of 30 software systems showed that ASMC performed better than five classifiers used alone and combined with majority vote integration [9].

At present, most event clustering studies are mainly based on word characteristics in online social networks. Different from long text, short text clustering has the problem of high dimensional sparsity [10], so some scholars consider introducing external features. Among them, Ma et al. [11] used emergent keywords to predict the importance of short texts and cluster these important short text detection events. By considering the release time, diffusion degree and diffusion sensitivity of events, Hompes et al. [12] adopted time characteristics to detect events and performed clustering. In addition, Adamopoulos et al. [13] explored the influence of users on social media data, using text content characteristics, user characteristics and usage characteristics to detect events and perform clustering. Liu et al. [14] made use of named entity features of text to enhance the effect of event detection.

In order to solve the problem of high-dimensional sparsity in traditional methods, Yang et al. [15] mapped high-dimensional text to low-dimensional semantic space through local save index (LPI), and made semantically related texts close to each other in low-dimensional space. Rezaee et al. [16] used a K-means clustering algorithm to cluster feature word sets to obtain feature clusters, and used feature clusters to represent sentence vectors, thus solving the problem of dimension explosion of vector space models and improving the clustering effect. Lilleberg et al. [17] proposed JS-IDF sequence for text embedding based on Word2Vec word vector and combined with temporal relationship. Vodrahalli et al. [18] proposed the SIF Embedding of the text, which was weighted average of the word vector and modified by PCA and SVD to obtain the low-dimensional vector representation of the text. Soni et al. [19] adopted a deep neural network based on DCNN to learn the deep feature representation of text. The model first obtained the binary encoding of the text by the existing unsupervised dimensionality reduction method, and then inputted the text into the convolutional neural network through Word Embedding (WE). The binary coding of the text was taken as the training object of the model, and the intermediate feature vector between the convolutional layer and the output layer was taken as the depth feature of the text, which was a self-training based unsupervised model.

For streaming data in social media education, common clustering algorithms include Singlepass incremental algorithm and local sensitive hash (LSH) algorithm [20]. To obtain a new sample, the Singlepass clustering algorithm first needs to calculate the similarity between the new text and the existing event. If the similarity exceeds the threshold, it will be added to the existing event with the greatest similarity; otherwise, it will be set as a new event. The key step of the algorithm is to calculate the text similarity, the most commonly used is cosine similarity cosine. The local sensitive hash (LSH) clustering algorithm is mainly based on the new event detection model. Its idea is to find the nearest neighbor

text set of the new text in the clustered text through the LSH algorithm, and find the nearest neighbor text of the new text from the set. If the maximum similarity between the two is greater than the set threshold, it is an existing event; otherwise, it is a new event. In the LSH clustering algorithm, the key step is to find the nearest neighbor text of the new text by LSH algorithm as soon as possible. Zhao et al. [21] improved the hashing algorithm for finding the nearest neighbor text, which improved the efficiency, but the accuracy was comparable to Petrovic [22]. Enguehard et al. [23] proposed a deep embedded clustering model based on self-training. This model used deep neural networks to learn both feature representation and cluster assignment, and was a partition-based clustering model, which was not suitable for processing streaming data. Zhao et al. [24] used SIF Embedding to represent the sentence and used auto-encoder to extract the low-dimensional feature representation of the text. The clustering algorithm could learn both the text feature representation and cluster assignment through the self-trained neural network model to obtain the clustering result.

Pavlinek et al. [25] used the supervised SVM model to determine whether two texts were related, and conducted text clustering, which regarded the sample distribution as a graph model and realized clustering by maximizing the similarity of sample pairs within the cluster. Zhang et al. [26] used the supervised LSTM model to extract text features, calculate text similarity, and adopt the incremental clustering algorithm to cluster social text. Compared with previous clustering algorithms, the performance was improved, but a large amount of data was needed to train the model. At present, the method of fully deep convolutional neural network has not been used in the event clustering method for social education media.

Therefore, we propose a novel deep fully convolutional encoder-decoder network and similarity analysis for Chinese education text event clustering analysis in online social networks in this paper. At the encoder end, the features of text events are extracted step by step through the convolution operation of the convolution layer. Based on the linear model, text similarity is calculated and incremental clustering is performed. Experiments show that the performance of the proposed method is better than that of the benchmark model. The rest of the paper is organized as follows. The deep fully convolutional encoder-decoder network and proposed social media education events clustering model are described in Sections 2 and 3. In Section 4, we conduct experiments for this proposed model. Finally, Section 5 concludes the paper.

2. Deep Fully Convolutional Encoder-decoder Network (DFCEDN)

2.1. Convolutional Encoder-decoder Network

Encoder-decoder network is a flexible framework model, composed of encoders and decoders, which can be flexibly applied to unsupervised or supervised tasks as required, usually for network pre-training, high dimensional raw data dimensionalization, feature extraction and data compression and generation. Encoders and decoders can choose different network models such as FCN, CNN and RNN according to different tasks. In this paper, a deep fully convolutional encoder-decoder network is used to enhance text features by using the local perception and weight sharing of CNN [27,28]. The traditional neural network splices the time-frequency features of adjacent frames into a long vector

as the input of the network. The change of relative position will increase the difficulty of the network to model the correlation structure in the text. Convolutional neural networks have local perception characteristics, and each neuron in the network only needs to be connected with some neurons in the previous layer. Meanwhile, the weight sharing feature of convolutional neural networks makes each filter feature map have the same weight, which will greatly reduce the number of parameters in the network and improve the robustness of the model.

The decoder end is composed of multiple convolutional layers, each of which consists of convolutional filtering, batch standardization, pooling and nonlinear transformation operations. The structure of the decoding end corresponds to that of the encoding end, which is composed of deconvolution, up-sampling, batch standardization and nonlinear transformation operation. This structure is applied to speech enhancement, and the time-frequency features of noisy speech are taken as the input of the network. The convolution network is used to model the local typical structure of noisy speech spectrum at the coding end, extract high-level speech features, and suppress the influence of background noise layer by layer. At the decoding end, the detailed speech components are recovered layer by layer from the high-level speech feature information extracted from the coding end through the deconvolution layer, and the speech signal is reconstructed.

2.2. Skip Connection

The skip connection is first introduced in neural network to solve the problem that the gradient disappears with the increase of network depth in training, which leads to the difficulty of network training. In this paper, skip connection is introduced between corresponding layers of encoder. On the one hand, the error can be directly transmitted to the lower layer, so that the training of the network is more effective, and an important role is that the details of the signal can be compensated. With the increase of the number of network layers, the main structural features of the signal will be extracted due to the operation of layer by layer, such as convolution and pooling, and some local details will be lost. Especially with the deepening of the number of network layers, it may cause the problem of recovering the details of the target signal from high-level information that cannot be obtained by the decoding end. The feature map of the coding side contains a lot of detail information of the signal. By introducing skip connections to transfer the feature map information of the coding side from the convolution layer to the corresponding deconvolution layer, it can help to recover the detail information of the target signal.

3. The Proposed Social Media Education Events Clustering Model

The clustering process of the proposed model is shown in Figure 1. For the input social media text t_i, firstly, the event cluster is carried out to determine whether t_i belongs to an existing event or a new event, or cannot determine. If t_i belongs to an existing event, it is added to the cluster. If it is a new event, a new cluster is created based on ; otherwise, t_i is set as an uncertain sample and Buffer is added. In this process, a deep fully convolutional encoder-decoder network (DFCEDN) is used to extract text features, and a linear model is used to calculate text similarity. Since the incremental clustering algorithm can obtain real-time clustering results, after each part of the data is clustered, samples are extracted

from the clustering results to form a training set to retrain the model, so that the model can further learn new event features and enhance the clustering effect. At the end of the clustering, all samples in clusters with fewer elements in the clustering result are set as uncertain samples. Then, the model after repeated retraining is used to cluster the uncertain samples, and the final clustering result is obtained. The clustering process is described in detail below.

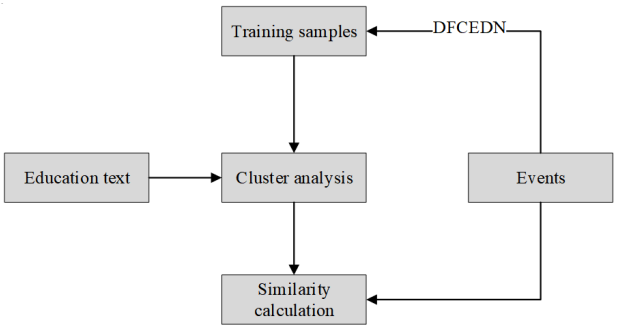


Fig. 1. Proposed education text clustering process

3.1. Text Representation

In this paper, DFCEDN model is used to extract text features, which is denoted as $M_{encoder}$. Before the text input, the first word is segmented and stopped words. The preprocessed text is represented by $X = w_1, w_2, \cdots, w_n$. Where w_i represents the number of the $i-th$ word in the sentence in the thesaurus. n represents the number of words in the sentence. The Chinese word vector pre-trained from the education entry of Baidu Encyclopedia is used to embed the sentence, and the d-dimensional word vector x_i is used to represent the words $w_i, x_i \in R^{n \times d}$. The text vector $X = x_1, x_2, \cdots, x_n$ gets a hidden sequence $h_1, h_2, \cdots, h_n$ through the DFCEDN model. Where h_t is calculated from the current input vector x_t and the output h_{t-1} of the previous time, $t \in (1, n)$, i.e.,

$$h_t = DFCEDN(x_t, h_{t-1}). \tag{1}$$

The initial parameters of the model are generated randomly, and we use a low-dimensional vector h_t to represent the text vector X.

3.2. Text Similarity Calculation

Vector similarity is a measure of the degree of similarity between vectors. If two vectors have the same starting point and the same ending point, then the two vectors are completely similar, and the similarity is equal to 1. At present, there are many researches on vector similarity, which are mainly divided into three categories. Reference [29] studied the similarity of vectors from the perspective of vector direction; Reference [30] studied

the similarity of vectors from the perspective of vector size; Reference [31] studied the similarity of vectors from the perspective of vector direction and magnitude. Since a vector contains two elements: magnitude and direction, the similarity of a vector should be measured from two aspects: direction and magnitude. It is feasible to measure the similarity between two vectors from the aspects of direction and magnitude, but the position information should also be considered when measuring the similarity between multiple vectors. Therefore, we give a method to measure the similarity between multiple vectors.

The Angle cosine method is an important method to measure the similarity between two vectors, but it has a certain one-sidedness. Taking a two-dimensional vector as an example, the cosine of the Angle between vector (1,1) and vector (3,3) is equal to the cosine of the Angle between vector (1,1) and vector (0.1,0.1), but we cannot consider vector (3,3) and vector (0.1,0.1) to be exactly equivalent. Since a vector is a vector containing two elements of magnitude and direction, we should measure the similarity between two vectors in terms of direction and magnitude.

Let the vector $A = a_1, a_2, \cdots, a_n$ and vector $B = b_1, b_2, \cdots, b_n$, then the similarity between vector A and vector B can be defined as:

$$\lambda = \frac{min(||A||, ||B||)}{max(||A||, ||B||)} \cdot \frac{[A, B]}{||A|| \cdot ||B||}. \tag{2}$$

Where $||A||$ and $||B||$ represent the magnitude of vector A and vector B, respectively. $[A, B]$ represents the dot product of vector A and vector B.

When measuring the similarity between multiple vectors, we construct a maximum vector consisting of the maximum values of each component, which ensures that the other vectors are on the same side of the maximum vector. Then, the similarity between each vector and the maximum vector is measured respectively, and the similarity between these vectors is determined according to the magnitude of the similarity, which is described as follows:

Assuming that there are m vectors $X_1 = x_{11}, x_{12}, \cdots, x_{1n}$, $X_2 = x_{21}, x_{22}, \cdots, x_{2n}$, $X_m = x_{m1}, x_{m2}, \cdots, x_{mn}$. Each of these vectors has n components.

1. Constructing maximum vector $X_{max} = x_1, x_2, \cdots, x_n$, where $x_i = maxx_{1i}, x_{2i}, \cdots, x_{mi}$, $i = 1, 2, \cdots, n$. Put together the maximum values of the corresponding components of each vector.
2. The similarity of each vector X_i to the maximum vector X_{max} is calculated separately.

$$\lambda_i = \frac{||X_i||}{||X_{max}||} \cdot \frac{[X_i, X_{max}]}{||X_i|| \cdot ||X_{max}||}. \tag{3}$$

3. According to the magnitude of the similarity λ_i, we know the degree of similarity between each vector and the maximum vector. Since these vectors are on the same side of the maximum vector, the value of λ_i also reflects the degree of similarity between any two vectors.

Due to the different importance of each index to describe the essential characteristics of the research object, the weight of the index was taken into account in the cluster analysis. There is one-sidedness when the two-vector similarity calculation formula is used to measure the similarity between multiple research objects, and the logic of multi-vector

similarity calculation is given. By introducing the iterative idea and using the multi-vector similarity calculation formula to accurately measure the similarity between the research objects, a new classification method is presented. The main calculation steps of cluster analysis based on multi-vector similarity are divided into the following six steps:

1. Determine the index system describing the research object. When classifying the research object, we first find out the index system that can describe the essential characteristics of the research object, and grasp the main contradiction of things. If two research objects have the same score in some major indicators, we believe that there is a high probability that they belong to the same category. Therefore, we need to carefully analyze and determine the index system describing the research objects.
2. Evaluate the research objects according to the index system. After establishing the index system of the research object, we can score and evaluate the research object according to the objective reality, and then obtain the original evaluation result matrix , which represents the data of samples containing indicators.
3. According to the consistency requirements, the original evaluation results are standardized. The order of magnitude of the subjects' scores in different indicators may vary by dozens or even hundreds of times. In addition, the direction of each index score is not necessarily the same. In reality, the higher the score of the research object in some indicators, the more it belongs to the same thing, while the lower the score of the research object in other indicators, the more it belongs to the same thing, which does not meet the consistency requirement. To this end, we need to normalize the original data by using the normalization formula of the income index or the normalization formula of the cost index, so as to eliminate the dimension of the original index, compress the evaluation result data into the interval [0,1], and ensure that the index score takes the same direction. Thus, it is helpful to calculate the similarity between the research objects, which is described as follows:
   Let $M_j = \min_i y_{ij}$, $N_j = \max_i y_{ij}$. Where M_j represents the minimum value of column j and N_j represents the maximum value of column j, then the normalization formula of income index and cost index are as follows.
   Normalization formula of income type index: $x_{ij} = \frac{y_{ij}-M_j}{N_j-M_j}$.
   Cost index normalization formula: $x_{ij} = \frac{N_j-y_{ij}}{N_j-M_j}$.
   According to the above formulas, we can transform the original evaluation result matrix $Y_{m\times n}$ into the standardized evaluation matrix $X_{m\times n}$.
4. The weight vector $W = w_1, w_2, \cdots, w_n$ is calculated by AHP. Combined with the standardized evaluation matrix $X_{m\times n}$, the weighted evaluation matrix $Z_{m\times n}$ is obtained. The importance of each index describing the essential characteristics of the research object is different, so the weight of the index must be considered when measuring the similarity between the research objects. AHP is a weighting method combining subjective and objective methods. We use AHP to find the weight vector $W = w_1, w_2, \cdots, w_n$, where w_i represents the weight of the $i-th$ indicator. By multiplying the weight vector by the corresponding component in the standardized evaluation matrix $X_{m\times n}$, the weighted evaluation matrix $Z_{m\times n}$ is obtained.

$$z_{ij} = x_{ij} \times w_j. \tag{4}$$

5. The vector similarity formula between multiple vectors is used to calculate the similarity between samples. The maximum component of each column in the weighted evaluation matrix $Z_{m\times n}$ is taken out and to construct one maximum vector $Z_{max} = z_1, z_2, \cdots, z_n$. Here, $z_i = max z_{1i}, z_{2i}, \cdots, z_{mi}$. On the basis of the above weighted evaluation matrix, we use the vector similarity formula between multiple vectors to calculate the similarity between each sample vector and the maximum vector.

$$\lambda_i = \frac{||Z_i||}{||Z_{max}||} \cdot \frac{[Z_i, Z_{max}]}{||Z_i|| \cdot ||Z_{max}||}. \tag{5}$$

6. Given the clustering level α, classify the research objects. After determining the clustering level α according to the historical management experience, the iterative idea is adopted to classify the research objects reasonably, which is described as follows.
   First, by comparing the similarity between the sample vector and the maximum vector and the size of the clustering level, the samples corresponding to $\lambda_i \geq \alpha$ are classified into the first class.
   Second, after removing the first class of samples, the maximum component of each column in the remaining sample vectors is extracted and a new maximum vector is constructed. Similarly, the vector similarity formula between multiple vectors is used to calculate the similarity β_i between each remaining sample vector and the new maximum vector, and the corresponding samples corresponding to $\beta_i \geq \alpha$ are classified into the second category.
   Third, repeat the above process until all samples have been classified.

3.3. Model Retraining

The feature extraction model $M_{encoder}$ and the similarity model $M_{decoder}$ are spliced together and trained simultaneously. We set an update threshold U. When U samples complete clustering, D group training data is extracted from U samples to form a training set and the model is retrained. The ratio of positive cases to negative cases in training set is 1:1. The positive example extraction method is as follows: randomly select a sample p from U samples, and then randomly select a sample q from U samples in the same cluster as p. Sample p and sample q form a group of positive examples, and set the label as 1. The extraction method of negative cases is as follows: randomly select a sample p from U samples, and then randomly select a sample q from the sample set with different clusters from p among U samples. If there are no clusters in U that are different from p, randomly select a sample q from all samples that have been clustered and are different from p, sample p and sample q form a group of negative cases, and the label is set as 0. If there are no clusters different from p in all the currently clustered samples, no training will be conducted this time.

4. Experiments and Analysis

4.1. Experiment Data

The data for this experiment came from education Weibo. A total of 10828 microblogs about 50 different education events are collected. Thirty education events are used as

training data, from which samples are randomly selected to form a training set. Two texts in the same education event are positive examples, labeled 1. Two texts in different education events are negative examples, labeled 0. The ratio of positive and negative examples in the training set is 1:1, and 350000 sets of samples are extracted to train the model. The remaining 20 education event data are used as a test set, containing 2520 texts in total, for the event clustering experiment.

4.2. Experiment Model Parameter

In the clustering algorithm, the value of N is 25, and the cluster is represented by the first N texts in the event cluster. The larger N is, the stronger the representation of the event cluster is, but the amount of computation will also increase. The value range of threshold L and H in the clustering process is $0 < L < 0.5 < H < 1$. The closer the value of L is to 0, and the closer the value of H is to 1, the more stringent the selection of uncertain samples will be, but at the same time, the calculation amount will be increased. In this experiment, the value of L is 0.3 and the value of H is 0.7. The value of the update threshold U should not be too small, otherwise the training is too frequent, and the model is easily affected by some extreme values, resulting in deviation. If the U value is too large, a large number of samples can only be clustered using the original model. The U value in this experiment is 200. The larger the value of sample number D in each training set, the more obvious the learning effect of the model on event features, but at the same time, it will increase the amount of computation. At the same time, D is also affected by U. In the experiment, the number of training set samples D is 800, which is 4 times of U. The number of training rounds is 5. The experimental environment is Keras framework and Tensorflow. In the training process of feature extraction model , the pre-trained word vector is used, the vector dimension is 300, and the embedding layer is set as untrainable. The output dimension of the DFCEDN layer is 128, Dropout=0.1, Recurrent Dropout=0.1, and the rest are default parameters. The output dimension of the fully connected layer in the similarity model M is 128, the activation function is ReLu, and it also contains two Dropout layers, Dropout=0.1.

4.3. Evaluation Index

We use Purity, normalized mutual information (NMI) and adjusted Rand coefficient (ARI) as the evaluation indexes for clustering effect. Purity is the ratio of the correctly calculated number of texts to the total number of texts. Its definition is shown in equation (6).

$$Purity(\Omega, C) = \frac{1}{N} \sum_{k} \max_{j} |o_k \cap c_j|. \tag{6}$$

Where N represents the total number of samples. $\Omega = o_1, o_2, \cdots, o_k$ represents the clustering cluster division obtained by the clustering model. k represents the total number of predicted event clusters. $C = c_1, c_2, \cdots, c_j$ represents the true classification. j represents the total number of predicted event clusters. The Purity range is $[0, 1]$, and the closer the Purity is to 1, the better the clustering effect is.

NMI is an entropy-based evaluation index, and its definition is shown in equation (7).

$$NMI(\Omega, C) = \frac{I(\Omega, C)}{[H(\Omega) + H(C)]/2}. \tag{7}$$

Where I represents mutual information, as shown in equation (8).

$$I(\Omega, C) = \sum_k \sum_j \frac{|o_k \cap c_j|}{N} \log \frac{N|o_k \cap c_j|}{|o_k| \cdot |c_j|}. \tag{8}$$

H represents entropy, as shown in equation (9).

$$H(\Omega) = -\sum_i \frac{o_i}{N} \log \frac{o_i}{N}. \tag{9}$$

The value range of NMI is [0,1]. The closer the NMI is to 1, the better the clustering effect is.

The ARI makes up for the insufficient punishment of Rand coefficient RI, and its definition is shown in equations (10) and (11).

$$RI = \frac{a + b}{C_N^2}. \tag{10}$$

$$ARI = \frac{RI - mean(RI)}{max(RI) - mean(RI)}. \tag{11}$$

Where, a represents the logarithm of elements of the same class in both the actual class cluster and the cluster prediction class cluster. b represents different categories of actual class clusters, and is also the element logarithm of different categories in the cluster prediction cluster. $mean(RI)$ indicates the average value of RI. The value range of ARI is [0,1]. The closer the ARI is to 1, the better the clustering effect is.

4.4. Model Training

When $M_{encoder}$ and $M_{decoder}$ are spliced together, a twin neural network model M can be formed to calculate text similarity. The feature vectors h_i and h_j of the two texts t_i and t_j are obtained by $M_{encoder}$ respectively, and h_i and h_j are taken as the inputs of $M_{decoder}$, and the similarity P_c of the two texts is finally obtained.

In the process of clustering, the same text t_i needs to calculate the similarity with multiple texts. If the complete twinned neural network model M is used to calculate the similarity, a large time cost will be generated, and the same text t_i will be repeatedly encoded. We split M into $M_{encoder}$ and $M_{decoder}$, each text only needs to be encoded once by $M_{encoder}$, and the simple $M_{decoder}$ is used to calculate the similarity several times, which can greatly reduce the time cost of the clustering process.

During model training, we splice $M_{encoder}$ and $M_{decoder}$ into a twin neural network M, trained M on the training set for 5 rounds, and then split M into $M_{encoder}$ and $M_{decoder}$ to achieve model training. Model retraining is also to retrain the twin neural network M, the number of training rounds is 5 rounds, and then M is divided into $M_{encoder}$ and $M_{decoder}$, so as to achieve the model retraining.

The output of the neural network M is y_{pre}, and the true label is y_{true}, which is 0 or 1. The optimizer is Adam, and the cross entropy loss function is adopted. The calculation steps are shown in equation (12).

$$loss = -\frac{1}{N}\sum_{i=1}^{N} y_{true}\log(y_{pre}) + (1 - y_{true})\log(1 - y_{pre}). \tag{12}$$

4.5. Education Text Clustering Results

We compare the clustering results obtained by the proposed event clustering model with the following clustering methods.

Singlepass [32]: This clustering method uses vector space model to represent text, uses cosine to calculate text similarity, and uses singlepass algorithm for clustering, which belongs to unsupervised clustering algorithm.

K-means: This clustering method uses vector space model to represent the text, uses cosine to calculate the text similarity, and uses K-means algorithm to cluster, which belongs to unsupervised clustering algorithm.

LSH: This is a clustering algorithm based on local sensitive hashing, using a vector space model to represent the text, using cosine to calculate the text similarity, belonging to the unsupervised clustering algorithm.

Hadifar [33], the clustering model uses SIF Embedding to represent the text, auto-encoder learning the low-dimensional feature representation of the text, and deep clustering algorithm is adopted for short text clustering, which is an unsupervised clustering algorithm. In this method, short texts from different fields are used as test sets in the experiment. In this experiment, the test sets are different events in the seismic field.

Wang [34]: This clustering model uses LSTM to extract text features, linear neural network model to calculate text similarity, and incremental clustering algorithm to cluster, which belongs to supervised clustering algorithm.

BERT: This method replaces the model of Wang et al. [34] with BERT word vectors and belongs to supervised clustering algorithm.

Since the clustering results are affected by the order of data input, we randomly scrambled the test data for 10 times of clustering, and recorded the average value of various clustering indicators. The clustering results are shown in Table 1.

Table 1. Cluster results

Model	Purity	NMI	ARI
Singlepass	0.65	0.67	0.50
K-means	0.87	0.86	0.76
LSH	0.59	0.48	0.31
Hadifar	0.61	0.56	0.45
Wang	0.89	0.90	0.71
BERT	0.74	0.71	0.58
Proposed	0.92	0.94	0.81

Among them, the supervised clustering model adopted by Wang [35] has better clustering effect than other clustering models, so this model is taken as the baseline. Under the same training, compared with the baseline model, the proposed model has improved in various clustering indexes, 3% in Purity, 4% in NMI and 10% in ARI.

The text in social media is short, the expression is arbitrary, and even the comments on the same event are expressed in different ways. The use of vector space model and word vector weighting to extract text features is easy to be affected by the interference words, so that the key feature information can not be prominent, resulting in poor clustering effect.

By training the model with supervision, the event features of the text can be identified more accurately and the clustering effect can be enhanced compared with the unsupervised clustering model. However, compared with Word2Vec word vector based clustering method, the clustering result has decreased, because BERT word vector model contains more information, and the model is easy to overfit existing events. The events in the test set did not appear in the training set, so the prediction effect is poor, and the BERT model takes more time than the Word2Vec word vector model, which is less efficient for massive social media texts.

The supervised clustering algorithm relies heavily on the training set. The key to make the model more accurately distinguish various events is whether there are sufficient event types in the training set. Therefore, 10, 15, 20, 25 and 30 education events are selected as the training data respectively, from which 350000 sets of texts are extracted to compose a training set and train the model. 20 education events in Section 3.1 are taken as the test set and the model of Wang et al. [35] is taken as the baseline. Taking NMI as a reference index, the test data is randomly disrupted for 10 times of clustering, and the average value of NMI is obtained, as shown in Figure 2.

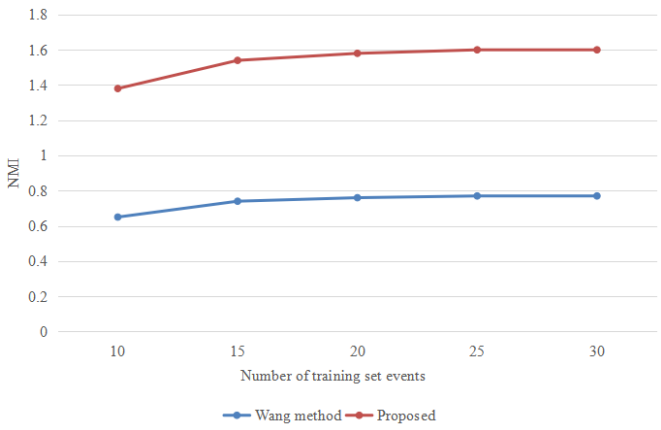


Fig. 2. The influence of the number of training set events on the clustering results

Both model retraining and uncertain sample reunion steps require setting some additional parameters. The update threshold U, which has the greatest influence on the clustering effect of new models, is mainly used to control the frequency of model retraining.

In this experiment, U is set as 10, 50, 200, 500 and 1000 respectively, and the sample number D of each training set is set as 4 times of U, which were 40, 200, 800, 2000 and 4000 respectively. With 10 educational events in Section 3.1 as the test set, Wang et al. [35] model as baseline, and NMI as the reference index, the test data are randomly disrupted for 10 times of clustering, and the NMI is averaged. The results obtained are shown in Figure 3.

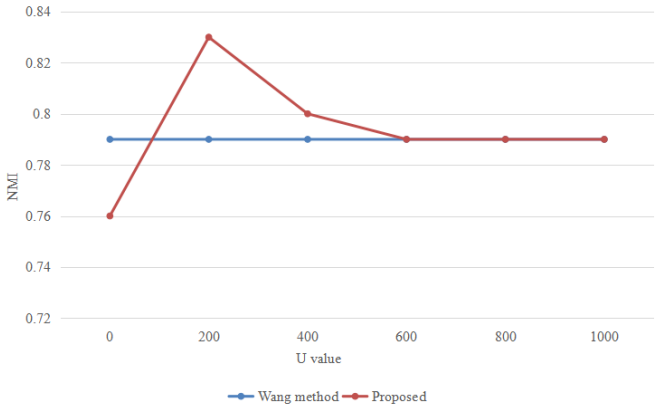


Fig. 3. Influence of U value on clustering results

As can be seen from the results of figure 3, when the U value is small, 10, the clustering effect is poor. The main reason is that the training set is small, and the probability of uneven distribution of categories is large. When U value gradually increases to 200, sample distribution tends to be uniform, and the clustering effect relative baseline will be significantly improved. However, increasing the U value will reduce the number of model retraining, resulting in a large number of data can only be clustered using the original model, so that the clustering results continue to approach but not lower than the baseline. Therefore, the value of U should be taken as small as possible under the condition that the data distribution is as even as possible, so that the proposed model can achieve the best clustering effect.

Table 2 and Figure 4 show the classification effects of different models.

Combined with the experimental results in Table 2 and Figure 4, the following conclusions can be drawn:

(1) It can be seen from the experimental results of K-means and LSH models that LSH has a slightly stronger ability to extract features than K-means, with an increase of 0.14% in accuracy and 0.55% in F1 value.

(2) Comparing experiments on Singlepass, Hadifar, LSH and Wang, it can be seen that BERT word embedding model is better than traditional word embedding to generate word vector. Compared with Singlepass model, the accuracy of Hadifar model is increased by 1.49%, and the F1 value is increased by 2.55%. Compared with the LSH model, the accuracy of the Wang model is increased by 1.37%, and the F1 value is increased by 2.31%.

Table 2. Cluster results

Model	Accuracy/%	F1/%
Singlepass	89.87	88.68
K-means	89.70	89.13
LSH	89.84	89.68
Hadifar	91.36	91.23
Wang	91.21	91.99
BERT	92.56	92.38
Proposed	93.88	93.74

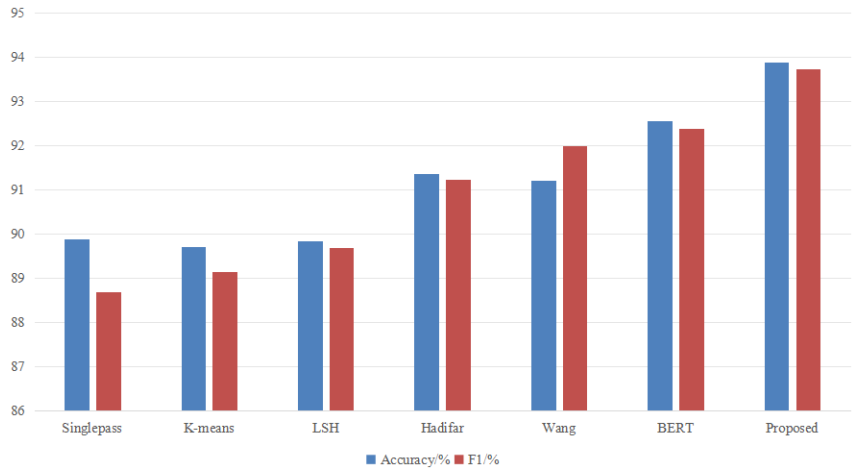


Fig. 4. Classification effect of different models

(3) According to the evaluation indicators of Hadifar, Wang, BERT and the Proposed model, it can be seen that comprehensive consideration of news headlines and text features is better than a general analysis of news text classification. Compared with the Hadifar model, the Proposed model has an increase of 2.52% in accuracy and 2.51% in F1 value. Compared with Wang model, the Proposed model is 2.67% higher in accuracy and 1.75% higher in F1 value.

After comprehensive analysis of the above experimental results, the advantages of the proposed model compared with other models mainly lie in that the proposed model comprehensively considers the differences between the features contained in the news headlines and the news text and fully exploits the text features of each. Since news headlines contain a large number of topic features and keyword features, this model uses LDA topic model to extract the topic features of headlines, Word2Vec model to extract the semantic features of headlines, and fuses the features to obtain the fusion features, and then uses TextCNN model to mine more fine-grained local features. For higher classification standards, just classifying news headlines is far from enough. Due to the short length of news headlines, they lack certain contextual information to supplement the theme features and keyword information contained in them, which easily leads to classification only according to the literal meaning of the headlines. Therefore, this model extracts the context information from the news text on the basis of the classification of news headlines. In the classification of news text, this paper uses BERT model to obtain text word vectors containing context information and complex semantics, and uses BERT to further mine the context information, so as to supplement the topic information and keyword information contained in news headlines. Therefore, the classification model in this paper has been improved compared with other models in various indexes.

The specific classification results of the model proposed in this paper in 10 categories of news are shown in Table 3 and Figure 5. The performance indicator is accuracy.

Table 3. Classification effect of different categories with proposed method

Class	Number	Precision/%	Recall/%	F1/%
Finance	0	91.97	92.80	92.38
property	1	95.39	95.20	95.29
stock	2	89.49	88.60	89.04
education	3	97.38	96.80	97.08
technology	4	89.76	90.30	90.02
society	5	92.58	93.60	93.08
politics	6	90.99	92.01	91.49
sports	7	98.69	98.20	98.44
game	8	96.66	95.10	95.23
recreation	9	95.38	95.10	95.23

From the above results, it can be seen that the model in this paper has a good effect on the classification of news texts in the fields of sports, education and games, especially in the field of sports, with its accuracy rate 98.69%, recall 98.20%, and F1 98.44%. The worst results are in the stock and technology sector, where the accuracy rate of the stock

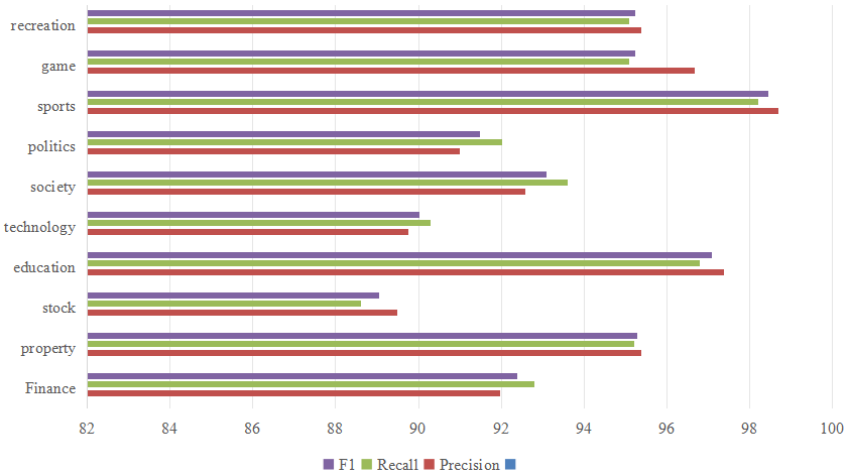


Fig. 5. Classification effect of different categories with proposed method

sector is 89.49%, the recall rate is 88.60%, and the F1 value is 89.04%; The accuracy rate in the technology sector is 89.76%, the recall rate is 90.30%, and the F1 value is 90.02%.

5. Conclusions

Efficient ubiquitous learning can achieve accurate service. Once learners have the right equipment and network, they can get the right resources pushed by the system accurately in the right time, and improve the effective use of information resources. Once the knowledge content gets the rapid flow, its utilization value will be maximized. In the process of research, it is necessary to explore how to optimize and maximize the value of resources in the process of dissemination and use, so as to promote the healthy and sustainable development of information resources.

This paper proposes a novel English social education text event clustering model, which uses DFCEDN to extract text features and linear model to calculate text similarity for incremental clustering. The samples assigned with uncertainty in the clustering process are reclustered after the end. The retraining process can make the model learn new event information, so that the accuracy of the model is constantly improved with the clustering process. Reclustering of uncertain samples can prevent uncertain samples from affecting the centroid representation, reduce the probability of retraining the model with wrong samples, and improve the clustering accuracy of uncertain samples. Compared with the supervised cluster model with the same pre-training, the clustering index of the proposed model is improved.

Acknowledgments. Authors are grateful for the anonymous review by the review experts.

References

1. Cinelli M, De Francisci Morales G, Galeazzi A, et al. The echo chamber effect on social media[J]. Proceedings of the National Academy of Sciences, 2021, 118(9): e2023301118.
2. 2 Li F, Larimo J, Leonidou L C. Social media marketing strategy: definition, conceptualization, taxonomy, validation, and future agenda[J]. Journal of the Academy of Marketing Science, 2021, 49: 51-70.
3. 3 Kross E, Verduyn P, Sheppes G, et al. Social media and well-being: Pitfalls, progress, and next steps[J]. Trends in Cognitive Sciences, 2021, 25(1): 55-66.
4. 4 Örs F K, Yeniterzi S, Yeniterzi R. Event clustering within news articles[C]//Proceedings of the Workshop on Automated Extraction of Socio-political Events from News 2020. 2020: 63-68.
5. 5 Rybski D, Buldyrev S V, Havlin S, et al. Communication activity in a social network: relation between long-term correlations and inter-event clustering[J]. Scientific reports, 2012, 2(1): 1-8.
6. 6 Xu J, Xu B, Wang P, et al. Self-taught convolutional neural networks for short text clustering[J]. Neural Networks, 2017, 88: 22-31.
7. 7 Costache R, Pham Q B, Arabameri A, et al. Flash-flood propagation susceptibility estimation using weights of evidence and their novel ensembles with multicriteria decision making and machine learning[J]. Geocarto International, 2021: 1-33.
8. 8 Balogun A O, Basri S, Capretz L F, et al. An adaptive rank aggregation-based ensemble multi-filter feature selection method in software defect prediction[J]. Entropy, 2021, 23(10): 1274.
9. 9 Long M, Cao Z, Wang J, et al. Conditional adversarial domain adaptation[J]. Advances in neural information processing systems, 2018, 31.
10. 10 Akritidis L, Alamaniotis M, Fevgas A, et al. Confronting sparseness and high dimensionality in short text clustering via feature vector projections[C]//2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI). IEEE, 2020: 813-820.
11. 11 Ma W, Hu X, Chen C, et al. Social media event prediction using DNN with feedback mechanism[J]. ACM Transactions on Management Information Systems (TMIS), 2022, 13(3): 1-24.
12. 12 Hompes B, Buijs J C A M, van der Aalst W M P, et al. Detecting Change in Processes Using Comparative Trace Clustering[J]. SIMPDA, 2015, 2015: 95-108.
13. 13 Adamopoulos P, Ghose A, Todri V. The impact of user personality traits on word of mouth: Text-mining social media platforms[J]. Information Systems Research, 2018, 29(3): 612-640.
14. 14 Liu J, Gao L, Guo S, et al. A hybrid deep-learning approach for complex biochemical named entity recognition[J]. Knowledge-Based Systems, 2021, 221: 106958.
15. 15 Yang Z, Yao F, Fan K, et al. Text dimensionality reduction with mutual information preserving mapping[J]. Chinese Journal of Electronics, 2017, 26(5): 919-925.
16. 16 Rezaee M J, Eshkevari M, Saberi M, et al. GBK-means clustering algorithm: An improvement to the K-means algorithm based on the bargaining game[J]. Knowledge-Based Systems, 2021, 213: 106672.
17. 17 Lilleberg J, Zhu Y, Zhang Y. Support vector machines and word2vec for text classification with semantic features[C]//2015 IEEE 14th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC). IEEE, 2015: 136-140.
18. 18 Vodrahalli K, Chen P H, Liang Y, et al. Mapping between fMRI responses to movies and their natural language annotations[J]. NeuroImage, 2018, 180: 223-231.
19. 19 Soni S, Chouhan S S, Rathore S S. TextConvoNet: a convolutional neural network based architecture for text classification[J]. Applied Intelligence, 2022: 1-20.
20. 20 Liu D, Shan L, Wang L, et al. P3oi-melsh: Privacy protection target point of interest recommendation algorithm based on multi-exploring locality sensitive hashing[J]. Frontiers in Neurorobotics, 2021, 15: 660304.

21. 21 Zhao Z, Gao M, Luo F, et al. LSHWE: improving similarity-based word embedding with locality sensitive hashing for cyberbullying detection[C]//2020 International Joint Conference on Neural Networks (IJCNN). IEEE, 2020: 1-8.
22. 22 Batanović V, Petrović M M. Cross-Level Semantic Similarity for Serbian Newswire Texts[C]//Proceedings of the Thirteenth Language Resources and Evaluation Conference. 2022: 1691-1699.
23. 23 Enguehard J, OHalloran P, Gholipour A. Semi-supervised learning with deep embedded clustering for image classification and segmentation[J]. IEEE Access, 2019, 7: 11093-11104.
24. 24 Zhao M, Wang M, Ma Y, et al. CEIL: A General Classification-Enhanced Iterative Learning Framework for Text Clustering[C]//Proceedings of the ACM Web Conference 2023. 2023: 1784-1792.
25. 25 Pavlinek M, Podgorelec V. Text classification method based on self-training and LDA topic models[J]. Expert Systems with Applications, 2017, 80: 83-93.
26. 26 Zhang X, Zhang L. Topics extraction in incremental short texts based on LSTM[J]. Social Network Analysis and Mining, 2020, 10(1): 83.
27. 27 Teng L, Qiao Y, Shafiq M, et al. FLPK-BiSeNet: Federated Learning Based on Priori Knowledge and Bilateral Segmentation Network for Image Edge Extraction[J]. IEEE Transactions on Network and Service Management, 2023. DOI: 10.1109/TNSM.2023.3273991
28. 28 Li P, Laghari A A, Rashid M, et al. A deep multimodal adversarial cycle-consistent network for smart enterprise system[J]. IEEE Transactions on Industrial Informatics, 2022, 19(1): 693-702.
29. 29 Liu Y, Tong D, Liu X. Measuring spatial autocorrelation of vectors[J]. Geographical Analysis, 2015, 47(3): 300-319.
30. 30 Iscen A, Furon T, Gripon V, et al. Memory vectors for similarity search in high-dimensional spaces[J]. IEEE transactions on big data, 2017, 4(1): 65-77.
31. 31 Yin S, Li H, Sun Y, et al. Data Visualization Analysis Based on Explainable Artificial Intelligence: A Survey[J]. IJLAI Transactions on Science and Engineering, 2024, 2(2): 13-20.
32. 32 Marroig G, Cheverud J. Size as a line of least resistance II: direct selection on size or correlated response due to constraints[J]. Evolution, 2010, 64(5): 1470-1488.
33. 33 Forestiero A, Pizzuti C, Spezzano G. A single pass algorithm for clustering evolving data streams based on swarm intelligence[J]. Data Mining and Knowledge Discovery, 2013, 26: 1-26.
34. 34 Hadifar A, Sterckx L, Demeester T, et al. A self-training approach for short text clustering[C]//Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019). 2019: 194-199.
35. 35 Wang H, Li F. A text classification method based on LSTM and graph attention network[J]. Connection Science, 2022, 34(1): 2466-2480.

Zhenping Jing is with the School of Foreign Languages, Zhengzhou University of Science and Technology. Research interests are Cluster analysis, English semantic analysis, Big Data research, Artificial intelligence.

Received: April 18, 2024; Accepted: June 29, 2024.

Computer Science and Information Systems 21(4):1801–1822 https://doi.org/10.2298/CSIS240418058W

Attitude Estimation of Aircraft Based on Quaternion SRCKF-SLAM Algorithm

Dandan Wang [1,2,*], Zhaokun Zhu[3], Liang Yu[1], Hongjie Li[4], and Kaituo Tan[1]

[1] School of Mechanical and Electrical Engineering, Huainan normal university, 232038 Huainan Anhui,P.R.China;
lansejingling1988@126.com
[2] Human-computer collaborative robot Joint Laboratory of Anhui Province, 232038 Huainan Anhui,P.R.China;
[3] School of Information Engineering, Zhengzhou Technology and Business University, 475000 Zhengzhou Henan,P.R.China;
[4] College of Electronic Information and Electrical Engineering, Anyang Institute Of Technology, 455000 Anyang Henan, P.R. China;

Abstract. In complex terrain environments such as high mountains and hills, traditional agricultural machinery cannot accurately complete tasks such as crop management and harvesting. This paper used plant protection drones as carriers to study the observation content of crops during their navigation process. Aiming at the low accuracy of the traditional quaternion cubature Kalman filtering algorithm for the attitude estimation of the carrier nonlinear state model, a quaternon-based square root cubature Kalman filtering algorithm was proposed in this paper.The algorithm takes the attitude quaternion error and the gyro drift error as the state quantity, and measures the attitude quaternion of SINS/SLAM navigation. The square root cubature Kalman filter algorithm is used for pose estimation, which not only solves the standardization problem of traditional quaternion, but also reduces the state dimension and complexity of the square root UKF algorithm of traditional quaternion, and improves the numerical stability. Compared with the quaternion SRUKF and quaternion SRCDKF algorithm, the simulation results showed that the new algorithm estimated the error mean values of the roll angle, pitch angle and runt angle, which are 0.05°, 0.08°, and 0.03°, respectively. The error is the smallest, and algorithm accuracy is about 30% higher than the quaternion SRUKF-SLAM algorithm, and it has high filtering accuracy and numerical stability, and the best time-consuming performanc.

Keywords: quaternion, state model, square root Cubature Kalman filter, attitude estimation, numerical stability.

1. Introduction

As a new star in smart agriculture, plant protection drones carry important tasks such as precise pesticide application, targeted fertilization, environmental monitoring, information collection and processing. The accuracy of drone tracking and pose

* Corresponding Author

estimation is the technical guarantee for achieving the detection and orderly management of crop growth environment[1-3].When dealing with non-linear state models in target tracking systems, attitude quaternions are usually used as the attitude model parameters of the carrier[4-6], which has the advantages of smaller global dimension and non-singularity than Euler angle attitude parameter description methods and can achieve target full attitude tracking.

Firstly, starting from the analysis of linear systems, which are referred to as the first type of systems in this article, their models are relatively simple, and the state parameters can also be obtained through analytical methods to achieve system state updates. Kalman filtering method is often used for implementation; The second type of system is a conditional nonlinear system, which satisfies Markov processes and has been applied in many military fields. The linear part of the system model can be obtained by analytical methods to obtain system parameters, while the nonlinear part can be estimated by other algorithms. The processing method used is mentioned in reference[7] as a Rao Blackwelled (RBPF) filter based on a combination of particle filtering and Kalman filtering algorithms, The basic idea is to first obtain the parameters of the linear system model state variables under nonlinear partial conditions through the posterior probability distribution in particle filtering, and then use Kalman filters to estimate the parameters of the linear state variables. However, such systems require the use of both linear and nonlinear algorithms, which obviously require a large amount of computation, and the stability and real-time performance of the system often do not meet the requirements.

Therefore, the key to solving the drawbacks of RBPF algorithm, such as high computational complexity, lies in finding a new method to solve the problem of state estimation of nonlinear systems in conditional nonlinear systems. This has also attracted the attention and research of scholars at home and abroad. In reference[8], an extended Kalman filtering algorithm is proposed, which linearizes the nonlinear system model, ignores the integration of high-order terms, but brings calculation errors, Known as truncation error, although it solves the problem of solving the state parameter of nonlinear systems, this algorithm also has lower filtering accuracy due to such errors.

A new combination algorithm based on the combination of unscented Kalman filter and traditional Kalman filter is proposed in reference[9]. The estimation error obtained by Unscented Kalman filter for solving the nonlinear part is much smaller than the probability density approximation method of particle filter algorithm. The idea is to use UT scale transformation, select 2n+1 Sigma points with certain weights, approximate the state parameters in the nonlinear system, usually their mean and covariance, The estimation accuracy of the algorithm is higher than that of traditional particle filtering algorithms and EKF algorithms.

For the singularity of 3D pose parameters and the redundancy of solving high-dimensional pose parameters for non-linear system modeling, the pose estimation of modeling using traditional quaternion parameters as state quantities cannot solve the above problems well. Vathsal[10-11] and others first proposed the quaternion second-order extended Kalman filter (Q-Extended Kalman filter (QEKF)). This algorithm has a low accuracy due to the complexity of the matrix solution. So literatures[12-14] proposed a quaternion-based unscented Kalman filtering algorithm filtering algorithm (Q- unscented Kalman filter, QUKF). However, due to the large selection of Sigma points of the traditional UKF algorithm and the high dimensionality of state quantities,

and those negative weighted Sigma points will cause non-positive definite problems of the algorithm's covariance, the accuracy of the algorithm needs further improvement. In 2009, the idea of generating a brand new set of 2N cubic points with the same weight after a non-linear cubature transformation with a sampling point was proposed, which is the famous cubic Kalman filter (CKF) Algorithm[15]. The algorithm first calculates and propagates the cubature points, through iterative updates of the equation of state, calculating the mean and estimating covariance of observations, and finally calculates the Kalman gain using the cubature radial criterion. Algorithm accuracy is improved compared to previous algorithms.

Although the filtering accuracy of the traditional CKF algorithm has been improved to some extent, when the standard CKF algorithm solves the nonlinear problem of SLAM, with the continuous addition of observation features, the calculation amount increases sharply, and the problem of numerical instability occurs. In this regard, a series of adaptive high-order cubature Kalman filter (HCKF), CDKF and other algorithms have been proposed, but they are basically combined with the covariance matrix to introduce a fading factor to improve the strong tracking ability, but all have their own problem in numerical calculations and filtering accuracy. Combining the ideas of the square root cubature Kalman filter algorithm (SRCKF) proposed in[16-18] etc, a quaternion based square root cubature Kalman filter algorithm (QSRCKF) is proposed.The square root cubature Kalman filter not only reduces the state quantity dimension and computational complexity, but also updates of the square root covariance subtype iteratively which can ensure the non-negative qualitative and symmetry of the algorithm. The new algorithm used in SINS / SLAM integrated navigation can reduce the position error of the carrier and improve the accuracy of carrier attitude estimation.

2. Rationrelated works

In recent years, the author has proposed a square root UKF attitude estimation algorithm based on fuzzy logic quaternion to address the issues of low filtering accuracy during navigation calculation and the inability of traditional quaternion extended Kalman filtering algorithms to meet the accuracy requirements of target attitude measurement in target tracking estimation algorithms, which use sensor discrete observations to estimate the continuous state of the target. This algorithm takes quaternions as the state of the fuzzy logic UKF filter, applies the measured angular rate to complete the time update of the filter, tracks and locates the target in real time, and updates the measurement, suppressing the problem of attitude error divergence. Among them, the fuzzy square root UKF algorithm based on quaternion was compared and analyzed with quaternion UKF and square root quaternion algorithms through simulation experiments, and the position estimation of observed feature targets in the x and y directions was achieved.

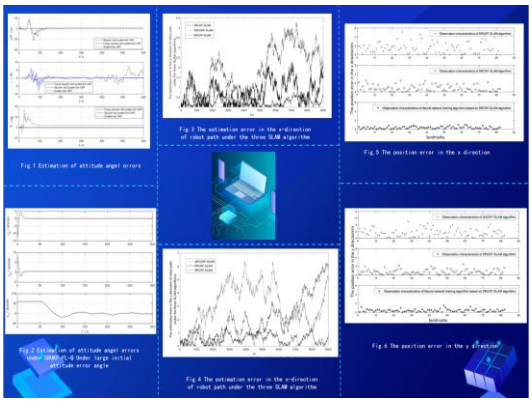


Fig. 1. Carrier attitude estimation error under SRUKF algorithm

From Figure, it can be seen that by changing the initial values of the three types of attitude angles of the carrier, the roll angle and pitch attitude angle errors under this algorithm can approach 0 in a relatively short period of time; Furthermore, as the simulation time progresses, as the yaw angle represents the direction of the carrier's navigation and remains constant for the first 50 seconds, its error also remains unchanged. In the following 50 seconds, its error gradually decreases as well. And achieved the position estimation of observed feature targets in the x and y directions.

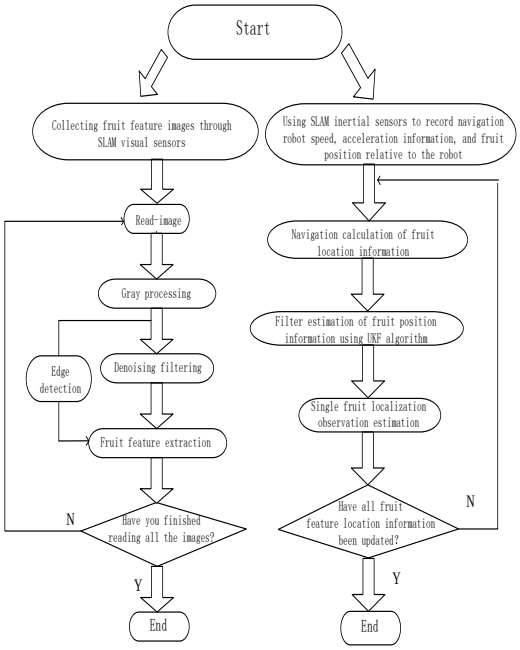


Fig. 2. The overall algorithm flowchart

This paper improves on the SRUKF-SLAM algorithm studied in recent years and proposes a new method for aircraft attitude estimation based on the quaternion SRCKF algorithm. The overall algorithm flowchart[19] is shown in figure 2.

3. Quaternion description of aircraft attitude

Using SINS / SLAM integrated navigation to estimate the attitude of the aircraft can effectively improve the positioning accuracy and strengthen tracking ability. Assumes local coordinate system as global reference $ox_n y_n z_n$ (n)coordinates and carrier coordinate system as $ox_b y_b z_b$ (b)system.Given the reference coordinate system, the rotation order of the carrier system relative to the system is $Z \to X \to Y$. The rotation angles are yaw γ , roll angle α and the pitch attitude angles are β . The attitude rotation matrix of the $n \to b$ system.

$$\begin{aligned} C_n^b &= \begin{bmatrix} \cos\beta & 0 & -\sin\beta \\ 0 & 1 & 0 \\ \sin\beta & 0 & \cos\beta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha \\ 0 & -\sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} \cos\gamma & \sin\gamma & 0 \\ -\sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} \cos\beta\cos\gamma - \sin\alpha\sin\beta\sin\gamma & \cos\beta\sin\gamma + \sin\alpha\sin\beta\cos\gamma & -\cos\alpha\sin\beta \\ -\cos\alpha\sin\gamma & \cos\beta\cos\gamma & \sin\alpha \\ \sin\beta\cos\gamma + \sin\alpha\cos\beta\sin\gamma & \sin\beta\sin\gamma - \sin\alpha\cos\beta\cos\gamma & \cos\alpha\cos\beta \end{bmatrix} \end{aligned} \tag{1}$$

defines the carrier attitude quaternion as

$$q = \begin{bmatrix} q_1 & q_2 & q_3 & q_4 \end{bmatrix}^T \tag{2}$$

According to the definition of quaternion, satisfy the relationship

$$q^T q = \|q\|^2 = q_1^2 + q_2^2 + q_3^2 + q_4^2 = 1 \tag{3}$$

$$r^b = C_n^b r^n \tag{4}$$

Then through the coordinate rotation relationship and quaternion rotation theory, it can be calculated the quaternion's attitude transformation matrix for the carrier in $n \to b$ system is

$$C_n^b = \begin{bmatrix} q_1^2 + q_2^2 - q_3^2 - q_4^2 & 2(q_2 q_3 + q_1 q_4) & 2(q_2 q_4 - q_1 q_3) \\ 2(q_2 q_3 - q_1 q_4) & q_1^2 + q_3^2 - q_2^2 - q_4^2 & 2(q_1 q_2 + q_3 q_4) \\ 2(q_1 q_3 + q_4 q_2) & 2(q_3 q_4 - q_1 q_2) & q_1^2 + q_4^2 - q_2^2 - q_3^2 \end{bmatrix} \tag{5}$$

According to the transformation relationship between the Euler angle attitude matrix and the quaternion attitude matrix, the carrier attitude angle is calculated as

$$\begin{cases} \alpha = \arctan 2\left[2\left(q_1 q_2 + q_3 q_4\right) \quad \sqrt{4\left(q_2 q_4 - q_1 q_3\right)^2 + \left(q_1^2 + q_4^2 - q_2^2 - q_3^2\right)^2} \right] \\ \beta = \arctan\left[2\left(q_2 q_4 - q_1 q_3\right) \quad \sqrt{4\left(q_1 q_2 + q_3 q_4\right)^2 + \left(q_1^2 + q_4^2 - q_2^2 - q_3^2\right)^2} \right] \\ \gamma = \arctan 2\left[2\left(q_2 q_3 - q_1 q_4\right) \quad q_1^2 + q_2^2 - q_3^2 - q_4^2 \right] \end{cases} \tag{6}$$

3.1. Gyro Quaternion Model

Suppose the real angular velocity of the carrier relative to the reference frame is According to the rigid body rotation theorem, the quaternion differential equation of the attitude of the carrier can be obtained as:

$$\dot{q} = \frac{1}{2}\Omega_{nb}^{b} q \tag{7}$$

among them $\Omega_{nb}^{b} = \begin{bmatrix} 0 & -\omega_{nbx}^{b} & -\omega_{nby}^{b} & -\omega_{nbz}^{b} \\ \omega_{nbx}^{b} & 0 & \omega_{nbz}^{b} & -\omega_{nby}^{b} \\ \omega_{nby}^{b} & -\omega_{nbz}^{b} & 0 & \omega_{nbx}^{b} \\ \omega_{nbz}^{b} & \omega_{nby}^{b} & -\omega_{nbx}^{b} & 0 \end{bmatrix}$.

Under ideal conditions, the quaternion state equation is:

$$q_k = q_{k-1} + \frac{\Delta t}{2}\Omega_{k|k-1} q_{k-1} \tag{8}$$

Among them, Δt is the angular velocity sampling period.

The measured gyroscope output is ω_{ib}^{b} including real angular velocity ω_{nb}^{b} and gyroscope random drift ω_{in}^{b} . That is $\omega_{ib}^{b} = \omega_{nb}^{b} + \omega_{in}^{b}$ so the gyroscope model is expressed as:

$$\omega_{nb}^{b} = \omega_{ib}^{b} - \omega_{in}^{b} = \omega_{ib}^{b} - C_{n}^{b}\omega_{in}^{n} \tag{9}$$

Where, ω_{in}^{n} is the value of the gyro drift in the reference n frame.

The quaternion output of the CCD star sensor is used as the gyroscope attitude measurement. It is assumed that the direction of the angular velocity is fixed during adjacent times. The discrete equation of the kinematics of the carrier represented by the quaternion is:

$$q_{k+1} = \Omega\left(\omega_k\right) q_k = \begin{bmatrix} \cos\left(\frac{1}{2}\|\omega_k\|\Delta t\right) I_{3\times 3} - \left[\psi_k^{\times}\right] & \psi_k \\ -\psi_k^{T} & \cos\left(\frac{1}{2}\|\omega_k\|\Delta t\right) I_{3\times 3} \end{bmatrix} \tag{10}$$

In this formula $\psi_k = \sin\left(\frac{1}{2}\|\omega_k\|\Delta t\right)\omega_k \Big/ \|\omega_k\|$, $\left[\psi_k^{\times}\right]$ is the antisymmetric matrix for ψ_k.

Expand the quaternion differential equation in equation (7):

$$\dot{q} = \frac{1}{2}\Omega(\omega)q = \frac{1}{2}\begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix} q \tag{11}$$

The gyroscope is used to measure the angular velocity of the carrier. The state model is:

$$\begin{cases} \omega_g = \begin{bmatrix} \omega_x & \omega_y & \omega_z \end{bmatrix}^T = \hat{\omega} - b - \xi_a \\ \dot{b} = \xi_b \end{cases} \tag{12}$$

In this formula ω_g is the true angular velocity. $\hat{\omega}$ is the actual gyroscope output value. b is the gyroscope drift error and ξ_a、ξ_b is the system's white Gaussian noise which meets $\xi_a \sim N\left(0, \sigma_a^2\right)$, $\xi_b \sim N\left(0, \sigma_b^2\right)$.

3.2. Star sensor observation model

The star sensor measurement model is described as:

$$z_k = H\left(q_k\right)r + v_k \tag{13}$$

In this formula z_k is the output of the star sensor; $H\left(q_k\right)$ is the attitude matrix of the star sensor at all times; is the reference vector of the star sensor; and is the zero mean Gaussian white noise.

4. SRCKF-SLAM algorithm based on quaternion

Let the system state quantity and covariance matrix in the SLAM model be:

$$x = \begin{bmatrix} x_v & x_m \end{bmatrix},\ P = \begin{bmatrix} P_{vv} & P_{vm} \\ P_{vm}^T & P_{mm} \end{bmatrix}.$$

Where, x_v, x_m are carrier state and map feature state respectively, and each contains two-dimensional position information in the direction x, y. P_{vv}, P_{vm}, P_{mm} are the carrier autocorrelation covariance matrix, the carrier and map feature cross-covariance matrix, and the map feature autocovariance matrix respectively.

Under control input u_k, consider the SLAM problem as the state quantity and the posterior joint probability density problem:

$$p\left(x_{v,k}, x_m \mid z_k, u_k\right) = p\left(z_k \mid x_{v,k}, x_m\right) \cdot \int p\left(x_{v,k} \mid x_{v,k-1}, u_k\right) \cdot p\left(x_{v,k-1}, x_m \mid z_{k-1}, u_{k-1}\right) dx_{v,k-1} \tag{14}$$

In this formula, $p\left(x_{v,k} \mid x_{v,k-1}, u_k\right)$ is a motion model and $p\left(z_k \mid x_{v,k}, x_m\right)$ is a measurement model. The general nonlinear motion model can be expressed as:

$$\begin{cases} x_k = f\left(x_{k-1}\right) + w_k \\ z_k = h\left(x_k\right) + v_k \end{cases} \tag{15}$$

In this formula, $f(\cdot)$ and $h(\cdot)$ are nonlinear system function and observation function. Uncorrelated system Gaussian white noise and observed Gaussian white noise are $w_k \sim N(0, Q)$、$v_k \sim N(0, R)$.

4.1. SRCKF algorithm update steps

State parameter initialization

$$\begin{cases} \hat{x}_0 = E\left[x_0\right], P_0 = E\left[\left(x_0 - \hat{x}_0\right)\left(x_0 - \hat{x}_0\right)^T\right] \\ S_0 = \mathrm{Chol}\left\{E\left[\left(x_0 - \hat{x}_0\right)\left(x_0 - \hat{x}_0\right)^T\right]\right\} \end{cases} \tag{16}$$

In this formula, Chol represents matrix Cholesky's factorization.

(1)Time update

Set $k-1$time joint posterior probability $P\left(x_{k-1}\right) = N\left(\hat{x}_{k-1|k-1}, P_{k-1|k-1}\right)$ known, through Cholesky decomposes error covariance $P_{k-1|k-1} = S_{k-1|k-1} S_{k-1|k-1}^T$ · $S_{k-1|k-1}$ is the error covariance equation in the formula.

①Use dimension n point cubature to generate dimension $2n$ clllection of cubature points:

$$x_{i,k-1} = S_{k-1}\varepsilon_i + \hat{x}_{k-1} \tag{17}$$

In the formula, $\varepsilon_i = \sqrt{n}\left[1\ \right]_{2n}$ is the point of origin. The corresponding weights for each cubature point are: $\tau_i = \frac{1}{2n}$, $i = 1, 2, \mathrm{L}\ , 2n$.

②Propagate the cubature point and calculate the state prediction value:

$$x_{i,k|k-1}^* = f\left(x_{i,k-1}\right) \tag{18}$$

$$\hat{x}_{k|k-1} = \sum_{i=1}^{2n} \tau_i x_{i,k-1}^* \tag{19}$$

③Calculate the predicted value of state error covariance:

$$P_{k|k-1}=\tau_i\sum_{i=1}^{2m}\left(x^*_{k|k-1}x^{*T}_{k|k-1}-\hat{x}_{k|k-1}\hat{x}^{*T}_{k|k-1}+Q_{k-1}\right) \tag{20}$$

Rewrite Q_{k-1} to Cholesky decomposition:

$$Q_{k-1}=S_{Q,k-1}S^T_{Q,k-1} \tag{21}$$

④State error covariance equation:

$$\begin{cases} S_{k|k-1}=qr\left[\chi_{k|k-1}\quad S_{Q_{k-1}}\right] \\ \chi^*_{k|k-1}=\dfrac{1}{\sqrt{2n}}\left[x^{*,1}_{k|k-1}-\hat{x}_{k|k-1},x^{*,2}_{k|k-1}-\hat{x}_{k|k-1},\right. \\ \left.\mathrm{L}\ ,x^{*,2n}_{k|k-1}-\hat{x}_{k|k-1}\right] \end{cases} \tag{22}$$

In the formula, qr is the decomposition of QR matrix.

(2)Measurement update

①Calculate the cubature point

$$x_{i,k|k-1}=S_{k|k-1}\varepsilon_i+\hat{x}_{k|k-1},\ i=1,2,\mathrm{L}\ ,2n \tag{23}$$

②Propagation cubature point

$$z^*_{i,k|k-1}=h\left(x_{i,k|k-1}\right) \tag{24}$$

③Calculate the predicted value of observation at k time:

$$\hat{z}_{k|k-1}=\frac{1}{2n}\sum_{i=1}^{2n}z^*_{k|k-1} \tag{25}$$

④Calculate autocorrelation covariance

$$P_{zz,k|k-1}=\tau_i\sum_{i=1}^{2n}\left(z_{i,k|k-1}z^T_{i,k|k-1}-\hat{z}_{i,k|k-1}\hat{z}^T_{i,k|k-1}+R_k\right) \tag{26}$$

Rewrite R_{k-1} into Cholesky decomposition

$$R_{k-1}=S_{R,k-1}S^T_{R,k-1} \tag{27}$$

Covariance square root error:

$$\begin{cases} S_{zz,k|k-1}=qr\left[\eta_{k|k-1}\quad S_{R_k}\right] \\ \eta_{k|k-1}=\dfrac{1}{\sqrt{2n}}\sum_{i=1}^{2n}\left(\begin{matrix} z^{*,1}_{i,k|k-1}-\hat{z}_{k|k-1},z^{*,2}_{i,k|k-1}-\hat{z}_{k|k-1}, \\ \mathrm{L}\ ,z^{*,2n}_{i,k|k-1}-\hat{z}_{k|k-1}\end{matrix}\right) \end{cases} \tag{28}$$

The square root of the estimated (weight) cross-covariance is

$$\begin{cases} \chi_{k|k-1}=\dfrac{1}{\sqrt{2n}}\left[\begin{matrix} x^1_{k|k-1}-\hat{x}_{k|k-1},x^2_{k|k-1}-\hat{x}_{k|k-1}, \\ \mathrm{L}\ ,x^{2n}_{k|k-1}-\hat{x}_{k|k-1}\end{matrix}\right] \\ P_{xz,k|k-1}=\chi_{k|k-1}\eta^T_{k|k-1} \end{cases} \tag{29}$$

Kalman gain is:

$$K_k = \left(P_{xz,k|k-1} / S_{zz,k|k-1}^T \right) S_{zz,k|k-1} \tag{30}$$

Correlation state root square error

$$S_{k|k} = qr\left[\chi_{k|k-1} - K_k \eta_{k|k-1}, \;\; k_k S_{R,k} \right] \tag{31}$$

Four-element SRCKF select attitude error $\delta x_{i,k-1}^q$ and gyro drift error $\delta x_{i,k-1}^{\mathrm{b}}$ to make joint status:

$$\delta x_{i,k-1} = \left[\delta x_{i,k-1}^q, \delta x_{i,k-1}^{\mathrm{b}} \right]^T = S_{k-1} \varepsilon_i, \quad i = 1,2,\mathrm{L}\ ,2n \tag{32}$$

The quaternion SRCKF state cubature points are

$$\begin{cases} x_{i,k-1}^q = \delta \hat{q}_{i,k-1} \otimes \hat{q}_{k-1} \\ x_{i,k-1}^{\mathrm{b}} = \hat{x}_{k-1}^{\mathrm{b}} + \delta \hat{x}_{i,k-1}^{\mathrm{b}} \end{cases} \tag{33}$$

Define the error quaternion as

$$\delta x_{i,k-1}^q = \hat{q}_{k|k-1} \otimes \left(x_{i,k-1}^q \right)^{-1} \tag{34}$$

4.2. SRCKF-SLAM algorithm update steps

(1) Update the quaternion SRCKF-SLAM time:

$$\begin{cases} x_{i,k|k-1} = f\left(x_{i,k-1} \right) \\ \hat{\mathrm{b}} = \sum_{i=1}^{2n} \tau_i x_{i,k|k-1}^{\mathrm{b}} \\ S_{k|k-1}^q = qr\left[\chi_{q,k|k-1}^* \quad S_{Q_{k-1}}^q \right] \\ S_{k|k-1}^{\mathrm{b}} = qr\left[\chi_{\mathrm{b},k|k-1}^* \quad S_{Q_{k-1}}^{\mathrm{b}} \right] \\ S_{k|k-1}^{q,\mathrm{b}} = qr \begin{bmatrix} \tau_i \delta x_{k|k-1}^{q,1} \left(x_{k|k-1}^{\mathrm{b},1} - \hat{\mathrm{b}}_{k|k-1} \right)^T \\ \tau_i \delta x_{k|k-1}^{q,2} \left(x_{k|k-1}^{\mathrm{b},2} - \hat{\mathrm{b}}_{k|k-1} \right)^T \\ \mathrm{M} \\ \tau_i \delta x_{k|k-1}^{q,2n} \left(x_{k|k-1}^{\mathrm{b},2n} - \hat{\mathrm{b}}_{k|k-1} \right)^T \end{bmatrix}^T \\ S_{k|k-1} = \begin{bmatrix} S_{k|k-1}^q & S_{k|k-1}^{q,\mathrm{b}} \\ \left(S_{k|k-1}^{q,\mathrm{b}} \right)^T & S_{k|k-1}^{\mathrm{b}} \end{bmatrix} \end{cases} \tag{35}$$

Where, $\hat{\mathrm{b}}$ is the predicted value for gyro drift part, $S_{k|k-1}^q$、$S_{k|k-1}^{\mathrm{b}}$ are square root covariance sub-forms for pose estimation and gyro drift error, and

$$\begin{cases} \chi^{*}_{q,k|k-1}=\dfrac{1}{2n}\left[\delta x^{q,1}_{k|k-1},\delta x^{q,2}_{k|k-1},\mathrm{L}\ ,\delta x^{q,2n}_{k|k-1}\right] \\ \chi^{*}_{\mathrm{b},k|k-1}=\dfrac{1}{2n}\begin{bmatrix} x^{\mathrm{b},1}_{k|k-1}-\hat{b}_{k|k-1},x^{\mathrm{b},2}_{k|k-1}-\hat{b}_{k|k-1}, \\ \mathrm{L}\ ,x^{\mathrm{b},2\mathrm{n}}_{k|k-1}-\hat{b}_{k|k-1} \end{bmatrix} \end{cases} \tag{36}$$

Update the quaternion SRCKF-SLAM measurement:

①Update the quaternion SRCKF state cubature point:

$$\xi^{q}_{i,k|k-1}=\delta z^{q}_{i,k|k-1}\otimes\hat{q}_{k|k-1} \tag{37}$$

②Measurement update and error auto-covariance sub-form and cross-covariance update

$$\begin{cases} z^{q}_{i,k-1}=\delta\hat{q}_{i,k-1}\otimes\xi^{q}_{i,k|k-1}\otimes\hat{q}_{k-1} \\ \delta z_{i,k}=S_{k|k-1}\varepsilon_{i}=\left[\delta z^{q}_{i,k-1},\delta z^{\mathrm{b}}_{i,k-1}\right]^{T} \\ S_{zz,k}=qr\left[\eta_{z,k}\quad S^{\eta}_{R_{k-1}}\right] \\ P_{xz,k}=\begin{bmatrix} \tau_{i}\sum\limits_{i=1}^{2n}\delta x^{q}_{k|k-1}\left(\delta z^{q}_{k|k-1}\right)^{T} \\ \tau_{i}\sum\limits_{i=1}^{2n}\left(x^{\mathrm{b}}_{i,k|k-1}-\hat{\mathrm{b}}_{k|k-1}\right)\left(\delta z^{q}_{i,k|k-1}\right)^{T} \end{bmatrix}^{T} \end{cases} \tag{38}$$

$$\begin{cases} \delta z^{q}_{i,k-1}=\hat{z}_{k|k-1}\otimes\left(\delta z^{q}_{i,k-1}\right)^{-1} \\ \eta_{z,k}=\dfrac{1}{2n}\left[\delta z^{q,1}_{k},\delta z^{q,2}_{k},\mathrm{L}\ ,\delta z^{q,2n}_{k}\right] \end{cases} \tag{39}$$

Kalman gain:

$$K_{k}=\left(P_{xz}/S^{T}_{zz,k}\right)\cdot\frac{1}{S_{zz,k}} \tag{40}$$

State error amount:

$$\delta\hat{x}_{k}=K_{k}\delta\hat{z}_{k|k-1} \tag{41}$$

Where, $\begin{cases} \delta\hat{x}_{k}=\left[\delta\hat{q}^{T}_{k}\quad\delta\hat{b}^{T}_{k}\right]^{T} \\ \delta\hat{z}_{k|k-1}=z_{k}\otimes\left(\hat{z}_{k|k-1}\right)^{-1}\cdot \\ \delta\hat{q}_{k}=\dfrac{\delta\hat{q}_{k}}{\left\|\delta\hat{q}_{k}\right\|_{2}} \end{cases}$

③Gyro drift update

$$\begin{cases} \hat{q}_{k}=\delta\hat{q}_{k}\otimes\hat{q}_{k|k-1} \\ \hat{b}_{k}=\hat{b}_{k|k-1}+\delta\hat{b}_{k} \end{cases} \tag{42}$$

The state quantity error covariance matrix is:

$$S_k = chol\left(S_k S_{k|k-1}^T, U_k U_k^T, -1\right) \tag{43}$$

Where, $U_k = K_k S_{zz,k}$.

5. Simulation experiments and analysis

5.1. Comparison of Three Filtering Algorithms for Attitude Angle Estimation

Taking a near-Earth satellite as an example, a star sensor is installed vertically, and a SINS / SLAM integrated navigation and CCD attitude estimation system platform is established, and tested in Matlab and C ++ software. Assumed satellite orbit altitude is 800Km and the inclination angle is 82 °. The initial position of the aircraft is 114 ° E, 36 ° N. Initial attitude angle error and initial gyro drift are zero. The gyro initial drift error variance matrix is $\mathrm{diag}\left(0.2, 0.2, 0.2\right)_{3\times3}$. The initial attitude angle error variance matrix is $\mathrm{diag}\left(0.25, 0.25, 0.25\right)_{3\times3}$. The initial process noise is $\mathrm{diag}(0.02^2 I_{3\times3}, 0.05^2 I_{3\times3})$. The initial measurement noise variance matrix is $\mathrm{diag}\left[20, 20, 20\right]_{3\times3}$. The simulation time is 600s, and other parameters are shown in Table 1.

Table 1. Parameters of aircraft and gyros

Parameter	Value	Parameter	Value
Satellite orbital angular velocity	0.001rad/s	Gyro constant drift error	$0.1°/h$
Satellite initial attitude angular velocity	$[0.1\ 0.1\ 0.1]^{\mathrm{T}}$(°/s)	Gyro drift standard deviation of noise	$0.02\left(°/\sqrt{\mathrm{h}}\right)$
Satellite three-axis initial attitude angle	$[0\ 0\ 5]^{\mathrm{T}}$(°)	Gyroscope measurement standard deviation of noise	$0.05\left(°/\sqrt{\mathrm{h}}\right)$
Satellite triaxial initial error angle	$[0.5\ 0.5\ 5]^{\mathrm{T}}$(°)	Gyroscope output frequency	50Hz
Star sensor Output frequency	2Hz	Star sensor measuring noise	$N\left(0, 20''\right)$
Satellite orbital angular velocity	0.001rad/s	Gyro constant drift error	$0.1°/h$

In order to verify the superiority of the proposed algorithm based on SLAM nonlinear model estimation accuracy, this experiment compared the quaternion SRUCF-SLAM and quaternion SRCDKF-SLAM algorithms to estimate the roll attitude angle error, pitch attitude angle error, and flight The magnitude of the deviation attitude error, and the simulation results are shown in figures 3 to 5:

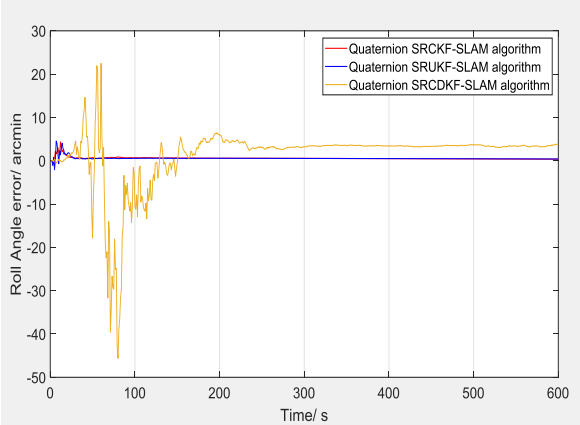


Fig. 3. Roll angle error

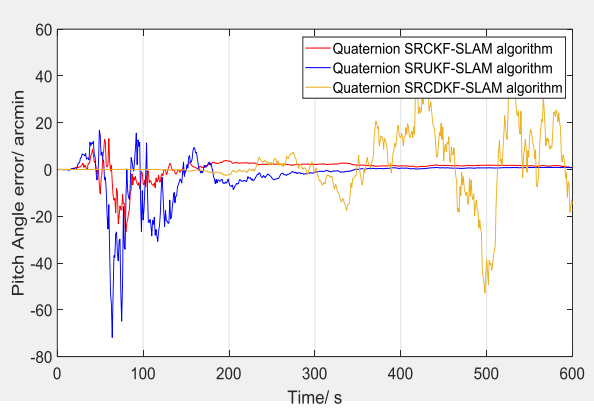


Fig. 4. Pitch angle error

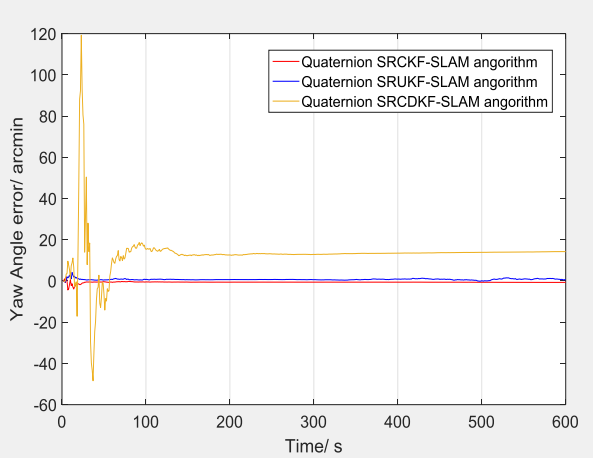


Fig. 5. Yaw angle error

As can be seen from Figures 3, 4, and 5, given the initial conditions, the estimation error based on the quaternion SRCDKF-SLAM is the largest for the roll attitude and yaw attitude angle estimates, and the quaternion SRCKF- SLAM algorithm has almost the same filtering accuracy as the quaternion SRUKF-SLAM algorithm.

However, the computational complexity of the former is smaller than that of the latter, and the numerical stability is higher.

For pitch attitude estimation, it is clear that the quaternion SRCKF-SLAM has the smallest error, and can quickly converge to zero, with the best stability.

The mean statistics of the attitude angle estimation errors of the three filtering algorithms are shown in Table 2:

Table 2. Comparison of performance based of three algorithms

Algorithm parameter	Quaternion SRCDKF-SLAM	Quaternion SRUKF-SLAM	Quaternion SRCKF-SLAM
Rolling angle error mean	8″	4″	3″
Absolute mean value of pitch error	19″	6″	5″
Absolute mean value of heading error	24″	4″	2″
Time spent in one iteration	0.028s	0.036s	0.012s

As can be seen from Table 3, the estimation errors of the quaternion SRCDKF-SLAM algorithm, quaternion SRUCF-SLAM algorithm, and quaternion SRCCKF-SLAM algorithm decrease sequentially, and the new algorithm has the shortest iteration time and the strongest time-consuming performance, which is consistent with simulation graphics.

5.2. Experimental simulation and analysis under conditions of large initial error angle

The quaternion SRCDKF and quaternion SRUKF algorithms have large errors in attitude angle estimation, which leads to a large proportion of the vertical coordinate of the three filtering simulation graphics. Robustness and filtering accuracy. In this set of experiments, the initial roll angle attitude angle error, pitch attitude angle error, and yaw attitude angle error are 30 arcmin, 30 arcmin, and 60 arcmin, respectively. The initial conditions of the gyroscope are consistent with 5.1, the simulation time is 300s, and the simulation graph is shown in Figure 6:

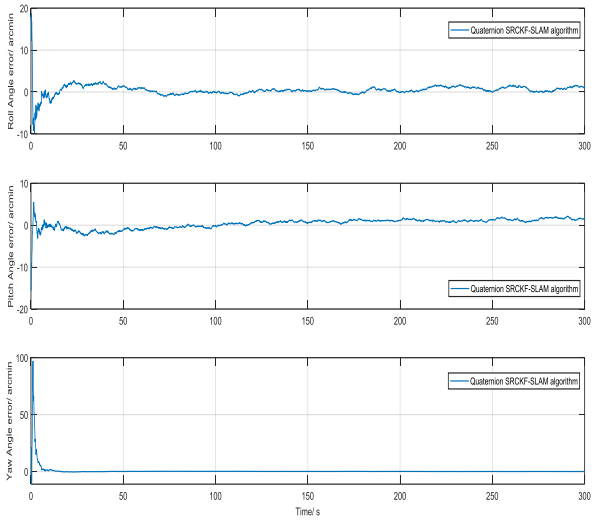


Fig. 6. Quaternion SRCKF-SLAM attitude angle estimation error

It can be seen from Figure 4 that given a large initial error attitude angle, the estimation error of the attitude angle based on the quaternion SRCKF-SLAM algorithm can be quickly reduced to about 50 in the first 50s of the simulation time, and the convergence speed is fast. And as the simulation time progresses, the filtering always remains stable, which shows that the algorithm has strong robustness and stability for large misalignment attitude angle filtering estimation.

5.3. Computer simulation experiment verification and analysis

In order to verify the effectiveness and feasibility of the SLAM algorithm proposed in this article, MATLAB simulation was conducted. The simulation environment was set up in the open-source SLAM simulator released by scholar Tim Bailey from the University of Sydney. The experimental environment is located in an outdoor environment area of 100 m ×100 m, including the actual operation path of the carrier set artificially and 10 stationary road signs. The carrier moves clockwise along the simulated and actually determined trajectory from the coordinates of (60, 10). The specific experimental environment is shown in Figure 7.

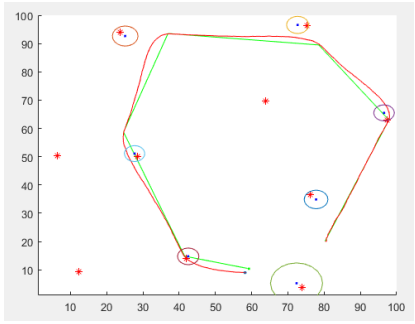


Fig. 7. Experimental environment

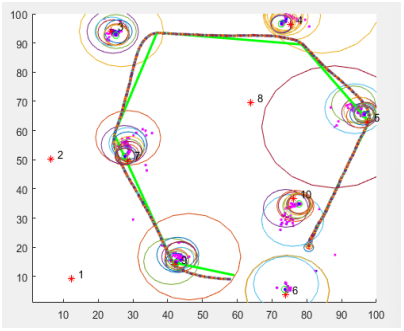


Fig. 8. SLAM feature observation

In Figure 7, the green line represents the actual navigation path of the robot, the red line represents the estimated tracking path of the robot under the SLAM algorithm, and the * point represents the 10 actual feature target objects present in the simulation area. In Figure 8, the actual observations of 10 feature targets by the SLAM algorithm in robot position tracking estimation are shown. The experimental results show that when the robot moves along the specified path, based on data fusion and observation in the SLAM algorithm, the 3rd, 4th, 5th, 6th, 7th, 9th, and 10th feature points under the navigation path are finally observed. Multiple judgments are made for each observation, and the position of the effective observation point is ultimately determined.

5.4. Analysis of cumulative error in inertial navigation systems

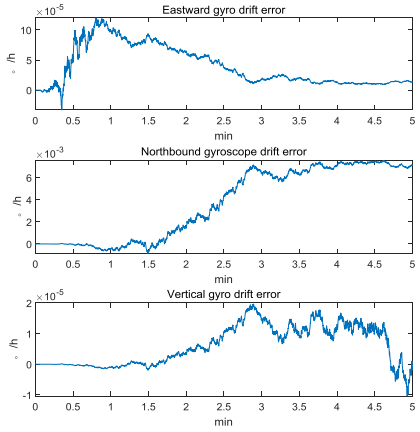


Fig. 9. Gyroscope drift error

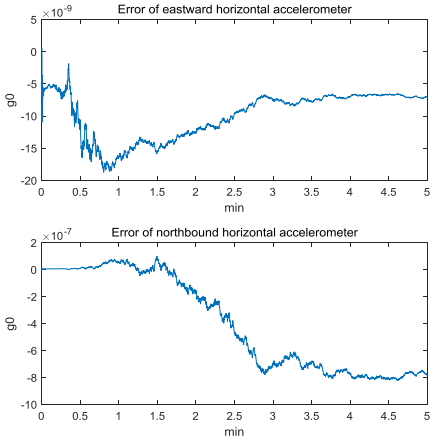


Fig. 10. Accelerometer drift error

In figure 9, the cumulative errors of the system gyroscope in three directions under inertial navigation are shown, and their mean errors are relatively large. If not combined with other navigation algorithms, this error will accumulate over time, leading to a gradual increase in carrier positioning error.

Figure 10 showed the parameter estimation of another inertial device accelerometer, and it can be seen that its error gradually accumulates over time. One integration leads to the accumulation of velocity error, while the second integration leads to the accumulation of position error. Therefore, a combination of other positioning methods must be used to weaken the above two drift errors.

5.5. Estimation of UAV position error under three filtering algorithms

Under the above experimental environment, 50 independent repeated simulation experiments were conducted on SRCKF-SLAM, SRUKF-SLAM, and SRCDKF-SLAM, and the estimation errors in the direction of the carrier path were analyzed and compared. The results are shown in the figure 11-13.

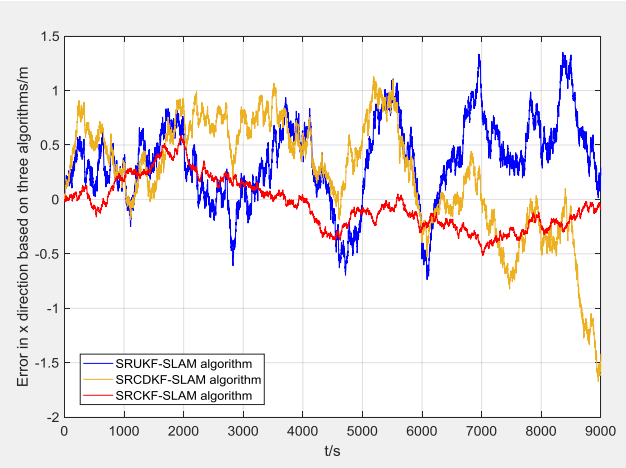


Fig. 11. Error in x direction based on three algorithms

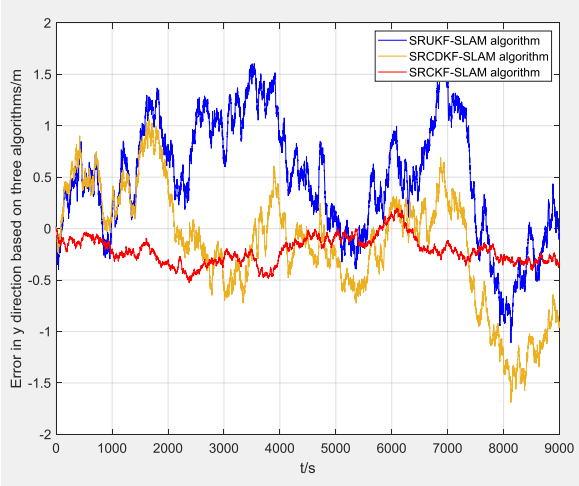


Fig. 12. Error in y direction based on three algorithms

From figure 11, it can be seen that the error range of the SRCKF-SLAM algorithm in the x-direction of the navigation trajectory of the plant protection unmanned aerial vehicle is approximately [-0.5,0.6] m. The errors of the SRUKF-SLAM algorithm and the SRCDKF-SLAM algorithm are approximately [-0.7,1.4] m and [-1.7,1.2] m. It can

be seen that the former has the smallest mean estimation error in the x-direction of the unmanned aerial vehicle, the smallest error fluctuation range, and the best stability.

From figure 12, it can be seen that the error range of the SRCKF-SLAM algorithm in the y-direction of the navigation trajectory of the plant protection UAV is approximately [-0.5,0.2] m. The errors of the SRUKF-SLAM algorithm and the SRCDKF-SLAM algorithm are approximately [-1.15,1.6] m and [-1.7,1.1] m, respectively. It can be seen that the former has the smallest mean position estimation error in the y-direction of the drone, the smallest error fluctuation range, and the best stability. The error based on the SRCDKF-SLAM algorithm is the largest, and the filtering fluctuation amplitude is also the largest, followed by the error of the SRUKF-SLAM algorithm.

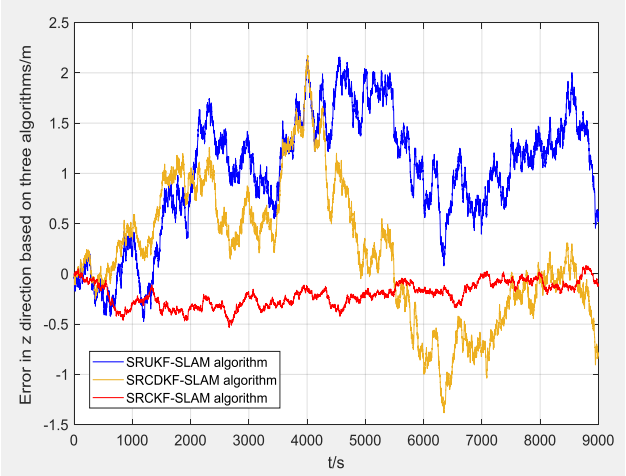


Fig. 13. Error in z direction based on three algorithms

From figure 13, it can be seen that the error range of the SRCKF-SLAM algorithm for the z-direction navigation trajectory of the plant protection UAV is approximately [-0.5,0.05] m. The errors of the SRUKF-SLAM algorithm and the SRCDKF-SLAM algorithm are approximately [-0.5,2.2] m and [-1.45,2.2] m, respectively. From this, it can be seen that the former has the smallest average position estimation error in the z-direction of the UAV, and its error positions are within a small range throughout the entire simulation time, It shows that the UAV for plant protection can perform spraying operation at a constant height at a basic position, and has high stability. The filtering algorithm has strong Uniform convergence. The error based on the SRCDKF-SLAM algorithm is the largest, and the filtering fluctuation amplitude is also the largest, followed by the error of the SRUKF-SLAM algorithm.

6. Discussion

This article mainly discusses the carrier attitude estimation based on three filtering algorithms, including quaternion based SRCKF, and provides several comparative experiments to verify that quaternion based SRCKF-SLAM algorithm has the highest

estimation accuracy, and its computational complexity is slightly reduced compared to the other two algorithms. However, more comparisons between algorithms still need to be validated. These will be subject to further research.

7. Conclusions

This article proposes a quaternion based SRCKF algorithm to address the problems of complex navigation solutions, high computational complexity, unstable filtering, and poor accuracy in traditional quaternion SRUKF and SRCDKF algorithms. The algorithm is applied to a nonlinear model of SLAM, and simulation experiments are conducted on three initial attitude angle errors under the same conditions. The results show that the new algorithm has a mean estimation error for carrier roll angle, pitch angle, and yaw angle, which are 0.05 °, 0.08 °, 0.03 °,respectively. Compared with the quaternion SRUKF-SLAM algorithm, the accuracy of the algorithm is improved by about 30%. In addition, given the large misalignment angle error, the estimation performance of the quaternion SRCKF-SLAM is examined. Experiments show that the filtering error of the algorithm can be quickly reduced to about 50 in the first 50s of the simulation time, and the convergence speed is fast. As the simulation time progresses, the filtering always remains stable, showing strong robustness and convergence.

The position error estimation experiment on the navigation trajectory of a plant protection UAV shows that the SRCKF-SLAM algorithm has the smallest mean filtering error in three directions, the smallest error fluctuation range, and the best convergence.

Although multiple filtering algorithms were compared and achieved high carrier localization accuracy in the experiments conducted throughout the article, further improvements are still needed for the algorithms mentioned in the paper. For example, combining neural networks or deep learning methods to train the volume points in the volume Kalman filter, and updating the time and measurement of parameters such as mean and covariance, in order to achieve faster, more stable, and more robust filtering results.

Acknowledgment. This research was funded by Henan Provincial Science and Technology Research Project, China (182102110295). Key Scientific Research Project of Colleges and Universities in Henan Province, China (21A590001). Key Scientific Research Project of Colleges and Universities in Anhui Province, China (2023AH051547, 2023AH051559). Key projects of Anyang Institute of technology's 2020 scientific research and Cultivation Fund, China (YPY2020001). The project of cultivating excellent young teachers of Anhui province, China (YQYB2023031). Teaching Quality Engineering Project of Anhui province, China (2023jyxm0796). Teaching Quality Engineering Project of Huainan Normal University, China (2023hskc06). Guiding Science and Technology Plan Project of Huainan Science and Technology Bureau (126).

Thank you for the funding from the Anhui Province Joint Construction Discipline Key Laboratory. Especially the support of the aircraft simulation platform, which has provided basic support for the experiment in this paper.

References

1. Zhao Y, Yang S, Jia R, et al. The statistical observation localized equivalent-weights particle filter in a simple nonlinear model. Acta Oceanologica Sinica, Vol. 41, No. 2:80-90, doi:10.1007/s13131-021-1876-1,(2022).
2. Cao H Q, Nguyen H X, Tran T N C, et al. A Robot Calibration Method Using a Neural Network Based on a Butterfly and Flower Pollination Algorithm.IEEE Transactions on Industrial Electronics,Vol. 69, No. 4:3865-3875. doi:10.1109/TIE. 2021.3073312,(2021).
3. Feng K, Li J, Zhang X, et al. A new quaternion-based Kalman filter for real-time attitude estimation using the two-step geometrically-intutive correction algorithm. Sensors, Vol. 17, No. 11:2530-1545, doi:10.3390/s17092146,(2017).
4. Min Y, Xiong Z, Xing L, et al. An improved SINS/GNSS/CNSS federal based on dual quaternions. Acta Armamentari,Vol. 39, No. 2, 315-324, doi:10.3969/j.issn.1000-1093.2018.02.014,(2018).
5. Yin S, Li H, Sun Y, et al. Data Visualization Analysis Based on Explainable Artificial Intelligence: A Survey. IJLAI Transactionss on Science and Engineering, Vol. 2, No. 3: 24–31. http://ijlaitse.com/index.php/site/article/view/34,(2024).
6. Jiang, Y., Yin, S. Heterogenous-view Occluded Expression Data Recognition Based on Cycle-Consistent Adversarial Network and K-SVD Dictionary Learning Under Intelligent Cooperative Robot Environment. Computer Science and Information Systems, Vol. 20, No. 4. https://doi.org/10.2298/CSIS221228034J,(2023).
7. Zhang H, Qin W, Zhou C, et al. Attitude Determination Algorithm for Micro-satellite Based on High-order UKF Using Information Fusion. Vol. 40, No. 6:1091-1101. doi:10.11728/cjss2020.06.1091,(2020).
8. Zhao Y, Liu Q. Causal ML:Python package for causal inference machine learning. SoftwareX, Vol. 21:101294. doi: org/10.1016/j.softx.2022.101294,(2023).
9. Jia R. Attitude estimation algorithm for low cost MEMS based on quaternion EKF. Chinese Journal Of Sensors And Acthators,Vol. 27, No. 1:90-95, doi:10.3969/j.issn.1004-1699.2014.01.017,(2014).
10. Luo X, Zhou M, Li S, et al.Non-Negativity Constrained Missing Data Estimation for High Dimensional and Sparse Matrices from Industrial Applications. IEEE transactions on cybernetics, Vol. 50, No. 5:1844-1855. doi:10.1109/TCYB.2019.2894283,(2020).
11. Wang D, Tan K, Dong Y, et al. Estimating the position and orientation of a mobile robot using neural network framework based on combined square-root cubature Kalman filter and simultaneous localization and mapping. Advances in Production Engineering & Management, Vol.15, No. 1:31-43. doi:10.14743/apem2020.1.347,(2020).
12. Xiong Y, Zhang Y, Guo X, et al. Seamless global positioning system/inertial navigation system navigation method based on square-root cubature Kalman filter and random forest regression. Review of Scientific Instruments, Vol.90, No.1:015101, doi: 10.1063/1.5079889,(2019).
13. Li Y, Fu K, Sun H. An Aircraft Detection Framework Based On Reinforcement Learning and Convolutional Neural Networks in Remote Sensing Images. Remote Sensing, Vol.10, No.1:243-262. doi: 10.3390/rs10020243,(2018).
14. Dai, Y., Xu, B., Yan, S., Xu, J. Study of cardiac arrhythmia classification based on convolutional neural network. Computer Science and Information Systems, Vol. 17, No. 2, 445–458. https://doi.org/10.2298/CSIS191229011D, (2020).
15. Wang D, Tan K, Li Z, et al. Research on Landmarks of SLAM Based on Square Root Cubature Kalman Filter. 3rd Annual International Conference on Information System and Artificial Intelligence (ISAI), 1037-1045. doi:10.1088/1742-6596/1069/1/012154,(2018).
16. Wang D, Tan K, Li H. Research on feature extraction method based on Simultaneous localization and mapping. China Control Conference (CCC), 3720-3724. doi:10.23919/ChiCC.2018.8482972,(2018).

17. Aydemir, F., Cetin, A. Point of Interest Coverage with Distributed Multi-Unmanned Aerial Vehicles on Dynamic Environment. Computer Science and Information Systems, Vol. 20, No. 3:1061–1084. https://doi.org/ 10.2298/CSIS221222037A,(2023).
18. Wang, Y., Han, D., Cui, M. Intrusion Detection Model of Internet of Things Based on Deep Learning. Computer Science and Information Systems, Vol. 20, No. 4. https://doi.org/10.2298/CSIS230418058W,(2023).
19. Zhang D , Shafiq M, Wang L,et al. Privacy-preserving remote sensing images recognition based on limited visual cryptography. CAAI Transactions on Intelligence Technology, Vol. 8, No. 4:1166-1177. https://doi.org/10.1049/cit2.12164,(2023).

Dandan Wang is a PhD candidate currently working at Huainan Normal University, with a main research focus on SINS and nonlinear filtering algorithms. Her five projects have received funding support from the provincial government, while her other projects have received funding support from other departments. She has published 10 SCI/EI papers and possesses rich teaching and strong research capabilities.

Zhaokun Zhu is a lecturer currently working at Zhengzhou Technology and Business University.His main research direction is electronic and communication engineering. In this article, he is mainly responsible for setting up the experimental environment.

Liang Yu is a lecturer currently working at Huainan Normal University, with a main research focus on mechanical design. One of his projects has received funding support from the provincial government and has been granted five patents. He has rich teaching experience.

Hongjie Li is a lecturer currently working at Anyang Institute of Technology. His main research direction is electronic technology, and he usually helps the college handle some academic affairs.

Kaituo Tan is a research assistant currently working at Huainan Normal University. His main research direction is automation technology. In this article, he is mainly responsible for experimental statistics and processing.

Received: April 18, 2024; Accepted: May 21, 2024.

Computer Science and Information Systems 21(4):1823–1843 https://doi.org/10.2298/CSIS240418060W

Spatio-Temporal-based Multi-level Aggregation Network for Physical Action Recognition

Yuhang Wang

School of Physical Education, Harbin University
Harbin, Heilongjiang 150086, China
wangyuhang1978@hrbu.edu.cn

Abstract. This paper introduces spatio-temporal-based multi-level aggregation network (ST-MANet) for action recognition. It utilizes the correlations between different spatial positions and the correlations between different temporal positions on the feature map to explore long-range spatial and temporal dependencies, respectively, generating the spatial and temporal attention map that assigns different weights to features at different spatial and temporal locations. Additionally, a multi-scale approach is introduced, proposing a multi-scale behavior recognition framework that models various visual rhythms while capturing multi-scale spatiotemporal information. A spatial diversity constraint is then proposed, encouraging spatial attention maps at different scales to focus on distinct areas. This ensures a greater emphasis on spatial information unique to each scale, thereby incorporating more diverse spatial information into multi-scale features. Finally, ST-MANet is compared with existing approaches, demonstrating high accuracy on the three datasets.

Keywords: Action recognition, spatial and temporal attention, multi-level aggregation network.

1. Introduction

With the development of information technology, videos have become an indispensable medium in our lives due to their powerful content delivery and expressive capabilities. The widespread deployment of image capture devices and the growth of computer networks have led to an explosive increase and dissemination of videos. These massive video datasets hold significant value, but traditional manual analysis is hindered by subjective factors and inefficiency. Therefore, utilizing computer technology for the analysis, understanding, and information mining of video content is a crucial topic in the field of computer vision and has been a forefront direction of research in recent years. Due to various factors such as different video data collection methods, diverse scenes, and varied content, designing manual features for algorithms is time-consuming and challenging to apply widely across multiple tasks. Deep learning technologies have undergone rapid development on computer vision, face recognition, and semantic segmentation, demonstrating superiority compared to traditional algorithms [2,6,15].

In human action recognition, behavior instances exhibit a complex temporal structure, especially with different behavior categories naturally having distinct visual rhythms, such as clapping and walking. This poses challenges for behavior recognition [13,11,10]. In certain situations, different behavior categories may look very similar in appearance, such

as walking, jogging, and running. The visual information of these behaviors is closely related, making accurate classification challenging based solely on appearance. In such cases, visual rhythm becomes a crucial factor in distinguishing them. Moreover, even for the same behavior, individual differences in factors like age, energy, and mood mean that each performer may execute these actions according to their own rhythm. For instance, older individuals typically move more slowly than younger ones, and individuals with greater weight generally exhibit slower movements than those with lesser weight. Therefore, precisely modeling the intra-class differences in the visual rhythm of action behaviors significantly enhance the performance of behavior recognition. In recent years, researchers have begun to explore the aspect of visual rhythm. The SlowFast Network is one such example, consisting of two pathways known as the Fast pathway and the Slow pathway [5]. It incorporates hard-coded visual rhythm at the input level, enabling the two pathways to sample frames at different rates. At each stage of the network, there is a lateral connection between the two pathways, facilitating the fusion of features from the Fast pathway into the Slow pathway. However, this hard-coded approach requires more input frames, and the temporal scale is limited to manually set and tuned values. While employing more pathways may enhance performance, it also comes with a significant increase in computational complexity, potentially making the network overly intricate. Considering a lot of utilized methods, i.e., C3D [22] and I3D [1], usually involve many operations of temporal convolution. That is, as the network depth increases, the temporal receptive field also expands. Therefore, multi-depth features in a method can be captured information covering various visual rhythms and possess different temporal receptive fields, indicating diverse temporal scales. Additionally, these features at different depths also encompass multi-scale spatial information. Meanwhile, the success of multi-scale features has been demonstrated in tasks like object detection [8] and pedestrian re-identification [9]. Feature extraction in deep CNNs is a process that transforms features from low levels into high levels. While deeper network layers can extract global information with highly discriminative features, they inevitably lose detailed information. Shallow features, on the other hand, maintain high resolution and weak semantics, serving as a complement to deep features.

Inspired by these considerations, we introduce a spatio-temporal-based multi-level aggregation network (ST-MANet) for action recognition. Specifically, ST-MANet, based on the self-attention mechanism, designs a temporal aggregation module to leverage the temporal correlations within videos. It aims to extract more temporal-related information and aggregate temporal information effectively. Subsequently, a temporal aggregation network is constructed based on the temporal aggregation module, consisting of two parts: the main branch and the temporal aggregation branch. The interaction between these branches further explores temporal information in the video, enhancing the network's temporal modeling capabilities and improving the accuracy of behavior recognition. Following this, a spatial attention module based on the self-attention mechanism is introduced. It evaluates the correlations between different spatial positions in the feature map, assessing the importance of each position and assigning different weights to different spatial locations. Furthermore, a multi-scale approach is incorporated, proposing a multi-scale behavior recognition framework that models various visual rhythms while capturing multi-scale spatiotemporal information. On this basis, a spatial diversity constraint is introduced to focus spatial attention at different scales on distinct spatial locations. This

ensures that different scales of spatial attention concentrate on spatial information unique to each scale, allowing for a more diverse representation of spatial information in multi-scale features. Finally, experimental results demonstrate the effectiveness of ST-MANet compared to existing approaches, showcasing high accuracy across three datasets.

The contributions of ST-MANet are threefold:

(1) A temporal aggregation module, designed through the self-attention mechanism, effectively extracts and aggregates temporal information from videos, significantly enhancing the network's ability to capture action sequence changes and thereby improving the accuracy of behavior recognition.

(2) By introducing a spatial attention module and a multi-scale approach, ST-MANet is capable of capturing unique spatial information at different scales, and through the spatial diversity constraint, ensures the comprehensiveness and diversity of this information, further strengthening the model's ability to recognize actions in complex scenes.

(3) Experimental results demonstrate the effectiveness of ST-MANet compared to existing approaches, showcasing high accuracy across three datasets in action recognition tasks.

The remaining sections of ST-MANet are organized as follows: Section II offers an extensive review of current methods in human action recognition. Section III detail the key components and the methodology behind ST-MANet. Section IV presents a detailed account of the experimental results and evaluations conducted to validate the efficacy of ST-MANet. Finally, Section VI concludes the paper by summarizing the key findings and contributions of ST-MANet.

2. Related Works

2.1. Dual-stream based methods

Dual-stream based methods incorporate both spatial and temporal information, which enhances behavior recognition in video analysis. Karpathy et al. propose the dual-stream network architecture to extract action features at different resolutions in two streams, extending convolutional neural networks into the realm of videos [14]. Simonyan et al. introduce the use of optical flow information to extract temporal features and provide a new direction for action recognition, which divides the model into two parts, with one part processing individual video frames for spatial information extraction, and the other part taking dense optical flow from consecutive frames for temporal feature extraction [19]. The classification results from both streams are combined with the Softmax to capture final action recognition results. Although the dual-stream network utilizing complex optical flow achieved superior results on public datasets at the time, it still had room for improvement as the network only utilized a small portion of information from the entire video. Building upon this, Wang et al. design a network based on the temporal segment (TSN) [26]. In the data processing stage, TSN divides video data into multiple segments. Each segment undergoes feature recognition based on the dual-stream network. The spatial and temporal stream scores from each segment are then aggregated to obtain the overall spatial and temporal scores for the entire video. Finally, the two scores are fused to yield the ultimate result. TSN effectively utilizes video information by uniformly extracting action features across different slices of the entire video, enhancing recognition accuracy.

However, TSN's fusion of information from different segments is only evident in the model's final stage, simply consolidating recognition results without capturing more nuanced temporal information from the feature extraction results. Sun et al. propose a novel LSTM structure and multimodal training approach to train networks [20]. They employed two streams to jointly train the input and optical flow gates in the network, departing from traditional methods that treated the two streams as independent training networks. Tran et al. introduce 3D CNNs in video recognition, asserting that 3D convolution kernels are more suitable for temporal features in video data [22]. They conducted tests on the optimal size for the third dimension (depth) based on Simonyan et al.'s recommended 3×3 2D convolution kernel and provided the most suitable three-dimensional convolution kernel and network architecture for video convolution. Carreira et al. present a dual-stream Inflated 3D ConvNet (I3D) based on dilated 2D convolutions [1]. They applied 3D convolutions in both streams, expanding the design and parameters of deep image classification convolutional networks seamlessly for video spatiotemporal feature extraction.

Dual-stream based methods have been a cornerstone in the field of action recognition due to their ability to capture both spatial and temporal information from videos. The spatial stream typically processes individual frames to extract features from the visual content, while the temporal stream exploits optical flow to analyze motion patterns over time. This dual approach has proven to be highly effective in distinguishing complex actions and interactions. However, the process of extracting optical flow is computationally intensive. It involves estimating the motion of objects between consecutive frames, which can be particularly challenging in videos with large motion variations, occlusions, or changes in illumination. Despite the significant progress in developing efficient optical flow algorithms, the computational demands remain a bottleneck for real-time applications and deployment on resource-constrained devices.

2.2. Behaviour rhythm based methods

Behaviour rhythm based methods aim to leverage the temporal structure and variations of actions within video sequences to enhance the performance of action recognition in videos. Wang et al. point out that the spatial stream of the dual-stream network is influenced by samples with similar backgrounds, and the temporal stream struggles to differentiate actions with similar trajectories in a short period [27]. Moreover, the two streams fail to deeply integrate spatial and temporal features. To address these issues, they proposed the Spatiotemporal Pyramid Network. In the temporal module, fixed-interval sampling is applied to the optical flow of the entire video, with the sampling intervals randomly determined to obtain features under different action rhythms. Zhang et al. introduce the Dynamic Temporal Pyramid Network (DTPN), based on multiscale modeling [30]. They directly employed inputs with five different random frame rates to extract features, designed a dual-branch multiscale temporal feature structure to handle intrinsic time scale differences in actions, and created a fusion module for different scale feature maps. This demonstrated the significance of both local and global features for video recognition. Feichtenhofer et al. propose the SlowFast network, training two different structures with data at two frame rates, sampling input frames with a larger interval to extract spatiotemporal features [5]. The latter, at a higher frame rate, samples frames with a smaller time gap on the same video segment. Similar to dual-stream networks, a fusion mechanism

is employed to integrate information from both pathways to obtain the final classification result. Yang et al. introduce the concept of visual rhythm and statistically analyzed the visual rhythms of different actions [28]. They proposed the Temporal Pyramid Network (TPN), which samples image features at different ratios to extract hierarchical action rhythms. A feature aggregation module is utilized to fuse the rhythm features. Inspired by machine learning feature selection methods, Facebook AI Research introduced the X3D extension structure for efficient video recognition [4]. This structure expands the model in three dimensions based on a 2D convolutional network, providing a meaningful reference example for further extension of efficient video recognition models.

The behavior rhythm-based methods for video action recognition represent a significant advancement in the field by focusing on the temporal dynamics of actions within a video sequence. These methods analyze the variations in human behavior over time to capture the rhythm and patterns that define different actions. By sampling videos at multiple rates and at different time intervals, these approaches can extract a rich set of features that encapsulate the nuances of human motion, leading to more accurate recognition of actions. However, the computational expense of such an approach is a significant drawback. The need for multiple video samplings and the subsequent processing to extract features at various rates results in a substantial increase in the data volume that needs to be handled. This not only slows down the training process but also requires substantial computational resources, which can be a limiting factor for large-scale video analysis.

2.3. Differential feature based Methods

This type of method directly acquires action features by utilizing the differential information between adjacent data points and features, significantly reducing the computational load involved in data processing and model training. Differential calculations are divided into RGB differential and feature differential. RGB differential, resembling optical flow in representing motion information, can capture certain action details, and its extraction method is often more efficient. It is frequently used to complement spatial features extracted based on RGB frames. For example, in the Temporal Segment Network (TSN) [26], differential feature extraction is added to the original image features in the spatial stream, and experimental results demonstrate the effectiveness of their combined features for video action recognition. Lin et al. propose the Temporal Shift Module (TSM), suitable for spatiotemporal feature extraction, where features of certain channels are temporally shifted to fill gaps in the time dimension, achieving an effect similar to extracting differential features [17]. Liu et al. introduced TEINet, which includes an action enhancement module and a time-increasing interaction module [18]. The action enhancement module represents motion information by computing the differential feature maps of adjacent frames, and the time module further performs convolution operations on features channel-wise, capturing temporal information. Jiang et al. propose a channel-wise action module for action feature extraction [12]. It calculates the differential features between different dimensional channel features, connects the differential results of each channel, and applies 2D convolution to obtain the final action feature information. Li et al. presented a temporal excitation and aggregation network, similar to the aforementioned methods, with the distinction that the aggregation module divides features into multiple slices [16]. Each slice's features undergo convolutional operations before subtraction. The action extraction module also conducts differential calculations on the features in the initial network layer,

enhancing the model's video recognition performance via multiple uses of differential features. In the temporal difference network proposed by Wang et al., appearance features are extracted in the lower-level network, introducing the short-term temporal difference module [25]. Meanwhile, temporal features are extracted in the higher-level network, introducing the long-term temporal difference module. Both modules employ differential techniques for RGB frames and features separately.

Despite the notable achievements in video action recognition through the use of differential information for feature extraction, these methods encounter several challenges. The RGB differential, which excels at capturing motion details, may fall short when it comes to complex actions that require a deeper understanding of semantic information. Additionally, while differential methods effectively capture local changes, they may struggle with long-term dependencies in videos, such as those addressed by the Temporal Segment Network (TSN). The challenge of effectively fusing features from multiple modules within network architectures like TEINet and others is significant, as is the design of these fusion strategies, which are critical to the final recognition performance but prone to information loss or redundancy. Computational resource optimization remains an issue, especially when dealing with large-scale video data, despite the computational benefits offered by differential calculations. The modular design of networks, although successful for specific tasks, may have limited generalization capabilities when applied to other action recognition tasks. Furthermore, the short-term and long-term temporal difference modules introduced in temporal difference networks, while handling features at different time scales, may not adapt well to rapid or nonlinear motion dynamics in videos. Lastly, the performance of these methods is heavily dependent on the quality and diversity of the training datasets, with biases or insufficiencies potentially leading to reduced generalization capabilities. In summary, while differential information provides valuable insights for action feature extraction, a range of challenges must be addressed to ensure robust and generalizable models in practical applications.

3. Spatio-Temporal-based Multi-level Aggregation Network for Action Recognition

A spatio-temporal-based multi-level aggregation network (ST-MANet) is proposed for action recognition, which contains the baseline network, the temporal-based deep aggregation network, and the spatio-based deep aggregation network. The main mathematical notations used in the paper are listed in Table 1.

3.1. Baseline Network

ST-MANet utilizes 3D ResNet-50 as the baseline network for human action recognition, which is composed of five stages, as shown in Fig. 1.

Specifically, stage 1 of 3D ResNet-50 consists of a convolutional layer, a time step of 1, and a spatial step of 2, followed by BN and ReLU layers in sequence, where the convolutional layer extracts features and reduces spatio-temporal dimensions. The Batch Normalization layer prevents the occurrence of gradient vanishing and accelerates network convergence. The ReLU layer enhances the expressive power of the extracted features.

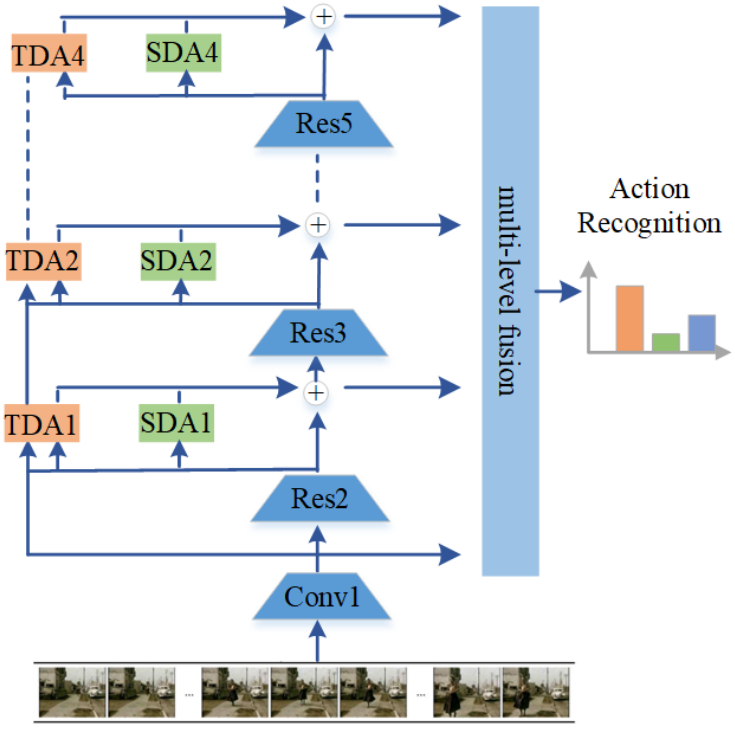


Fig. 1. The illustration of the proposed ST-MANet

Table 1. Frequently used notations

Notations	Description
Z^i	the feature map from the i-th stage of 3D ResNet-50
Z_{TDA}^{i-1}	the feature map from the i-th stage of temporal-based deep aggregation network
Z_c^i	the concatenation feature of Z^i and Z_{TDA}^{i-1}
A_t^i	the temporal attention matrix
A_s^i	the spatio attention matrix
$\bar{Z}_t^i$	the temporally aggregated feature map
$\bar{Z}^i$	the final output of the i-th stage of 3D ResNet-50
$f(\cdot)$	the cosine similarity function
$S^i(\cdot)$	the downsampling function
$Con(\cdot)$	the fusion function
L_{ce}	the cross-entropy loss
L_s	the spatial diversity constraint
L_t	the temporal diversity constraint
λ_1,λ_2	the balance coefficients

The subsequent four stages contain 3, 4, 6, and 3 Bottleneck Residual Blocks, respectively, where each block is a concatenation of three convolutional blocks. The first convolutional block consists of a convolutional layer, BN layer, and a ReLU layer. The output channel number is the same as the input channel number, aiming to promote interaction between different channels. The second convolutional block consists of a convolutional layer, a BN layer, and ReLU layer. The channel number is again the same as the input channel number, intended to expand the spatio-temporal receptive field of the network. The third convolutional block consists of a convolutional layer, a BN layer, and a ReLU layer. The output channel number is four times the input channel number. Each bottleneck residual block is constructed with a residual structure, stacked together to form the different stages of 3D ResNet-50. The feature space and temporal resolution of the output from each stage are reduced by a factor of two, while the channel dimension is doubled, giving the network powerful abstract feature extraction capabilities.

3.2. Temporal-based Deep Aggregation Network

Considering the varying contributions of video frames at different time points to action recognition, ST-MANet employs a visual self-attention mechanism to construct a temporal-based deep aggregation network (TDA), aiming to enhance the capturing of temporal information in action videos.

Specifically, TDA employs a dual-branch design, with inputs originating from the ResNet-50 branch and the temporal-based deep aggregation network branch, which is shown in Fig. 2. The ResNet-50 branch retains the relative temporal relationships of the original input and transfers information to the temporal-based deep aggregation network branch. TDA makes full use of a self-attention mechanism to aggregate and enhance the mining of temporal information about actions, which not only is temporal information reinforced, but also contributes to the extraction of holistic features.

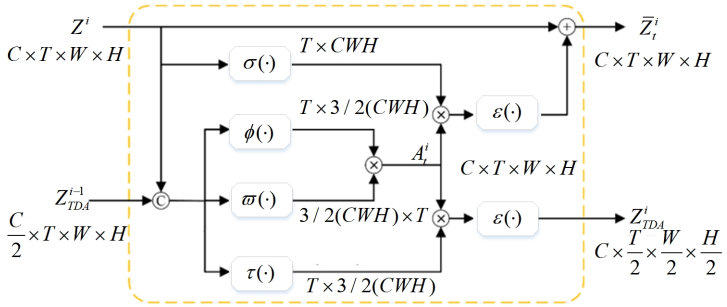


Fig. 2. Temporal-based deep aggregation network

More specifically, given the feature map from the i-th stage of 3D ResNet-50, $Z^i \in \mathbb{R}^{C\times T\times W\times H}$, and the feature map from the previous temporal-based deep aggregation network, $Z_{TDA}^{i-1} \in \mathbb{R}^{C/2\times T\times W\times H}$, as the input of the i-th temporal-based deep aggregation network, where C is channel number. T, W, and H are the time steps, the width, and

the height, respectively. TDA concatenates Z^i and Z^{i-1}_{TDA} along the channel dimension to obtain the concatenation feature $Z^i_c \in \mathbb{R}^{3C/2\times T\times W\times H}$, and then computes the temporal attention matrix A^i_t via:

$$A^i_t = softmax(\phi(Z^i_c) \otimes (\varpi(Z^i_c)) \tag{1}$$

where $\phi(\cdot)$ and $\varpi(\cdot)$ denote reshaping operations to obtain the feature map with the size of $(T \times 3/2(CWH))$ and $(3/2(CWH) \times T)$, respectively. The elements in A^i represent the degree of correlation between features at each moment in time. Moments with strong correlations to other moments in the feature map have higher time attention scores, and vice versa. It is worth noting that the convolutional kernel sizes used here are $3 \times 1 \times 1$, which, compared to the commonly used $1 \times 1 \times 1$ kernels in visual self-attention mechanisms, have a larger size along the time dimension. This results in a larger temporal receptive field, providing stronger temporal modeling capabilities.

Next, the feature map Z^{i-1}_{TDA}, after the $\tau(\cdot)$ operation, is multiplied element-wise with the temporal attention matrix A^i_t, normalized using the softmax function, and further processed by the $\epsilon(\cdot)$ operation to obtain the temporally aggregated feature map Z^i_{TDA}:

$$Z^i_{TDA} = \epsilon(softmax(A^i_t \otimes \tau(Z^{i-1}_{TDA}))) \tag{2}$$

where $\tau(\cdot)$ denote the reshaping operation to obtain the feature map with the size of $(T\times 3/2(CWH))$. $\epsilon(\cdot)$ first reshapes the feature map back to a size of $3C/2\times T\times W\times H$, then passes it through a convolutional layer with a kernel size of $3 \times 1 \times 1$, followed by a pooling layer to output the temporally aggregated feature X^i_{TDA}. It is worth noting that for the final TDA, $\epsilon(\cdot)$ does not include a pooling layer to ensure that the ResNet-50 branch and the temporal-based deep aggregation network branch outputs have the same dimensions for subsequent feature fusion.

Similarly, after the $\sigma(\cdot)$ operation, the feature map Z^i is multiplied element-wise with the attention map A^i_t. Subsequently, it undergoes the softmax activation function and the $\epsilon(\cdot)$ operation, and then is element-wise added with the X^i to obtain the temporally aggregated feature map $\bar{Z}^i_t$:

$$\bar{Z}^i_t = Z^i + \epsilon((softmax(A^i_t \otimes \sigma(Z^i)))) \tag{3}$$

where $\sigma(\cdot)$ denotes the reshape operation to obtain the feature map with the size of $(T \times CWH)$.

TDA uncovers additional temporal correlation information within videos, exploring long-range dependencies over time and consolidating temporal information. Notably, at each stage, the output from the upper part of the temporal aggregation module is integrated into the main branch, reinforcing temporal information in the primary pathway. The introduction of the temporal aggregation module enhances the temporal modeling capabilities of 3D ResNet-50. Additionally, the interaction between the two branches delves into more information within the video.

3.3. Spatio-based Deep Aggregation Network

To focus on regions containing more discriminative information, ST-MANet designs the spatio-based deep aggregation network(SDA) to calculate the correlation between different spatial positions in the feature map, as illustrated in Fig. 3.

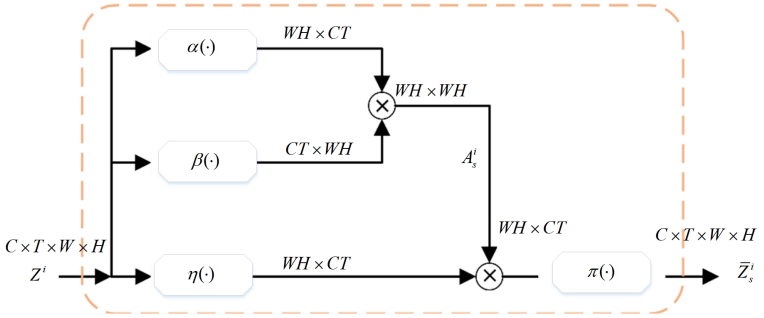


Fig. 3. Spatio-based deep aggregation network

Specifically, given the feature map from the i-th stage of 3D ResNet-50, $Z^i \in \mathbb{R}^{C\times T\times W\times H}$, as the input of the i-th spatio-based deep aggregation network, SDA computes the spatio attention matrix A_s^i via:

$$A_s^i = softmax(\alpha(Z^i) \otimes \beta(Z^i)) \tag{4}$$

where $\alpha(\cdot)$ and $\beta(\cdot)$ denote reshaping operations to obtain the feature map with the size of $(WH \otimes CT)$ and the size of $(CT \otimes WH)$, respectively. The elements in A_s^i indicate the correlation between features at each spatial position in Z^i and features at other spatial positions. In other words, positions with strong correlations to other spatial positions in the feature map have higher spatial attention scores, and vice versa.

Then, the spatially enhanced feature map $\bar{Z}_s^i$ is obtained from:

$$\bar{Z}_s^i = \pi(softmax(A_s^i \otimes \eta(Z^i))) \tag{5}$$

Where $\eta(\cdot)$ and $\pi(\cdot)$ denote reshaping operations to obtain the feature map with the size of $(WH \otimes CT)$ and the size of $C \times T \times W \times H$, respectively.

SDA unveils additional spatial correlation information within images, exploring intricate dependencies across different regions and consolidating spatial information. Notably, in each processing stage, the output from the upper part of the spatial aggregation module is seamlessly integrated into the main branch, fortifying spatial information in the primary pathway. The incorporation of the spatial aggregation module elevates the spatial modeling capabilities of the 3D ResNet. Moreover, the interaction between the two branches delves into a richer trove of information within the image. This synergistic approach enhances explorations in nuanced spatial relationships and dependencies.

3.4. The multi-level fusion loss of ST-MANet

Convolutional neural networks generate hierarchical feature maps through repeated pooling and subsampling operations. As the hierarchy goes deeper, the receptive field becomes larger, indicating stronger representational capabilities but leading to the loss of detailed spatial information. In other words, deep features in convolutional neural networks possess high-level visual perception but discard some crucial local information. Shallow layers, on the other hand, exhibit higher spatial resolution features, preserving fundamental

details of the image but lacking semantic understanding. To enhance the overall expressive power of the network, ST-MANet introduces temporal and spatial diversity constraints for different depths, which aims to encourage attention matrices at different depths to focus on distinct temporal and spatial information.

Considering that temporal and spatial attention matrices also have different dimensions, a downsampling operation is applied to the attention matrices of shallower layers to ensure that all attention matrices have the same size:

$$\bar{A}^i = S^i(A^i), i = 2, 3, 4, 5 \tag{6}$$

where i represents the i-th stage of ResNet-50 and $\bar{A}^i$ represents the attention matrix after downsampling for the i-th stage of 3D ResNet-50. $S^i(\cdot)$ is the downsampling function of the i-th stage of ResNet-50. Then, ST-MANet minimizes the cosine similarity between the aligned attention matrices to maximize the differences at different network depths in terms of time and space:

$$\begin{aligned} L_t &= \sum_{(i,j)\in\delta} f(\bar{A}_t^i, \bar{A}_t^j) \\ L_s &= \sum_{(i,j)\in\delta} f(\bar{A}_s^i, \bar{A}_s^j) \end{aligned} \tag{7}$$

where L_s and L_t represent the spatial and temporal diversity constraints, $\delta = \{(i, j) \mid i, j \in \mathbb{Z}, 2 \leq i, j \leq 5, i \neq j\}$, where i and j denote the stages of the network. This constraint is computed only for stages 2 and onwards. $f(\cdot)$ denotes the cosine similarity function. During training, the spatial and temporal diversity constraints tend to decrease, encouraging a reduction in the cosine similarity of the spatial and temporal attention maps across different stages. This reduction further contributes to compelling the network to focus on distinct the spatial and temporal locations at various stages. This enhances the diversity of the spatial and temporal information within multi-scale features.

Meanwhile, ST-MANet fuses four feature maps of different depths by cascading along the channel dimension to obtain a comprehensive and good fusion representation Z for the classification of action recognition;

$$Z = Con(Z^2; Z^3; Z^4; Z^5) \tag{8}$$

where $\bar{Z}^i = Z^i \oplus \bar{Z}_t^i \oplus \bar{Z}_s^i$ denotes final output of the i-th stage of 3D ResNet-50, which fully aggregates spatio-temporal information in each layer.

The multi-level fusion loss of ST-MANet is defined:

$$L = L_{ce} + \lambda_1 L_s + \lambda_2 L_t \tag{9}$$

where L_{ce} is the cross-entropy loss. L_s is the spatial diversity constraint. L_t is the temporal diversity constraint. λ_1 and λ_2 are the balance coefficients.

4. Experiments

4.1. Set up

Dataset and metric: The performance of the ST-MANet in action recognition is verified based on three popular datasets, i.e., UCF-101, HMDB-51, and Diving 48. The comprehensive statistical details for each dataset are provided in Table 2. Building upon prior

research, the evaluation of performance relies on accuracy as the primary metric, with higher values indicating superior performance.

- UCF-101 is a coarse-grained description of the action recognition. It contains 13320 action videos spanning 101 distinct classes.
- HMDB-51 is a coarse-grained description of the action recognition. It contains 6849 action videos spanning 51 distinct classes.
- Diving 48 is a fine-grained description of the action recognition. It contains 18404 action videos spanning 48 distinct classes.

Table 2. Statistical Details For Each Dataset

Dataset	Sample	Class	Type
Ucf-101	13320	101	Coarse-Grained Description
Hmdb-51	6849	51	Coarse-Grained Description
Diving 48	18404	48	Fine-Grained Description

Implementation Details: ST-MANet conducts preprocessing and data augmentation on the input video data. Initially, it randomly samples continuous video segments of 64 frames from the videos. Subsequently, the video segments undergo subsampling with a stride of 2, resulting in frame sequences of 32 frames. Next, random cropping is applied to the frames, with the length and width set between 0.5 and 1 times the length of the shorter side of the original video frame. Finally, a spatial scale transformation is applied to resize both the length and width of the frame sequences to 112. To increase training samples, random horizontal flipping is employed as a data augmentation technique. During the training process, the paper utilizes batch SGD to train the ST-MANet, employing cross-entropy loss function with a momentum coefficient of 0.9 and 32 batch size. The network parameters are initialized using the Kaiming initialization strategy. The initial learning rate is set at 0.01, and every 15 epochs, the learning rate is decayed by a factor of 0.1. The entire training process spans 90 epochs. In contrast to the preprocessing of data before training, during testing, the entire video is first divided into multiple continuous video segments of 64 frames each. These segments then undergo subsampling with a stride of 2, resulting in multiple video frame sequences of 32 frames. Subsequently, central cropping is applied to each frame sequence. Finally, multiple frame sequences of a video are input into the network.

4.2. Comparison with baselines

Comparison methods: Eight baseline methods are compared to evaluate the performance of ST-MANet in the experiments, including TCNet [19], LFSNet [23], TSNet [26], LTNet [24], MRNet[7], RMRNet [21], MVTNet [3], and A3DNet [29].

Comparison results: Table 3 presents a comparison of experiment outcomes regarding accuracy on the three few-shot datasets. Specifically, ST-MANet outperforms eight baseline methods across both settings, which proves effectiveness and superiority in the action recognition. The seasons are fourfold. (1) ST-MANet designs temporal-based deep

Table 3. Comparison of ST-MANet to top-performing methods on three datasets in terms of the accuracy

Method	UCF-101	HMDB-51	Diving 48
TCNet	0.8200	0.5940	0.5222
LFSNet	0.8580	0.5492	0.5645
TSNet	0.8640	0.5881	0.5987
LTNet	0.9270	0.6720	0.6719
MRNet	0.8440	0.5900	0.5904
RMRNet	0.9418	0.6275	0.6668
MVTNet	0.9040	0.5460	0.6048
A3DNet	0.9260	0.6126	0.6521
ST-MANet	**0.9605**	**0.6872**	**0.7215**

aggregation network to enhances the temporal information in the 3D RestNet-50 and incorporates a temporal aggregation branch for exploring and aggregate additional temporal information. (2) ST-MANet designs the spatial-based deep aggregation network to strengthen discriminative spatial information relevant to action recognition. (3) ST-MANet introduces a multi-level fusion strategy to model various visual rhythms and explore multi-scale temporal and spatial information concurrently. (4) By introducing temporal and spatial diversity constraints, ST-MANet encourages temporal and spatial attention at multiple scales to focus on different positions, ensuring that the ultimately fused multi-scale features contain more comprehensive and complete spatiotemporal information.

In addition, there are four observations. (1) TCNet and TSNet, while leveraging the powerful dual-stream architecture, struggle with capturing long-term temporal dynamics due to their focus on optical flow between adjacent frames. This limitation suggests the need for more advanced mechanisms that can handle temporal dependencies over larger time spans. Furthermore, their spatial stream networks, which are crucial for understanding the context and structure within each frame, could benefit from more innovative designs to better extract relevant spatial features. (2) LFSNet's adoption of the C3D network, although it offers a straightforward and efficient architecture, falls short in delving deeper into the intricate spatiotemporal relationships present in videos. LTNet's enhancement to include more frames is a step in the right direction, yet the network's overall simplicity may not be sufficient to fully harness the rich information encoded in video sequences. This points towards the necessity for more complex and sophisticated network structures that can integrate information from multiple frames effectively and robustly. (3) MRNet's focus on refining the GRU for temporal exploration is commendable, but the inherent challenges in extracting spatial information using RNN-based methods cannot be overlooked. ST-MANet's integration of a spatial attention module with 3D ResNet-50 represents a significant advancement, as it allows the network to adaptively focus on relevant regions within the frame, leading to a more nuanced understanding of the spatial content. This underscores the importance of attention mechanisms in enhancing the model's ability to discern and prioritize spatial information. (4) RMRNet's investigation into residual frames is an innovative approach to bolster the motion capture capabilities of 3D convolutional networks. However, the lack of structural innovation and the reliance on 3D

ResNet-18 for temporal feature extraction may limit the network's overall performance. This indicates that while residual frames can offer valuable insights into motion, the underlying model architecture must also be optimized to ensure comprehensive temporal analysis. In summary, these observations reveal that while current networks have made strides in video action recognition, there is still much room for improvement. The key lies in striking a balance between the complexity of the network, which allows for thorough spatiotemporal feature extraction, and the efficiency required for practical applications. Future research should focus on developing architectures that can effectively handle long-term temporal dependencies, innovate in spatial feature extraction, and integrate attention mechanisms to refine the model's focus on critical information. By addressing these aspects, we can expect significant advancements in the field of video action recognition, leading to more accurate and efficient models capable of understanding and analyzing the rich content of video data.

4.3. Ablation Study

Table 4. Ablation experiments of each component in ST-MANet

3D ResNet-50	TDA	SDA	MF	Accuracy
√				0.8317
√	√			0.8945
√		√		0.9012
√	√	√		0.9278
√			√	0.8416
√	√	√	√	0.9605

We conducts five ablation experiments about 3D ResNet-50, TDA (temporal-based deep aggregation network), SDA (spatial-based deep aggregation network), and MF (multi-level fusion strategy). Specifically, (1) ST-MANet utilizes 3D ResNet-50 to extract features for performing action recognition tasks. (2) ST-MANet intergrades 3D ResNet-50 and the temporal-based deep aggregation network to extract features for performing action recognition tasks. (3) ST-MANet intergrades 3D ResNet-50 and the spatial-based deep aggregation network to extract features for performing action recognition tasks. (4) ST-MANet intergrades 3D ResNet-50, the temporal-based deep aggregation network and the spatial-based deep aggregation network to extract features for performing action recognition tasks. (5) ST-MANet intergrades 3D ResNet-50 and the multi-level fusion strategy to extract features for performing action recognition tasks.

As shown in Table 4, there are six conclusions: (a) The Effectiveness of 3D ResNet-50 as a Basic Feature Extractor: Through experiment (1), we found that using 3D ResNet-50 alone for action recognition tasks, the model can effectively extract spatiotemporal features. This indicates that 3D ResNet-50, as a deep learning base network, has good performance in the field of action recognition. However, its performance may be limited by the rigidity of the network structure, which may not fully capture the complex spatiotemporal information in videos. (b)The Complementary Role of the Temporal-Based

Deep Aggregation Network (TDA): In experiment (2), by integrating 3D ResNet-50 with the Temporal-Based Deep Aggregation Network (TDA), the model's performance in action recognition tasks was improved compared to using 3D ResNet-50 alone. This suggests that TDA can effectively supplement the temporal feature extraction capabilities of 3D ResNet-50, enhancing the model's ability to capture the temporal sequence changes in actions. (c) The Integration Effect of the Spatial-Based Deep Aggregation Network (SDA): In experiment (3), the model's performance was also enhanced by combining 3D ResNet-50 with the Spatial-Based Deep Aggregation Network (SDA). This indicates that SDA effectively strengthens the feature extraction in the spatial dimension, enabling the model to better understand spatial relationships and layouts in videos. (d) Significant Improvement with Spatiotemporal Fusion Strategy: In experiment (4), by fusing both Temporal-Based and Spatial-Based Deep Aggregation Networks, the model's performance in action recognition tasks reached its peak. This result confirms the importance of joint optimization of spatiotemporal information for enhancing the accuracy of action recognition, where spatiotemporal fusion strategy can more comprehensively capture the complex spatiotemporal features in videos. (e) Further Optimization with Multi-Level Fusion Strategy (MF): in experiment (5), by introducing the Multi-Level Fusion Strategy (MF) in conjunction with 3D ResNet-50, the model's performance was further optimized in some cases. This indicates that the Multi-Level Fusion Strategy can more meticulously integrate spatiotemporal information at different levels, providing richer feature representations for action recognition tasks. (f) the combined effect of these modules transcends the sum of their individual contributions, highlighting the synergy and complementarity inherent in ST-MANet's design. This observation underscores the rationality and effectiveness of the modules within ST-MANet, demonstrating their collective power in enhancing the model's accuracy and performance. Such a finding not only validates the architectural choices made in ST-MANet but also offers insights into the importance of holistic system design in achieving superior results in action recognition tasks.

4.4. The effectiveness of multi-level fusion

To delve deeper into the contributions of each stage within ST-MANet to the overall performance of action recognition, we conducted a comparative analysis. This analysis involved directly classifying features extracted from individual stages and then comparing these results with the outcomes obtained from features that have undergone multi-level fusion. The detailed results of this comparison are presented in Table 4.

These are some observations: (1) The top-level features, which are the output of the deepest layers in the network, are found to contain rich semantic information that is crucial for understanding the actions within the videos. These features have undergone a series of transformations and aggregations that enable them to capture complex patterns and relationships present in the data. On the other hand, features from the shallower layers tend to lack the sophistication to adequately model semantic information. They exhibit limited reasoning capabilities when it comes to processing both temporal and spatial information, which is evident from the decreasing trend in recognition accuracy as we move towards shallower features. (2) The features extracted from Stage 5 represent a turning point in this analysis. At this stage, the features have been fused from both the main branch and the temporal aggregation branch. This fusion results in a rich amalgamation of temporal and spatial information, alongside semantic insights. As a result, Stage 5 features achieve the

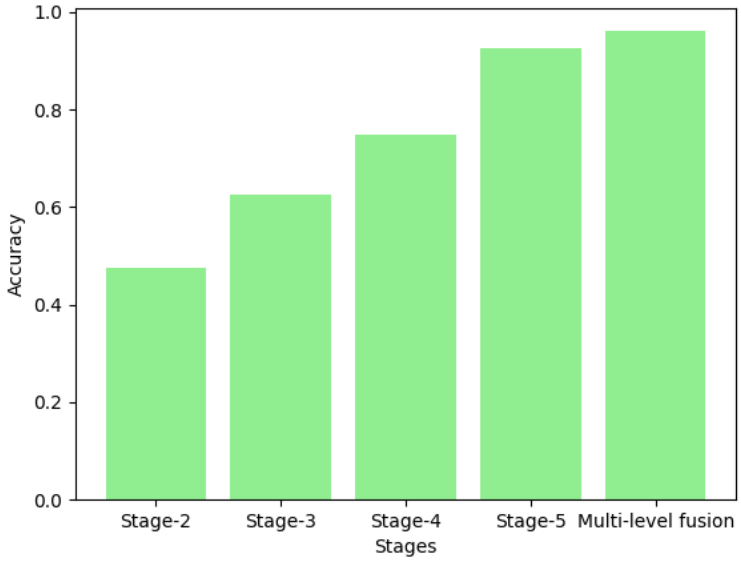
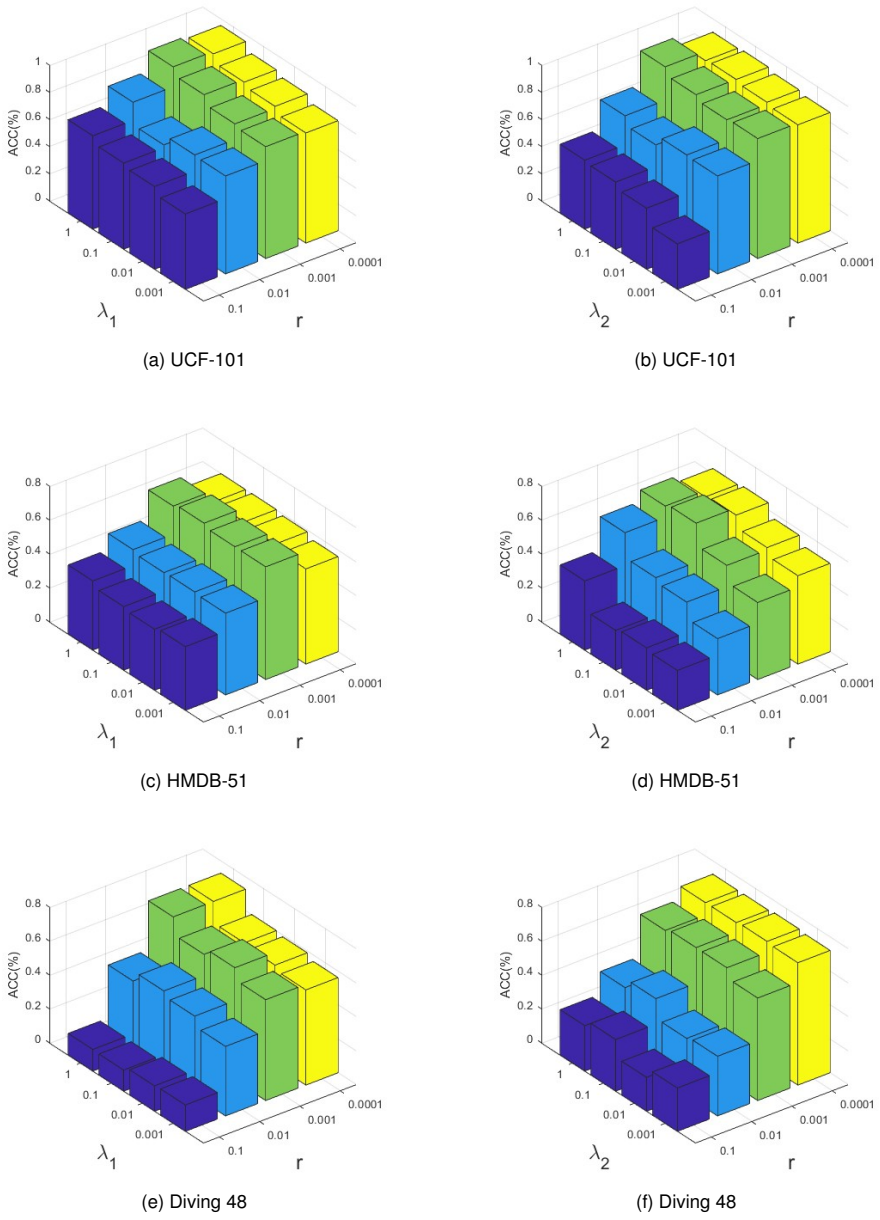


Fig. 4. Contribution ablation of each stage of in ST-MANet

highest accuracy among the single-scale features, indicating the importance of the temporal aggregation module in enhancing the network's temporal modeling capabilities. (3) The multi-level fusion strategy takes this a step further by integrating features from multiple stages of the network. Temporally, this strategy spans a wide range of information, capturing various visual rhythms and exploring multiple time scales. This comprehensive temporal coverage allows the network to recognize actions that may vary in speed and duration. Spatially, the fused features not only include deeply learned, highly discriminative information but also incorporate finer details from the shallower layers. This combination of deep and shallow features ensures that the model can leverage both the nuanced understanding of the scene's layout and the broader context of the action being performed. Consequently, the multi-level fusion strategy stands out as the most effective approach, achieving the highest accuracy in the action recognition task. This strategy exemplifies the power of combining diverse types of information at different levels of abstraction, which is a key principle in designing deep learning models for complex tasks such as video action recognition. By harnessing the collective strength of features from various stages, ST-MANet is able to outperform other approaches and set a new benchmark for accuracy in the field.

4.5. Parameter Analysis

Fig. 5 illustrates the results of the analysis experiments investigating the influence of trade-off parameters λ_1 and λ_2 and learning rate r on three datasets. Specifically, we



(a) UCF-101 (b) UCF-101

(c) HMDB-51 (d) HMDB-51

(e) Diving 48 (f) Diving 48

Fig. 5. The sensitivity analysis of parameters λ_1 and λ_2 and learning rate r on three datasets for ST-MANet

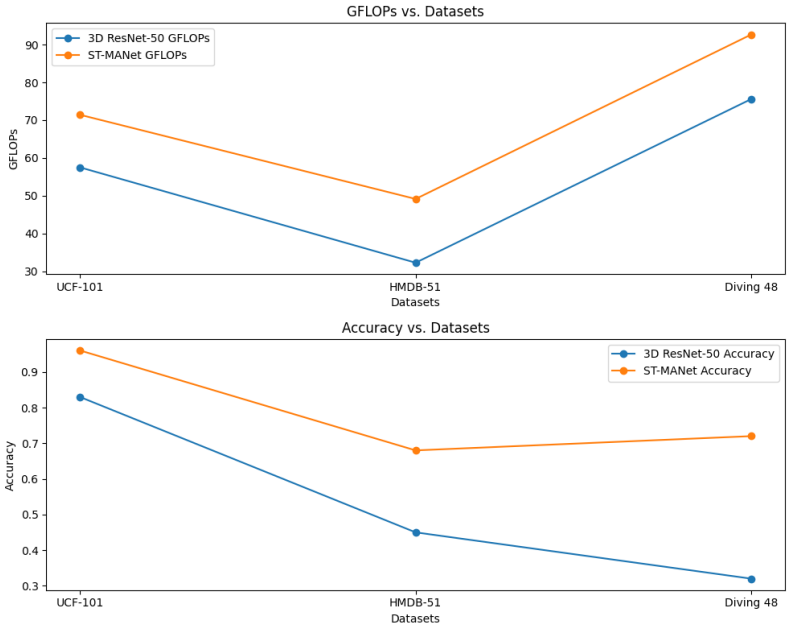


Fig. 6. Comparison of computational complexity and accuracy of ST-MANet and 3D ResNet-50 in terms of three datasets

constrained λ_1 and λ_2 within the set $\{0, 0.0001, 0.001, 0.01, 0.1, 1\}$. Throughout the experiments, one parameter was maintained at a constant value while the other was varied. The results displayed in Fig. 5 demonstrate the robustness of ST-MANet to changes in λ_1 and λ_2. The performance consistently remains satisfactory, particularly when λ_1 is set to 0.01 and λ_2 is set to 0.001. As a result, for the three datasets, ST-MANet is configured with $\lambda_1 = 0.01$ and $\lambda_2 = 0.001$ in the experiments. This configuration has been empirically determined to yield a high level of accuracy, suggesting that it effectively weights the different components of the network's loss function in a manner that is conducive to learning robust action recognition features. The parameters λ_1 and λ_2 play a crucial role in the network by controlling the balance between certain regularization terms and the overall loss. The optimal values found in this study provide a valuable reference for future research and applications of ST-MANet, as they offer a blueprint for achieving high performance with a reasonable computational cost.

In addition to the qualitative assessment of the ST-MANet's performance, we have conducted a quantitative analysis to evaluate its computational efficiency and effectiveness. This analysis includes a direct comparison of the floating-point operations required by ST-MANet and the 3D-ResNet-50 architecture when applied to the UCF-101 dataset, a benchmark for action recognition research. As illustrated in Table 6, the implementation of ST-MANet results in an approximate 24% increase in floating-point operations when contrasted with the 3D ResNet-50. This increase is attributed to the additional temporal and spatial modules integrated into ST-MANet, which are designed to enhance the net-

work's ability to capture complex spatiotemporal features within videos. Despite this rise in computational demand, it is important to note that the computational footprint of both ST-MANet and 3D ResNet-50 remains relatively comparable. This suggests that while ST-MANet is more computationally intensive, the increase is not prohibitive and is justified by the performance gains. The accuracy improvements achieved by ST-MANet are particularly noteworthy. On the UCF-101 dataset, ST-MANet demonstrates a 12.88% increase in accuracy over the baseline 3D ResNet-50. This significant leap in performance underscores the value of the network's advanced feature extraction capabilities, particularly its ability to model the intricate temporal dynamics that are critical for accurate action recognition. In conclusion, while ST-MANet does require more computational resources than 3D ResNet-50, the increase is moderate and does not lead to a significant disparity in computational requirements. The substantial improvement in accuracy, especially on challenging datasets like UCF-101, is a testament to the network's advanced feature aggregation and modeling capabilities. This makes ST-MANet a compelling choice for applications where high accuracy in action recognition is paramount, justifying the additional computational investment.

5. Conclusion

This paper introduces ST-MANet, a novel spatio-temporal-based multi-level aggregation network for action recognition.Specifically, by leveraging correlations between different spatial and temporal positions on the feature map, ST-MANet effectively explores long-range spatial and temporal dependencies, generating spatial and temporal attention maps that assign varying weights to features at different locations. Moreover, the integration of a multi-scale approach enables the modeling of various visual rhythms and the capture of multi-scale spatiotemporal information. The proposed spatial diversity constraint further enhances the network's ability to focus on distinct spatial areas at different scales, thereby incorporating more diverse spatial information into multi-scale features. Experimental results demonstrate the effectiveness of ST-MANet compared to existing approaches, showcasing high accuracy across three datasets. In the future, there are several promising avenues for future research. Firstly, exploring more sophisticated attention mechanisms could further improve the network's ability to capture salient spatial and temporal features. Additionally, investigating ways to incorporate domain knowledge or context information may enhance the model's robustness in real-world scenarios. Furthermore, extending the application of ST-MANet to other related tasks beyond action recognition, such as anomaly detection or event localization, could offer valuable insights into its broader applicability and effectiveness. Overall, ST-MANet presents a strong foundation for advancing the state-of-the-art in spatio-temporal-based action recognition and opens up exciting possibilities for future research in the field.

References

1. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6299–6308 (2017)

2. Chai, W., Jiang, Z., Hwang, J.N., Wang, G.: Global adaptation meets local generalization: Unsupervised domain adaptation for 3d human pose estimation. In: ICCV. pp. 14655–14665 (2023)
3. Chen, J., Ho, C.M.: Mm-vit: Multi-modal video transformer for compressed video action recognition. In: Proceedings of the IEEE/CVF winter conference on applications of computer vision. pp. 1910–1921 (2022)
4. Feichtenhofer, C.: X3d: Expanding architectures for efficient video recognition. In: computer vision and pattern recognition. pp. 203–213 (2020)
5. Feichtenhofer, C., Fan, H., Malik, J., He, K.: Slowfast networks for video recognition. In: international conference on computer vision. pp. 6202–6211 (2019)
6. Gao, J., Liu, M., Li, P., Zhang, J., Chen, Z.: Deep multiview adaptive clustering with semantic invariance. IEEE Transactions on Neural Networks and Learning Systems (2023)
7. Guan, X., Yang, Y., Li, J., Xu, X., Shen, H.T.: Mind the remainder: taylor's theorem view on recurrent neural networks. IEEE Transactions on Neural Networks and Learning Systems 33(4), 1507–1519 (2021)
8. Guo, C., Fan, B., Zhang, Q., Xiang, S., Pan, C.: Augfpn: Improving multi-scale feature learning for object detection. In: computer vision and pattern recognition. pp. 12595–12604 (2020)
9. Gupta, P., Thatipelli, A., Aggarwal, A., Maheshwari, S., Trivedi, N., Das, S., Sarvadevabhatla, R.K.: Quo vadis, skeleton action recognition? International Journal of Computer Vision 129(7), 2097–2112 (2021)
10. He, Z., Lv, J., Fang, S.: Representation modeling learning with multi-domain decoupling for unsupervised skeleton-based action recognition. Neurocomputing p. 127495 (2024)
11. Huo, J., Cai, H., Meng, Q.: Independent dual graph attention convolutional network for skeleton-based action recognition. Neurocomputing p. 127496 (2024)
12. Jiang, B., Wang, M., Gan, W., Wu, W., Yan, J.: Stm: Spatiotemporal and motion encoding for action recognition. In: the IEEE/CVF international conference on computer vision. pp. 2000–2009 (2019)
13. Karim, M., Khalid, S., Aleryani, A., Khan, J., Ullah, I., Ali, Z.: Human action recognition systems: A review of the trends and state-of-the-art. IEEE Access (2024)
14. Karpathy, A., Joulin, A., Fei-Fei, L.F.: Deep fragment embeddings for bidirectional image sentence mapping. Advances in neural information processing systems 27 (2014)
15. Li, P., Gao, J., Zhang, J., Jin, S., Chen, Z.: Deep reinforcement clustering. IEEE Transactions on Multimedia (2022)
16. Li, Y., Ji, B., Shi, X., Zhang, J., Kang, B., Wang, L.: Tea: Temporal excitation and aggregation for action recognition. In: computer vision and pattern recognition. pp. 909–918 (2020)
17. Lin, J., Gan, C., Han, S.: Tsm: Temporal shift module for efficient video understanding. In: international conference on computer vision. pp. 7083–7093 (2019)
18. Liu, Z., Luo, D., Wang, Y., Wang, L., Tai, Y., Wang, C., Li, J., Huang, F., Lu, T.: Teinet: Towards an efficient architecture for video recognition. In: Proceedings of the AAAI conference on artificial intelligence. vol. 34, pp. 11669–11676 (2020)
19. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. Advances in neural information processing systems 27 (2014)
20. Sun, L., Jia, K., Chen, K., Yeung, D.Y., Shi, B.E., Savarese, S.: Lattice long short-term memory for human action recognition. In: IEEE international conference on computer vision. pp. 2147–2156 (2017)
21. Tao, L., Wang, X., Yamasaki, T.: Rethinking motion representation: Residual frames with 3d convnets. IEEE Transactions on Image Processing 30, 9231–9244 (2021)
22. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3d convolutional networks. In: IEEE international conference on computer vision. pp. 4489–4497 (2015)

23. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3d convolutional networks. In: IEEE international conference on computer vision. pp. 4489–4497 (2015)
24. Varol, G., Laptev, I., Schmid, C.: Long-term temporal convolutions for action recognition. IEEE transactions on pattern analysis and machine intelligence 40(6), 1510–1517 (2017)
25. Wang, L., Tong, Z., Ji, B., Wu, G.: Tdn: Temporal difference networks for efficient action recognition. In: computer vision and pattern recognition. pp. 1895–1904 (2021)
26. Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., Van Gool, L.: Temporal segment networks: Towards good practices for deep action recognition. In: European conference on computer vision. pp. 20–36 (2016)
27. Wang, Y., Long, M., Wang, J., Yu, P.S.: Spatiotemporal pyramid network for video action recognition. In: Computer Vision and Pattern Recognition. pp. 1529–1538 (2017)
28. Yang, C., Xu, Y., Shi, J., Dai, B., Zhou, B.: Temporal pyramid network for action recognition. In: computer vision and pattern recognition. pp. 591–600 (2020)
29. Yang, H., Yuan, C., Li, B., Du, Y., Xing, J., Hu, W., Maybank, S.J.: Asymmetric 3d convolutional neural networks for action recognition. Pattern recognition 85, 1–12 (2019)
30. Zhang, D., Dai, X., Wang, Y.F.: Dynamic temporal pyramid network: A closer look at multi-scale modeling for activity detection. In: Computer Vision. pp. 712–728 (2019)

Yuhang Wang is with the School of Physical Education, Harbin University. Research direction: image processing, sports analysis, education data analysis, artificial intelligence.

Received: April 18, 2024; Accepted: June 29, 2024.

Computer Science and Information Systems 21(4):1845–1864 https://doi.org/10.2298/CSIS240418059W

Multi-object Real-time Tracking for Intelligent Breeding of Animal

Fei Wang[1], Bin Xia[2,⋆], and Liwu Pan[1]

[1] School of Information Engineering, Henan University of Animal Husbandry Economy
Zhengzhou 450044 China
feinix@126.com
[2] Zhengzhou Cotton&Jute Engineering Technology and Design Research Institute, All China Federation of Supply and Marketing Cooperatives
Zhengzhou 450044 China
xiabinwto@163.com

Abstract. Animal intelligent breeding utilizes advanced technology and intelligent systems to monitor, analyze, and optimize animal growth environments and management, which can enhance breeding efficiency and animal health levels. In this paper, we propose a new multi-object real-time tracking within deep framework for intelligent breeding of animal (MRT-IB), which consists of semantic feature extraction module, center point prediction module, and object and trajectory calibration module. MRT-IB reduces the difficulty of modeling animal trajectories by performing animal detection on consecutive frames, resulting in higher robustness in real farming scenarios compared to traditional multi-object tracking schemes that directly model animal motion trajectories.

Keywords: animal intelligent breeding, multi-object real-time tracking, object and trajectory calibration.

1. Introduction

Intelligent breeding is an innovative approach that integrates advanced technology with animal husbandry practices [4,21,19]. It improves farming efficiency and animal welfare by continuously and automatically monitoring animal behavior and taking timely and appropriate measures when anomalies are detected. Intelligent breeding technology is supported by advanced artificial intelligence algorithms and hardware systems, including various sensors such as microphones, cameras, accelerometers, and global positioning systems. These sensors continuously collect biological response data from animals, such as sound, behavior, movement, and location. Then, by analyzing these data and extracting features relevant to target behaviors, machine learning models are trained to achieve real-time detection, tracking, and monitoring of abnormal animal behavior [17,22,8]. Once the system detects abnormal behavior or health issues, it immediately issues alerts and takes corresponding actions, such as alerting veterinarians or automatically adjusting the animals' environmental conditions. This real-time monitoring and intervention help farm managers promptly identify and address issues, thereby improving animal welfare and production efficiency.

⋆ Corresponding Author

In recent years, with the rapid development and widespread application of deep neural networks, numerous real-time animal monitoring algorithms based on multi-object tracking have been proposed for smart farming. Ahrendt et al. designed a tracking system for free-range animals, using partial animal images to establish a 5D Gaussian model for individual animal targets. This system can track at least 3 animals over long time spans [1]. Martin et. al proposed a multi-object tracking method based on joint kernel correlation filtering. They designed a joint filter based on a pure target appearance model of kernel correlation filtering, effectively addressing the problem of poor algorithm performance in cases of severe target occlusion and complex environmental backgrounds [25]. Matthews et al. proposed a multi-object tracking algorithm using the Hungarian algorithm to match detection results from adjacent frames. The algorithm performed well in detecting individual targets and tracking consecutive frames in 3D video sequences of group-raised pigs, achieving an overall tracking accuracy of 89% for pig targets [18]. Van der Zande proposed a multi-object tracking algorithm based on optical flow and interframe interpolation for tracking dairy cows during rumination, achieving a final accuracy of 89.12% [27]. Philipp et al. proposed a multi-object tracking algorithm based on multi-feature detection and target association. They used color and texture features for adaptive detection of animals, edge features for image segmentation, and the Kuhn-Munkres algorithm to determine the association between animal targets and their mask image regions. Experimental results showed that the algorithm has strong robustness and meets real-time tracking requirements for multi-object tracking [5]. Zhang et al. combined a CNN-based object detection model with a deep correlation filtering tracking algorithm to track all pig targets' trajectory coordinates in structured pigsty environments. Test results showed that the tracking accuracy reached 94.72% [28]. Ali et al. proposed a multi-object tracking algorithm based on object detection, locating the positions of detected targets in single-image video frames, and then connecting the trajectories of detected targets in consecutive frames. Experimental results showed that the best tracking accuracy of 92.68% was achieved with a synchronous search for subsequent frames of 9 frames and a tracking duration of 30 frames [2].

In current research combining computer vision technology with the animal husbandry, researchers mostly focus on introducing computer vision technology to address specific problems within the farming process. This approach often yields relatively good results. However, experiments in these studies often require artificially imposed constraints. For instance, in the detection of hoof problems in cattle, researchers limit the number and poses of cattle in a video segment. They then detect, record, and calculate the frequency of each cow's hoof strikes to infer whether the cow is lame based on empirical data. While this method can achieve high detection accuracy, the manual recording of identities for each detected cow increases the burden on farming personnel. Many studies also attempt to use detection networks to simultaneously detect the behavior of multiple animals in images, aiming to improve farming accuracy by monitoring the daily status of livestock. These efforts often yield highly accurate behavior detection results. However, since these methods cannot distinguish between different animals within a video segment, they can only provide the number of animals exhibiting a particular behavior at a given time, without capturing the specific behavioral changes of individual animals over time. Therefore, further research is needed in current computer vision technology applied in the farming

industry to continuously identify and differentiate between different animals over a period of time, in order to obtain more precise records of animal status information.

To this end, we propose a new multi-object real-time tracking based on feature consistency for intelligent breeding of animal (MRT-IB), which comprises three parts, i.e., semantic feature extraction module, center point prediction module, and object and trajectory calibration module. Specifically, In the tracking task, MRT-IB considers the similarity of theposition and appearance of the same animal between consecutive frames.and the differences from other animals, and proposes a multi-objecttracking scheme that uses animal detection results and animal position and appearance semantic features as similarity criteria. Based on this, acenter point prediction module is designed to use the continuity of animalposition in the video to assist in detection and tracking. This schemereduces the difficulty of modeling animal trajectories by performinganimal detection on consecutive frames,and is more robust thantraditional multi-object tracking schemes that directly model animaltrajectories in real farm scenes. At the same time, experimental results onmultiple farm scene datasets demonstrate that the center point predictionmodule can effectively improve the detection and tracking accuracy ofmany similar animals appearing in the same scene. Finally, a plethora of experiments validate the effectiveness of MRT-IB in the intelligent breeding of animal.

Section 2 reviews existing smart farming methodologies, particularly focusing on computer vision techniques for livestock monitoring. Section 3 introduces MAR-IB (Multi-Animal Recognition with Individual Behavior) and its methodology for addressing individual animal identification challenges in video streams. Section 4 presents experimental results evaluating MAR-IB's performance in real-world farming scenarios. Finally, Section 5 draws conclusions on MAR-IB's effectiveness and implications for livestock management in smart farming.

2. Related Works

In this section, we will introduce researches on computer vision technology and livestock intelligent breeding.

2.1. Computer vision technology

Computer vision technology is an extremely important branch of artificial intelligence, and it can now be divided into three most basic tasks: Image Classification, Object Detection, and Image Segmentation. In response to different task scenarios and requirements, research in computer vision has continuously refined, leading to the emergence of tasks such as Multiple Object Tracking (MOT) and Action Recognition [11].

For image classification algorithms, traditional methods such as KNN, SVM, and the Mixture of Gaussians model dominated the field before 2012, offering better performance in classification tasks. However, after the neural network-based AlexNet achieved a performance far exceeding traditional algorithms in the ImageNet LSVRC-2012, neural networks became the mainstream in the field of image classification [15]. Subsequent models like VGGNet, GoogLeNet, ResNet, and EfficientNet have continuously pushed the upper limits of classification task performance [21]. Recently, classifiers based on Transformers have reached the state of the art on major public datasets, becoming a new research

hotspot. Regarding object detection algorithms, the current mainstream algorithms are all based on deep learning. Object detection algorithms can be divided into two-stage object detection algorithms and one-stage object detection algorithms based on the algorithm's phase. Two-stage object detection algorithms require an additional Region Proposal Network (RPN) module; the most classic algorithms in this category are the R-CNN series, such as Fast R-CNN and Mask R-CNN [19]. One-stage object detection algorithms do not have an extra region proposal module and are trained end-to-end. A highly classic algorithm in this category is the YOLO series, including YOLOv3 and YOLOv4. Other algorithms like SSD, CenterNet, M2Det, and EfficientDet also have their innovative aspects [22]. Image segmentation algorithms can be divided into two major categories: Semantic Segmentation and Instance Segmentation. Semantic segmentation only requires the segmentation of objects of different types without distinguishing between different individuals of the same category. Classic algorithms include FCN, U-Net, PSPNet, DeepLab v3 Plus [8]. Instance segmentation, on the other hand, needs to differentiate between different individuals of the same category. Currently, the most popular algorithms are Mask R-CNN, SOLO v2 [23].

Multi-Object Tracking (MOT) algorithms can be divided into two subtasks: object detection and data association. Object detection algorithms are used to identify the objects that need to be tracked; data association algorithms assign identity identifiers (IDs) to the detected objects, ensuring that the same object is assigned the same ID across different image frames. These two tasks can be completed by a single network or by two independent networks in cascade. Based on the way these two subtasks are completed, the current mainstream deep learning-based multi-object tracking algorithms can be divided into two categories: single-step MOT and two-step MOT. Single-step MOT uses a single network to complete the detection and tracking tasks of objects, usually outputting information such as category, bounding box, ID, etc., through a detection head, for example, FairMOT, which has achieved high performance on multiple datasets [24]. Two-step MOT separates object detection and data association, so the actual focus of the work is on the data association task, such as Sort, DeepSort, and MOTDT, graph neural network algorithms [9]. The final implementation of the algorithm can be completed through the cascade of a detector and an associator, for example, the classic YOLOv4+Deepsort algorithm. Some algorithms delegate most of the work to the detector, such as Trackot++ proposed by Philipp Bergmann et al., pushing the limit of detection-based tracking [5]. Action recognition algorithms, in a narrow sense, only require the classification of individual videos or sequential images, such as the classic Two-Stream algorithm. However, in a broader sense, action recognition algorithms can be further divided into temporal action detection algorithms and spatiotemporal action detection algorithms. These corresponding tasks are more challenging, as they require the analysis and processing of long videos and can actually be considered a part of video content understanding. Among the most classic algorithms in this field are R-C3D, BMN, and TSP [3].

2.2. Livestock intelligent breeding

Current research combining livestock farming with computer vision often focuses on specific scenarios, where researchers apply various deep neural networks to the study of animal husbandry for multiple purposes within the breeding process. Object detection networks are primarily used by researchers for the detection of specific animals, such as

pigs, cattle, and sheep. There are also studies that apply detection networks to the identification of animal abnormalities. For example, Kang et al. developed a dairy cow limp scoring system, using a deep learning-based object detection network for the localization of cow hooves in videos, with an average precision of 87.0% [13]. The study then used the detected positions of cow hooves from consecutive frames as input, calculating the difference in lift-off and ground contact times of a specific cow's hooves to determine if the cow was limping. Cermek et al. used YOLOv2 on RGB images to detect the occurrence of digital dermatitis in cattle through digital image detection, achieving an accuracy rate of 88% [6]. Yang et al. employed Faster R-CNN and extracted the occupation index (the ratio of the head to the feeding area) as a spatial feature to identify the feeding behavior of multiple pigs [26]. Chen et al. used Inception and LSTM to extract spatiotemporal features for identifying pig feeding behavior at the group level [7]. This study proposed an image processing algorithm based on maximum entropy segmentation, HSV (hue, saturation, and value) color space transformation, and template matching, calculating the roundness of the head, the ratio of the head to the feeding area, the cumulative pixel count of head movement, and the distance from the head to the numbered pig's back to determine the identity and feeding time of each pig. These studies demonstrate the great potential of computer vision systems in identifying dairy cows with digital dermatitis, reducing the incidence of skin disease, and improving animal welfare.In addition to using deep neural algorithms to directly achieve research objectives, some studies combine deep neural networks with other algorithms in an attempt to achieve better results. For example, Lee et al. proposed the use of deep learning methods to remove backgrounds or detect objects in order to eliminate unnecessary noise and thus enable better post-processing analysisn [16]. They first processed video frames using some image enhancement methods, such as treating moving frames in the video with a Gaussian mixture model. Subsequently, they utilized TinyYOLOv3 to detect individual pigs in each selected frame, and then segmented the detected pigs. By using automatic image thresholding to complete the size calculation of the pigs, they achieved better results than simply using a segmentation detection model like Mask R-CNN [12]. These methods, which do not employ end-to-end deep learning approaches, often have better detection accuracy than their corresponding end-to-end deep learning algorithms by proposing hybrid models like the one mentioned above. Regarding the application of tracking technology in the breeding industry, since traditional computer vision tracking technology mainly relies on trajectory fitting to track detected objects, in animal husbandry, researchers primarily complete tracking tasks by directly contacting the breeding animals. For instance, distinguishing different animals by applying pigmented markings on their bodies, tracking them with the human eye, and then recording the status of animals with different markings at different time points [14]. Alternatively, animals are fitted with electronic devices such as ear tags based on RFID technology, and signals are received by a receiver to obtain information corresponding to different animals. With the increasing concern for animal welfare in recent years, and considering that methods like ear tags, which involve direct contact with animals, can cause injury during the contact process or be damaged during the animal's movement, researchers are also looking for ways to complete tracking tasks without contacting the animals. Su et al. applied a single-object tracking network based on the Siamese mechanism to track individual goats in surveillance videos, successfully tracking a specific goat even when multiple goats appeared in the same video segment [20]. Gan et al. applied the

Deepsort, a multi-object tracking framework based on detection, to the task of tracking multiple pigs in a pen. By leveraging the prior knowledge that the number of pigs in a pen is constant, they introduced a trajectory state correction mechanism to build a real-time pig tracking system [10]. Although deep learning-based tracking frameworks emerged relatively late compared to object detection frameworks, and the difficulty of data annotation is high, there is limited work combining livestock farming with single-object and multi-object tracking algorithms. However, the existing work indicates that tracking tasks have significant application scenarios in animal husbandry. They can effectively handle real-time tracking tasks for specific individuals or groups of animals, improving the efficiency of breeders and reducing their workload.

The widespread application of computer vision research in the field of animal husbandry primarily focuses on utilizing anchor-based detection architectures to identify animals in specific scenes within a single frame of an image. This approach requires researchers to manually adjust anchor parameters according to different scenes to accommodate changes in the environment and does not efficiently analyze detection results over time down to the level of individual animals. Additionally, detection networks tend to focus excessively on extracting features related to the current frame, neglecting the continuous information present in video frames, which can be used to assist in detection and tracking. These issues are also the focus of the research in this paper. The reliance on anchor-based detection architectures, while effective in many cases, can be limiting when it comes to adapting to dynamic environments and varying animal behaviors. The need for manual tuning of anchor parameters can be labor-intensive and may not always result in optimal detection performance, especially when there are significant changes in the scene or when the lighting conditions vary. Moreover, the focus on per-frame feature extraction can lead to the loss of valuable temporal information that is inherent in video data. This continuous information can provide context that helps in understanding the behavior and movement patterns of animals over time, which is crucial for accurate tracking and identification of individual animals. To address these challenges, the research presented in this paper aims to explore alternative approaches that can better leverage the temporal dimension of video data, potentially improving the accuracy and efficiency of animal detection and tracking in various farming scenarios. By incorporating methods that can capture and utilize the temporal coherence between frames, the goal is to develop a more robust and automated system for animal monitoring in the context of livestock farming.

3. Multi-object Real-time Tracking for Intelligent Breeding of Animal

In this section, a new multi-object real-time tracking based on feature consistency is proposed for intelligent breeding of animal (MRT-IB), which comprises three parts, i.e., semantic feature extraction module, center point prediction module, and object and trajectory calibration module. The main mathematical notations used in the paper are listed in Table 1.

3.1. Semantic feature extraction module

DLA-34 network is composed of multiple modules, each of which includes a series of convolutional layers and residual connections. It is capable of fusing feature maps from

Table 1. Frequently used notations

Notations	Description
$f(\cdot)$	the feature extractor
v^t	the t-th video frame in the breeding farm
z^t	the deep semantic representation of v^t
H^t	the heatmap spatiotemporal feature
$\bar{H}^t$	the predicted spatiotemporal feature
P^t	the predicted center heatmap P^t
$\bar{z}^t$	the fusion representations $\bar{z}^t$
a_i	an animal in the t-th video frame
τ_i	the trajectory of the animal a_i
e_i	feature vector of the animal a_i
d_{ij}	the similarity between the i-th animal in the current frame and the j-th active trajectory in the current system.
L_k	the detection box loss
α,β	the balance coefficients

shallow and deep layers, allowing the network to simultaneously learn high-level semantic information and low-level detail information. This aggregation of multi-scale features helps the network to better understand the content of images, especially when dealing with complex structures and diverse textures in the environment of breeding farms.

Therefore, MRT-IB utilizes the DLA-34, which has been pre-trained on the COCO detection dataset, as the feature extractor $f(\cdot)$ to learn deep semantic representations of video frames. Specifically, given the t-th video frame v^t in the breeding farm, the deep semantic representation z^t is obtained via:

$$z^t = f(v^t) \tag{1}$$

3.2. Center point prediction module

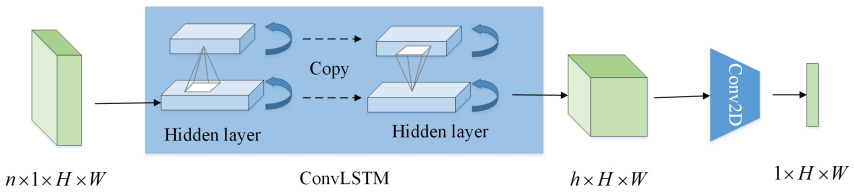


Fig. 1. The structure of center point prediction

The appearance and position of each object in surveillance videos exhibit a certain degree of continuity, which can be leveraged to assist detection in the current frame. Many related works in video detection and multi-object tracking have attempted to model the similarity of object appearances across consecutive frames using Siamese networks or attention mechanisms to enhance the network's detection capabilities in the current frame. This paper optimizes the complex process of feature similarity learning into directly predicting the positions of each object in the current frame. Additionally, an anchor-free

architecture can utilize the center layer to directly detect the position of the object. Therefore, this paper designs the center point prediction module to generate the center points of the current frame's center layer based on the output of the previous frame, as shown in the fig. 1.

Specifically, the center point prediction module first needs to dimensionally expand the centers of the previous n frames. Then, by concatenating along the expanded dimensions, a heatmap spatiotemporal tensor H^t with dimensions of $n \times 1 \times H \times W$ is obtained, which contains information about the changing positions of various animals in the video over time. In surveillance videos, if the animals are far apart from each other, the movement trajectories of each animal can be considered as independent trajectories. When animals are in close proximity, due to the constraints of their volume, the trajectories of the animals may have spatial relative relationships that could potentially influence each other. To capture these spatial relative relationships, this study considers using convolutional kernels for acquisition. Therefore, the center point prediction module selects ConvLSTM to process the animal heatmap spatiotemporal tensor, which can extract spatial features and perform temporal feature processing through h hidden layers, thus addressing the length and width of the current heatmap. ConvLSTM first performs h spatiotemporal convolution operations to extract the spatiotemporal hidden layer features of the movement trajectories of each animal within the spatiotemporal tensor. After obtaining the spatiotemporal hidden layer features, it reverses the output of each hidden layer to obtain a predicted spatiotemporal feature $\bar{H}^t$. Subsequently, the center point prediction module utilizes a convolutional layer to decode this predicted spatiotemporal feature into the predicted center heatmap P^t with dimensions of $1 \times H \times W$ for the current frame.

$$\bar{H}^t = ConvLSTM(H^t) \tag{2}$$

$$P^t = Conv(\bar{H}^t) \tag{3}$$

Then, the Hadamard product is utilized to aggregate the predicted center heatmap P^t and deep semantic representations for obtaining fusion representations $\bar{z}^t$.

$$\bar{z}^t = z^t \circ P^t \tag{4}$$

Such a method can strengthen the features in the semantic representation that are related to the object's position, while suppressing background features unrelated to the object, thereby enhancing the performance of animal detection. Finally, we utilize the detection network to generate detection boxes and appearance feature vectors for each animal in the current frame, which are used for subsequent multi-object trajectory matching.

3.3. Object and trajectory calibration module

For an animal a_i in the t-th video frame, MRT-IB designates the position of the animal a_i as the center coordinates of the detection box [x_i, y_i]. The appearance of the animal a_i consists of two parts: first, the size of the animal, represented by the width and height of the detection box [w_i, h_i]. The second part is the appearance feature vector e_i. In MRT-IB, the detection box and the feature vector are used as information to distinguish between different animals. The trajectory quintuple τ_i for a_i is defined as follows:

$$\tau_i = [x_i, y_i, w_i, h_i, e_i] \tag{5}$$

where each value in the trajectory τ_i represents the detection box information and feature vector of the matched object for animal a_i in the $t-1$ frame.

Due to the inherent continuity in the movement trajectories and size changes of animals, MRT-IB considers using the Kalman filter algorithm to obtain the best estimate of the animal's detection box when matching with the trajectory. Taking the horizontal coordinate x_i^t of animal a_i at time step t as an example, let the best estimate of the horizontal coordinate after Kalman filtering for animal a_i and trajectory j be denoted as x_{ij}. In consecutive frames, an animal's movement trajectory can be considered as uniform linear motion. Generally, the greater the speed, the greater the distance of position change should be, and vice versa. If there is a certain contradiction between these two values, it indicates that the current observation may be affected by some noise. The discrepancy can be corrected using the relationship between position and velocity information to obtain the best estimated mean x_{ij}. The algorithm can first obtain the predicted value s^t of the animal in its current state, where t represents the current state, $\bar{x}_{ij}$ represents the predicted position, and $\bar{v}_{ij}$ represents the predicted velocity.

$$s^t = \begin{bmatrix} \bar{x}_{ij} \\ \bar{v}_{ij} \end{bmatrix} \tag{6}$$

Based on the relationship between velocity and displacement, under the condition that the video frame rate is sufficiently high, it can be approximated that each animal is undergoing uniform motion. Ignoring acceleration, the state transition equation for s^t can be derived as follows:

$$s^t = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} s^{t-1} = F s^{t-1} \tag{7}$$

The state transition matrix is denoted as F, and the relationship between velocity and displacement in any adjacent frames can be defined as mutually constraining R, represented by the covariance of position information x and velocity information v. Since the actual state transition of animals may be affected by external factors such as sudden movements or collisions, leading to certain variations in their state distribution, a noise distribution is introduced with its covariance Q^t represented as shown below.

$$R^t = F^t R^{t-1} (F^{t-1})^T + Q^t \tag{8}$$

$$R^t = \begin{bmatrix} \sum_{xx} & \sum_{xv} \\ \sum_{vx} & \sum_{vv} \end{bmatrix} \tag{9}$$

Simultaneously, based on the detection network, one can obtain the observed values of the current positions of the animals. These observed values are influenced by various factors such as the precision of the detection network itself, the posture of the animals during detection, and the background, and thus often have certain deviations. Let the mean of the observed value distribution be m^t and the variance be U^t. Meanwhile, since it is not possible to directly obtain the animals' velocities from the detection network, the Kalman filter introduces an observation matrix K^t, which allows the algorithm to

derive the predicted observation value $K^t s^t$ from the current state's predicted value s^t. To reconcile the predicted value $K^t s^t$ with the observed value m^t and have their distributions multiplied, resulting in a new distribution, the Kalman gain K is used, as shown below.

$$K = R^t R^t (R^t)^T (R^t R^t (H^t)^T + U^t)^{-1} \tag{10}$$

$$\begin{bmatrix} x_{ij} \\ v_{ij} \end{bmatrix} = s^k + H^t K (m^t - H^t s^t) \tag{11}$$

The final value of $\begin{bmatrix} x_{ij} \\ v_{ij} \end{bmatrix}$ is the best estimate of the match between object a_i and trajectory j after being processed by the Kalman filter. Other parameters can be derived in a similar manner to obtain an optimal estimated quadruple $[x_i, y_i, w_i, h_i]$. The loU (largest overlapping Union) algorithm is then used to calculate the similarity between the current trajectory and the detection box d_{ij}^b.

Furthermore, for any given animal a_i, an appearance feature vector e_i can be extracted. To represent the similarity between it and the most recently matched feature vector along the trajectory, the cosine metric can be employed as follows:

$$d_{ij}^e = \frac{e_i e_j}{\|e_i\| \, \|e_j\|} \tag{12}$$

After obtaining the similarity between animal a_i and trajectory j on the detection box, and the similarity between the appearance feature vectors, we have:

$$d_{ij} = w_b d_{ij}^b + w_e d_{ij}^e \tag{13}$$

where d_{ij} denotes the similarity between the i-th animal in the current frame and the j-th active trajectory in the current system.

3.4. Overall loss function

During the object extraction phase, a re-ID branch is introduced for extracting animal appearance features. After this branch extracts the feature vector of the animal, a fully connected network is used to obtain a confidence vector $P = \{p(c)|c \in [1, N]\}$, where N is the total number of ID labels used to distinguish different animals during training. For the animal a_i, there is a one-hot vector $V^i(c)$ used to indicate its assigned ID label. The cross-entropy loss function for classification constraint is as follows:

$$L_r = -\sum_c \sum_i V^i(c) \log(p(c)) \tag{14}$$

The center point prediction requires supervision during training to enable the module to infer the current frame's predicted heatmap P through the heatmap spatiotemporal tensor I. To this end, MRT-IB employs the structural similarity loss function (SSIM) as a constraint, which quantifies the similarity between two images by comparing their brightness, contrast, and structural information. The label heatmap is denoted as $\hat{P}$, with the brightness of the heatmap represented by its mean μ and the contrast represented by

its variance σ. The structural relationship between the predicted heatmap and the label heatmap is captured by their covariance $\Sigma_{p\hat{p}}$. The SSIM loss function L_p is given by:

$$L_p = \frac{(2\mu_p\mu_{\hat{p}} + c_1)(2\sigma_{p\hat{p}} + c_2)}{(\mu_p^2 + \mu_{\hat{p}}^2 + c_1)(\sigma_p^2 + \sigma_{\hat{p}}^2 + c_2)} \tag{15}$$

where μ_P and $\mu_{\hat{P}}$ are the mean values of the predicted and label heatmaps, respectively, $\Sigma_{P\hat{P}}$ is their covariance, and c_1 and c_2 are constants to prevent instability when the mean or covariance are near zero.

Furthermore, MRT-IB employs a detection box loss to guide the accurate detection of animals:

$$L_k = -\frac{1}{N}\left\{\begin{array}{cc}(1-\hat{p}_{xyc})\log(\hat{p}_{xyc}) & p_{xyc}=1\\ (1-\hat{p}_{xyc})\hat{p}_{xyc}\log(\hat{p}_{xyc}) & otherwise\end{array}\right\} \tag{16}$$

where $\hat{p}_{xyc}$ denotes represents the output of the detection box at position $[x, y]$ for the category c.

According to the aforementioned losses, the overall loss function of MRT-IB is defined:

$$L = L_r + \alpha L_p + \beta L_k \tag{17}$$

where α and β are trade-off parameters.

4. Experiments

4.1. Set up

Dataset: The performance of the MRT-IB in animal detection is verified based on two datasets. The first is the MOT (Multiple-object Tracking Challenge) pedestrian tracking dataset, which is one of the most authoritative datasets in the field of multi-object tracking. It is used to demonstrate the robustness of the algorithm. The second dataset consists of sheep in farm surveillance videos, which is used to explore the effectiveness of the algorithm in research on multi-object tracking of animals in farms. Unlike general tasks such as detection and classification, where the division of training and validation data often follows a 4:1 ratio in favor of the training set, or using five-fold cross-validation, the effectiveness of multi-object tracking models is typically validated by arranging the dataset in sequential order of video frames. The images from the first half of each video sequence are used for training, while the images from the latter half are used to validate the model's performance.

Metric: (1) Multiple Object Tracking Accuracy (MOTA): It is used to evaluate the tracking process over a period of time, accounting for detection errors and track switches. By defining the total number of objects at any time t as G_t, the number of false positives as N_t, the number of false negatives as P_t, and the number of track switches as I_t, the higher the MOTA, the more accurate the onject detection and the more stable the assignment. The MOTA is calculated using the following formula:

$$MOTA = 1 - \frac{\sum_t (N_t + P_t + I_t)}{\sum_t G_t} \tag{18}$$

(2) The IDF1 Score (IDF1): It is used to assess the correctness of track assignment over a period of time. By defining the correctly assigned detections in a period as T, false positives as N, and false negatives as M, the IDF1 is higher if the ID assignment of the detected objects is more accurate during the multi-object tracking process. The calculation formula for IDF1 is given by:

$$IDF1 = \frac{2T}{2T + M + N} \tag{19}$$

(3) ID Switch(IDs): It is used to evaluate the stability of tracks over a period of time, and the formula is as follows:

$$IDs = \sum_{t} I_t \tag{20}$$

Furthermore, Recall and Precision are also used to verify the effectiveness of MAR-IB.

Implementation details: MRT-IB is conducted in the Ubuntu 18.04 operating environment. The model is implemented using Python 3.7 and the PyTorch 3.6 framework, and both training and testing of the model are completed on CUDA 10.1 with an RTX 3080. In the experiment, MRT-IB utilizes the DLA-34 network, which has been pretrained on the COCO detection dataset, as the backbone for feature extraction. The initial learning rate during the training process is set to 10^{-3}, which is reduced to 10^{-4} when the training reaches the 20-th epoch, with a total of 30 epochs for training. Throughout the training process, images in the dataset undergo standard data augmentation procedures such as rotation, flipping, and color space transformations.

4.2. Analysis of Object and Trajectory Matching

In MRT-IB, to verify the effectiveness of the object and trajectory matching, three methods for calculating similarity between the object and trajectory are discussed, as shown in Table 2: (1) the first method directly calculates the IoU distance between the object detection box and the trajectory box recorded in the trajectory, (2) the second method predicts the position of the trajectory in the current frame using the Kalman filtering algorithm, and also filters the object position when matching the object and trajectory. Then, the oU distance is used for similarity calculation. (3) the third method calculates the cosine distance between the feature vector recorded in the trajectory and the feature vector of the object in the current frame for similarity calculation.

There are some observations obtained from the results in Table 2: (1) In the context of surveillance video in a breeding farm, the MRT-IB method can achieve satisfactory tracking results by simply using the Intersection over Union (IoU) distance to calculate the similarity between each object and the detection boxes of the matched objects in the previous frame. This is because in a breeding farm, the movements of the animals are relatively slow for the majority of the time. With a fast detection frame rate, the changes in the detection boxes of the same animal between adjacent frames are not significant, thus yielding fairly good tracking outcomes. However, when sheep flock together or move towards each other, the detection boxes of different sheep are in close proximity, which makes it easier for the IDs of different sheep to be swapped. Therefore, the ID swapping is noticeably higher in the IoU-only scheme compared to other methods. (2) MRT-IB can

associate objects based on the appearance feature information of different animals, thus effectively improving the IDF1 and IDs. However, the degree of improvement is not as significant as that of the Kalman filter model. This indicates that in the context of animal breeding, the motion information of animals plays a more substantial role in enhancing tracking accuracy compared to their appearance information. This is largely due to the fact that the appearance similarity among animals in breeding farms is not as high. Moreover, if an animal is occluded in one posture and then reappears in a different posture, its appearance information actually undergoes significant changes. Considering feature information in such cases could lead to misguidance. On the other hand, by introducing the re-ID branch to utilize appearance information during the tracking process, the detection model's ability to extract object features can be effectively enhanced, thereby improving the detection results of the objects. (3) The combination of the Kalman filter and IoU achieves the best matching because during the association process between each sheep's detection box and the detection boxes saved in the trajectory, it predicts the current detection box parameters for that trajectory based on the changes of past detection boxes. By combining the current frame's detection box of each sheep and filtering it, an optimal estimated value is obtained, effectively improving the accuracy of target-trajectory association. As a result, both IDF1 and IDs are significantly enhanced.

Table 2. Analysis of object and trajectory matching in MRT-IB in terms of the animal dataset

method	MOTA	IDF1	Recall	Precision	IDs
IoU	0.8020	0.8210	0.8770	0.9710	121
IoU+ Feature similarity	0.8240	0.8420	0.8770	0.9710	94
IoU+ Kalman filtering	0.8420	0.8770	0.8770	0.9710	98

4.3. Convergence Analysis of MRT-IB with respect to Animal Dataset

In the performance evaluation of the MRT-IB algorithm applied to an animal dataset, we conducted a thorough convergence analysis. This analysis aimed to explore the performance of the algorithm in key performance indicators such as Precision, Multiple Object Tracking Accuracy (MOTA), and Loss as the number of iterations increased.

From the Fig. 2, it can be observed that the algorithm's Precision rapidly improved in the initial stages, growing from around 0.4 to close to 0.9, demonstrating the algorithm's efficient capability in identifying animal targets. This swift enhancement indicates that the MRT-IB effectively distinguishes targets from the background when processing the animal dataset, thereby reducing false positives and false negatives.

Similarly, the MOTA metric also shows an upward trend, gradually increasing from a lower baseline to above 0.8, reflecting the algorithm's accuracy and stability in tracking multiple animal targets. The improvement in MOTA suggests that the MRT-IB algorithm can effectively maintain the tracking of targets, even under the complex and variable behaviors of animals.

The decreasing trend in the Loss value further confirms the improvement in algorithm performance. In the early stages of iteration, the Loss value significantly reduced, indi-

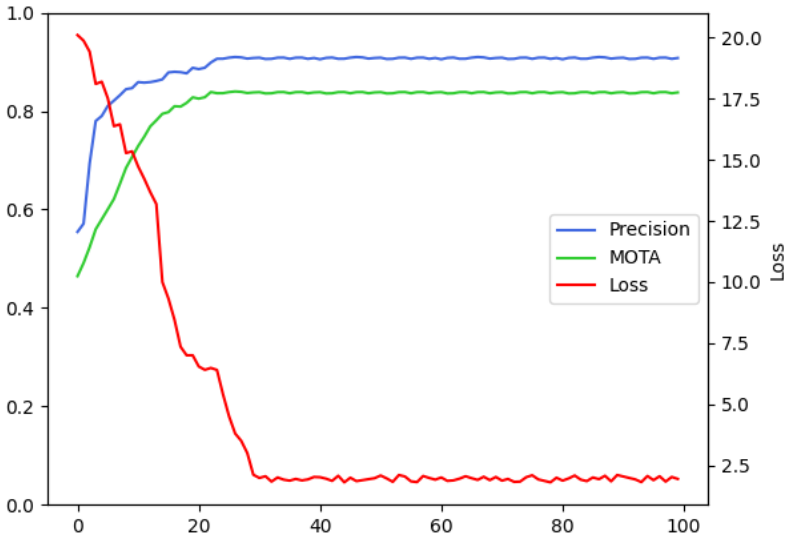


Fig. 2. Convergence Analysis of MRT-IB Performance on Animal Dataset

cating that the gap between the model's predictions and the actual annotations is quickly narrowing. As iterations progress, the Loss value tends to stabilize, suggesting that the model has converged to a more accurate predictive state.

Overall, the convergence analysis of MRT-IB on the animal dataset reveals its potential in multi-object tracking tasks. With an increase in the number of iterations, the algorithm shows significant improvement in key indicators such as Precision, MOTA, and Loss, ultimately reaching a stable high-performance state. These results indicate that MRT-IB is a promising algorithm suitable for animal behavior analysis and other scenarios requiring precise multi-object tracking.

4.4. Effectiveness of Center Point Prediction

The current multi-object tracking model, FairMOT [29], primarily considers the use of location and appearance information extracted by the detection network to associate targets with trajectories across consecutive frames. However, it overlooks the potential of utilizing this information during the detection phase to assist, thereby better aiding detection-based multi-object tracking models in completing tracking tasks. In previous experiments, it was observed that detection-based models could achieve fairly good results by relying solely on the position information, and the accuracy of the detection results could significantly enhance tracking precision. Therefore, this paper considers the introduction of a Central Point Prediction Module into the detection network to explore the impact of incorporating position-based multi-frame information during the detection phase on the results of multi-object tracking. This module can leverage the continuity of individual

animal trajectories in the video and the relative relationships between multiple animal trajectories to predict the positions of each animal in the current frame. By combining these predictions with the semantic features of animals extracted from the current frame image by the feature extraction network, it can assist in animal detection and tracking. The experimental results are shown in Table 3

From the comparative experiments, it can be seen that in MOT-16 and MOT-20, MRT-IB has improved MOTA by 1.0 and 1.3 points, respectively. However, IDS has only shown a slight increase compared to the baseline model. In contrast, in MOT-17, MRT-IB has enhanced MOTA by 1.9 points, while ID has decreased by approximately 25% compared to the baseline model. This indicates that in MOT-16 and MOT-20, although MRT-IB can improve the baseline model's MOTA, it is mainly due to the enhancement of the model's detection capabilities. This is evident from the recall and precision rates of the two datasets, where recall has increased while precision has slightly decreased. Through MRT-IB, the baseline model is better able to detect targets that were previously undetected, but it also introduces some false positives. In the case of MOT-17, MRT-IB not only improves the model's detection capabilities but also enhances its trajectory maintenance capabilities. Unlike the first two datasets, MRT-IB shows a decrease in recall, while the precision increases.

The results above indicate that MRT-IB can leverage the continuity of object positions to enhance the tracking performance of the model in two aspects. On one hand, when an object that has appeared before is missed by the original model in the current frame, MRT-IB will strengthen such targets, thereby reducing the model's false negatives. However, this may also lead to the detection of targets that should have disappeared in the current frame, resulting in false positives. Such false positives can cause the tracking model to assign new IDs to these targets, thus weakening the model's trajectory maintenance capability. On the other hand, when the original model produces false positives in the current frame, these targets, having not appeared before, will be relatively weakened, thereby improving the model's precision. But this could also lead to the weakening of new targets that should have been detected, which in turn reduces the recall rate of the tracking model. Meanwhile, as MRT-IB can improve detection precision, it can effectively enhance the tracking model's ability to maintain trajectories. Across the three datasets, MRT-IB has effectively increased the model's IDF1, indicating that MRT-IB can provide more precise information for the association phase, thereby helping the tracking model to improve the accuracy of matching trajectories to targets during tracking.

Table 3. Enhancement of the Central Point Prediction on the MOT Dataset for the Baseline Model FairMOT

method(dataset)	MOTA	IDF1	Recall	Precision	IDs
FairMOT(MOT-16)	0.6760	0.6940	0.7340	0.9370	456
MRT-IB(MOT-16)	0.6860	0.7050	0.7500	0.9310	479
FairMOT(MOT-17)	0.6930	0.7350	0.7800	0.9110	477
MRT-IB(MOT-17)	0.7140	0.7490	0.7680	0.9410	357
FairMOT(MOT-20)	0.7290	0.7210	0.7640	0.9610	2900
MRT-IB(MOT-20)	0.7424	0.7300	0.7830	0.9550	3014

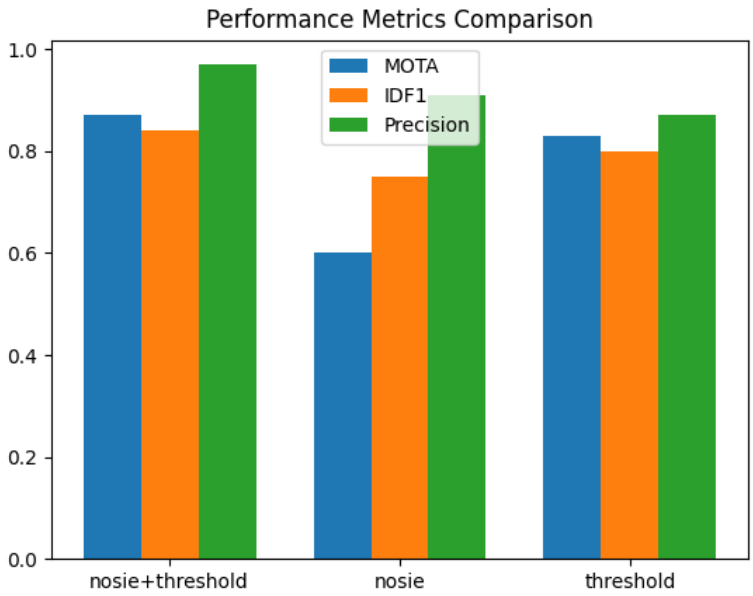


Fig. 3. Influence of noise and threshold on center point prediction

4.5. Influence of noise and threshold on Center Point Prediction

In real tracking scenarios, networks often encounter situations of missed detections and false alarms. However, during training, using real labels directly cannot simulate the missed detections and false alarms that occur in real scenarios. Therefore, it is necessary to process the input heatmaps during training to simulate missed detections and false alarms, thereby improving the robustness of the position prediction network. One intuitive method is to directly train with continuous sequences of images. When training the position prediction network, the heatmaps generated by the model in the previous few frames are used directly as the input to the position prediction network. Although this method allows the position prediction model to use the network's output during training, thereby improving robustness, it introduces fixed temporal continuity during training, which can lead to decreased generalization performance and overfitting. Additionally, in the early stages of training the detection network, the quality of the output of the detection network is not guaranteed, which can lead to the position prediction module learning irreversible noise. Therefore, during the training phase, for any frame image, this study extracts the true keypoint heatmaps of the previous frames and adds noise to these labels to simulate missed detections and false alarms. Specifically, this is achieved by randomly removing annotated points from the heatmap to simulate missed detections, while randomly adding new annotated points near the annotated points to simulate false alarms. This approach allows for the use of random image inputs during training to avoid the model being affected

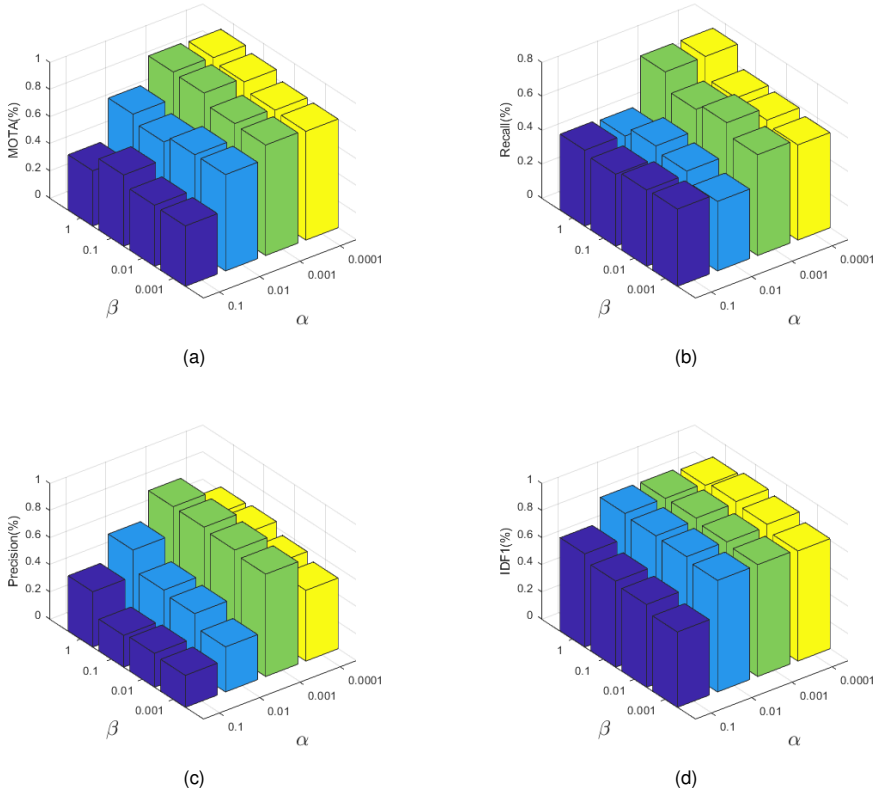


Fig. 4. The sensitivity analysis of parameters α and β on the animal dataset for MAR-IB

by fixed temporal data. During testing, the network produces small responses for many regions without targets. Although these responses do not appear in the results due to not reaching a predefined threshold, they can introduce considerable noise if the unprocessed keypoint heatmaps are directly fed into the central position module.

From the results in Fig 3, it can be observed that without threshold processing, multi-object tracking metrics experience a significant decline, while not adding noise to the labels also leads to a slight decrease in multi-object tracking metrics. To further analyze the experimental results at a deeper level, it is necessary to compare the various detection metrics with the trajectory matching metrics, as shown in Table 4.

Considering the responses in the heatmaps that do not reach a certain threshold during the testing process greatly reduces the accuracy of the tracking network, resulting in numerous false alarms. Consequently, many IDs are erroneously assigned, further weakening the network's ability to maintain trajectories. As a result, all metrics for multi-object tracking experience a significant decrease. However, after adding noise to the training labels, the tracking model effectively improves both recall and precision. This indicates

Table 4. Influence of noise and threshold on center point prediction in multi-object tracking scenario

method	noise+threshold	noise	threshold
Recall	0.7680	0.7660	0.7580
IDs	16.52	15.85	16.26

a notable enhancement in the detection capability of the tracking model, as well as an increase in precision in trajectory assignment and maintenance.

4.6. Parameter Analysis

Fig. 4 presents the outcomes of the sensitivity analysis for the trade-off parameters α and β within the MAR-IB framework when applied to the animal dataset. These parameters were confined to the values within sets $\{0.0001, 0.001, 0.01, 0.1\}$ and $\{0.001, 0.01, 0.1, 1\}$. The graphical representation in Fig. 4 highlights the resilience of MAR-IB to variations in α and β. Notably, the model exhibits consistent and reliable performance across different parameter settings, with an optimal balance achieved when α and β are both set to 0.001 and 1. This suggests that the model is relatively insensitive to fine-tuning of these parameters, which is advantageous for practical applications where computational resources may be limited or where the dataset's complexity necessitates a more straightforward tuning process.

5. Conclusion

In this paper, MRT-IB first highlights the limitations of current object detection algorithms in farm imagery, which are unable to automatically associate detection results of specific animals across consecutive frames for statistical analysis. It then introduces a detection-based multi-object tracking scheme that leverages the continuity of individual animals' movement trajectories and appearance changes in farm imagery, as well as the differences in position and appearance among different animals, to associate detection results. Furthermore, it proposes enhancing the detection capability in multi-object tracking by utilizing animal position information. Finally, this chapter designs multiple control experiments for the multi-object tracking model in farm imagery. The experimental results show that in the process of multi-object tracking of animals in farms, the animals' positions can effectively serve as a reference for data association. The animal appearance information extracted by the re-ID branch can effectively supplement the animal position information, improving the network's feature extraction ability and leading to more accurate detection results. Through ablation experiments on pedestrian datasets and animal tracking datasets, the chapter demonstrates that the model based on center point prediction proposed in this study can further effectively utilize object position information to enhance the model's detection capability and help the tracking model achieve more precise animal ID associations. The performance improvement of this module in multi-object tracking models is evident in multiple scenarios, proving the robustness of the model.

References

1. Ahrendt, P., Gregersen, T., Karstoft, H.: Development of a real-time computer vision system for tracking loose-housed pigs. Computers and Electronics in Agriculture 76(2), 169–174 (2011)
2. Alameer, A., Kyriazakis, I., Bacardit, J.: Automated recognition of postures and drinking behaviour for the detection of compromised health in pigs. Scientific reports 10(1), 13665 (2020)
3. Alwassel, H., Giancola, S., Ghanem, B.: Tsp: Temporally-sensitive pretraining of video encoders for localization tasks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3173–3183 (2021)
4. Bao, J., Xie, Q.: Artificial intelligence in animal farming: A systematic literature review. Journal of Cleaner Production 331, 129956 (2022)
5. Bergmann, P., Meinhardt, T., Leal-Taixe, L.: Tracking without bells and whistles. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 941–951 (2019)
6. Cernek, P., Bollig, N., Anklam, K., Döpfer, D.: Hot topic: Detecting digital dermatitis with computer vision. Journal of dairy science 103(10), 9110–9115 (2020)
7. Chen, C., Zhu, W., Steibel, J., Siegford, J., Wurtz, K., Han, J., Norton, T.: Recognition of aggressive episodes of pigs based on convolutional neural network and long short-term memory. Computers and Electronics in Agriculture 169, 105166 (2020)
8. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Proceedings of the European conference on computer vision (ECCV). pp. 801–818 (2018)
9. Chen, L., Ai, H., Zhuang, Z., Shang, C.: Real-time multiple people tracking with deeply learned candidate selection and person re-identification. In: 2018 IEEE international conference on multimedia and expo (ICME). pp. 1–6 (2018)
10. Gan, H., Ou, M., Zhao, F., Xu, C., Li, S., Chen, C., Xue, Y.: Automated piglet tracking using a single convolutional neural network. Biosystems Engineering 205, 48–63 (2021)
11. Gao, J., Liu, M., Li, P., Zhang, J., Chen, Z.: Deep multiview adaptive clustering with semantic invariance. IEEE Transactions on Neural Networks and Learning Systems (2023)
12. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 2961–2969 (2017)
13. Kang, X., Zhang, X., Liu, G.: Accurate detection of lameness in dairy cattle with computer vision: A new and individualized detection strategy based on the analysis of the supporting phase. Journal of dairy science 103(11), 10628–10638 (2020)
14. Kolarevic, J., Aas-Hansen, Ø., Espmark, Å., Baeverfjord, G., Terjesen, B.F., Damsgård, B.: The use of acoustic acceleration transmitter tags for monitoring of atlantic salmon swimming activity in recirculating aquaculture systems (ras). Aquacultural engineering 72, 30–39 (2016)
15. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Communications of the ACM 60(6), 84–90 (2017)
16. Lee, S., Ahn, H., Seo, J., Chung, Y., Park, D., Pan, S.: Practical monitoring of undergrown pigs for iot-based large-scale smart farm. IEEE Access 7, 173796–173810 (2019)
17. Li, P., Gao, J., Zhang, J., Jin, S., Chen, Z.: Deep reinforcement clustering. IEEE Transactions on Multimedia (2022)
18. Matthews, S.G., Miller, A.L., PlÖtz, T., Kyriazakis, I.: Automated tracking to measure behavioural changes in pigs for health and welfare monitoring. Scientific reports 7(1), 17582 (2017)
19. Ren, S., He, K., Girshick, R., Sun, J., Faster, R.: Towards real-time object detection with region proposal networks, adv. Neural Inf. Process 28 (2015)
20. Su, Q., Tang, J., Zhai, M., He, D.: An intelligent method for dairy goat tracking based on siamese network. Computers and Electronics in Agriculture 193, 106636 (2022)
21. Tan, M., Le, Q.: Efficientnet: Rethinking model scaling for convolutional neural networks. In: International conference on machine learning. pp. 6105–6114 (2019)

22. Tan, M., Pang, R., Le, Q.V.: Efficientdet: Scalable and efficient object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10781–10790 (2020)
23. Wang, X., Zhang, R., Kong, T., Li, L., Shen, C.: Solov2: Dynamic and fast instance segmentation. Advances in Neural information processing systems 33, 17721–17732 (2020)
24. Wang, Z., Zheng, L., Liu, Y., Li, Y., Wang, S.: Towards real-time multi-object tracking. In: European conference on computer vision. pp. 107–122 (2020)
25. Wutke, M., Heinrich, F., Das, P.P., Lange, A., Gentz, M., Traulsen, I., Warns, F.K., Schmitt, A.O., Gültas, M.: Detecting animal contacts—a deep learning-based pig detection and tracking approach for the quantification of social contacts. Sensors 21(22), 7512 (2021)
26. Yang, A., Huang, H., Zhu, X., Yang, X., Chen, P., Li, S., Xue, Y.: Automatic recognition of sow nursing behaviour using deep learning-based segmentation and spatial and temporal features. Biosystems Engineering 175, 133–145 (2018)
27. Van der Zande, L.E., Guzhva, O., Rodenburg, T.B.: Individual detection and tracking of group housed pigs in their home pen using computer vision. Frontiers in animal science 2, 669312 (2021)
28. Zhang, L., Gray, H., Ye, X., Collins, L., Allinson, N.: Automatic individual pig detection and tracking in pig farms. Sensors 19(5), 1188 (2019)
29. Zhang, Y., Wang, C., Wang, X., Zeng, W., Fairmot, W.L.: On the fairness of detection and re-identification in multiple object tracking. DOI: https://doi. org/10.1007/s11263-021-01513-4 pp. 3069–3087 (2021)

Acknowledgments. This work was supported by the: 1. Science and Technology Research Project of Henan Provincial Science and Technology Department in the 2021 year "Real-time cattle status Recognition and disease early Warning Application Research based on Convolutional Representation Flow" (Project number: 212102210138); 2. Annual science and Technology Research project of Henan Provincial Science and Technology Department in the 2023 year "Application Research of Pig status and Behavior Recognition and disease Warning based on space-time and Multi-objectives" (Project number: 232102210082); 3. Annual Science and Technology Research Project of Henan Provincial Science and Technology Department in the 2022 year "Application Research of Vision Transformer in Bovine Body Recognition Based on Adaptive Patch Segmentation" (Project number: 222102210280).

Fei Wang is with the School of Information Engineering, Henan University of Animal Husbandry Economy. Research direction: image processing, artificial intelligence.

Bin Xia is with the Zhengzhou Cotton&Jute Engineering Technology and Design Research Institute, All China Federation of Supply and Marketing Cooperatives. Research direction: big data processing, pattern recognition.

Liwu Pan is with the School of Information Engineering, Henan University of Animal Husbandry Economy. Research direction: image processing, artificial intelligence.

Received: April 18, 2024; Accepted: June 29, 2024.

Computer Science and Information Systems 21(4):1865–1885 https://doi.org/10.2298/CSIS240418061G

MFE-Transformer: Adaptive English Text Named Entity Recognition Method Based on Multi-feature Extraction and Transformer

Liuxin Gao

School of Foreign Languages, Zhengzhou University of Science and Technology
450064 Zhengzhou, China
publicgj@163.com

Abstract. English text named entity recognition aims to alleviate the problem of insufficient labeling data in the target domain. Existing methods usually use feature representation or model parameter sharing to realize cross-domain transfer of entity recognition capability, but there is still a lack of full utilization of structured knowledge in text sequences. Therefore, this paper proposes an adaptive English named text entity recognition method based on multi-feature extraction and transformer. Firstly, a bidirectional long term memory conditional random field entity recognition model based on BERT pre-trained language model is constructed on a generic domain dataset. In the training process, the weights of two character vectors of text words are dynamically calculated and combined, which makes the model make full use of the information in the character granularity, and the parts-of-speech information and block analysis are added as additional features. The word vectors, character-level features and additional features are spliced into the BiLSTM-CRF neural network model for training. Finally, experiments are carried out on five English datasets and specific cross-domain named entity recognition datasets respectively. The results show that the average performance of the proposed model is improved by 0.43% and 1.47% compared with the current cross-domain model, indicating that the structured knowledge in feature representation can effectively improve the entity recognition capability of the target domain.

Keywords: Adaptive English named text entity recognition, multi-feature extraction, transformer, domain invariant knowledge.

1. Introduction

Named entity recognition (NER) is a fundamental information extraction task that plays a crucial role in natural language processing applications such as information retrieval, automatic text summarization, intelligent question answering, machine translation4, and knowledge graphs. NER is to extract some predefined specific entity from the sentence and identify its correct type, such as person, place, organization [1-3].

Early NER methods can be divided into two types: rule-based methods and statistics-based methods. The rule-based approach is to manually design a large number of rules for specific fields to match named entities according to the task, and generalize and restrict them to other fields. Therefore, a rule-based approach is time-consuming and expensive. A statistics-based approach transforms NER tasks into sequential labeling tasks and trains

them using a corpora of manual labeling. Because the cost of labeling is much lower than the cost of designing rules, statiscy-based methods became mainstream before the explosion of deep learning, such as hidden Markov models (HMM) [4] or conditional random fields (CRF) [5].

NER has gone through three major stages of development, rule-based recognition methods, statistics-based machine learning methods and depth-based learning methods. In NER, the appearance of BiLSTM-CRF has opened the prelude of named entity recognition in deep learning. Its emergence makes the model more concise and robust, and becomes the deep learning benchmark for solving NER problems. The representation layer is based on character information, adding radicals, glyphs, parts of speech, pronunciation, dictionaries and other information on the basis of character information. Text encoding can use neural networks to obtain context dependencies. The label decoding layer predicts and labels the input sequence. These models have their own advantages and disadvantages when dealing with entities [6-8]. For example, CNN can parallelize data, so the calculation speed is faster, but there is a problem of memory loss of context information. LSTM is a variant of RNN that is efficient at learning long-haul dependent information, but there are still gradient problems. GNN can mine the relationship between entities more efficiently by constantly mining the graph data model, but the model structure is too large, so the flexibility and expansion is poor. Transformer is often pre-trained in combination with BERT to generate deep language features, but it consumes a lot of computing resources. Therefore, the selection model should be analyzed according to the specific situation.

Character representation is the focus of NER research. Simple external features such as parts of speech, radicals, and strokes, which contain less information, are effective in the absence of enough information, and perform well before the proposed BERT pre-trained model. However, with the development of pre-trained language models such as BERT, pre-trained character representations can capture most of the semantic information of Chinese characters, so it is difficult for NER models to benefit from simple external features when using these pre-trained language models. In addition, it is necessary and valuable to learn from the existing English NER method to solve the Chinese NER problem. Uzair et al. [9] showed that the randomly initialized output layer performed better in the target domain than the output layer in the migrated source domain. The word embedding layer and the hidden layer had better generalization ability, and could be effectively mapped to the semantic vector space of the target domain after migrating to the target domain. Liang et al. [10] used BERT-WWM (whole word mask) model in the Chinese field. The biggest difference between Ernie and the original model was that it adopted the entity-level masking training strategy and combined external knowledge to train the BERT model. The performance of NER experiment conducted in the Chinese field is better than that of the original BERT. Although these models have reached the state of the art (SOTA) in the general field, whether the pre-trained model trained on a large number of unstructured corpus can improve the NER task effect in a specific field still needs to be verified by relevant research [11,12].

To solve these problems, this paper proposes an adaptive English named text entity recognition method based on multi-feature extraction and transformer. The main contributions of this paper are as follows:

1. Firstly, a bidirectional long term memory conditional random field entity recognition model based on BERT pre-trained language model is constructed on a generic domain dataset.
2. In the training process, the weights of two character vectors of text words are dynamically calculated and combined, which makes the model make full use of the information in the character granularity, and the parts-of-speech information and block analysis are added as additional features.
3. The word vectors, character-level features and additional features are spliced into the BiLSTM-CRF neural network model for training.

2. Related Works

2.1. Pre-trained Model

In NER deep learning model, word embedding is a common data preprocessing method that can learn word vector representations and capture the corresponding semantic and syntactic information of sentences.

Pokharel et al. [13] provided a comprehensive review of the pre-trained language model of NLP, and divided PTM into pre-trained word embeddings and pre-trained context encoders. On the basis of classification, pre-trained character embeddings are divided into static embeddings and dynamic contextual embeddings. Static embeddings are trained to look up tables in which the embeddings of each character are fixed, such as NNLM, Word2vec, FastText, Glove, etc. However, because the training result of static embedding is a fixed word vector matrix, which cannot be dynamically modified, it cannot solve the polysemy phenomenon and realize the real semantic understanding of text.

Dynamic context embedders are also known as pre-trained language models, which generate representations that change depending on the context, such as ELMo, BERT, ERNIE, ALBERT, NEZHA, etc. BERT is the most commonly used. For a given character, BERT takes its character position embedding, sentence position embedding and character embedding as input, and then uses mask language model (MLM) to pre-train the input sentence with deep bidirectional representation to obtain robust contextual character embedding. Because of BERT excellent performance, some scholars began to explore how to achieve similar results with fewer training resources [14-16]. Therefore, RoBERTa, SpanBERT and other improved models based on BERT model have been proposed successively. Such models limit the length of input sequences and thus do not perform well on generative tasks such as automatic text summarization. To solve this problem, XLNet extracts long text features by applying Transformer-XL. In addition, the ERNIE model [17] also focuses more on word vector training in the Chinese field.

2.2. The NER Correlation Between Chinese and English

In recent years, the superior nature of deep learning in the field of NLP, which can learn feature representations directly from data, has brought significant breakthroughs in the field. In terms of English named entity recognition, deep learning models have also significantly improved their performance. At the same time, compared with Chinese named entity recognition technology, NER technology of English text has reached a relatively

mature stage due to the unique English word segmentation rules, that is, the natural space barrier between words. Therefore, in recent years, scholars at home and abroad began to apply English NER technology to Chinese named entity recognition, because both Chinese and English texts have obvious grammatical and lexical characteristics. Secondly, both Chinese and English are context-important languages, and the context information of entities has an important impact on entity recognition. In addition, the problems faced by Chinese NER and English NER are similar, for example, the problem of unknown words. With the development of various fields and the era of big data, there will be a large number of new entities, but these new entities lack uniform naming rules in the dictionary. Therefore, named entity recognition (NER) requires strong contextual reasoning ability to identify nested entities in both English and Chinese, including both outer and inner entities. This is one of the hot topics in current NER research. At the same time, both Chinese and English NER have the problem of text ambiguity, and the same text may represent different entity types in different positions, so it is necessary to conduct entity disambiguation before NER [18,19].

To sum up, due to the particularity of Chinese language and the late start of NER in Chinese, some deep learning methods directly applied to Chinese named entity recognition tasks cannot achieve the same good results as English named entity recognition. Therefore, it is difficult to recognize Chinese named entities. The particularity and difficulty of NER in Chinese are reflected in the following points:

1. The boundary of Chinese words is fuzzy. Unlike English texts, Chinese texts do not have displayed separators (such as spaces) and obvious parts-of-speech changes (for example, place names and personal names in English are capitalized) as boundary markers. Therefore, it is difficult to determine the segmentation boundary.
2. Nested entities. Entity contains other entities or is contained by other entities, and it is one of the current research hotspots to identify both internal and external entities.
3. Entity ambiguity. In the results of entity recognition, there may be different references to the same entity, or there may be multiple meanings of one word, which will lead to inaccurate and ambiguous results of entity recognition. Therefore, it is necessary to disambiguate the entity recognition results before obtaining accurate and unambiguous information.
4. Low resource NER. At present, for limited domains and limited entity types, named entity recognition can achieve good recognition results in these places, and cannot be transferred to other specific domains.

2.3. Flat Solid Boundary Problem

The process of named entity recognition usually includes two parts: (1) entity boundary recognition; (2) Identify the type of entity (person, place, institution or other) [20-22]. So determining the entity boundary plays an important role in named entity recognition. NER entities can often be identified by some obvious formal sign, such as a place or person entity capitalized in a word. Therefore, entity boundary identification is relatively easy in English. However, compared with English, Chinese named entity recognition task is more complicated. This is because Chinese entities often do not have obvious formal signs, and the composition of entities is more complex. Compared with entity category labeling subtask, entity boundary recognition is more difficult in Chinese NER task. Therefore,

in the Chinese NER task, it is necessary to adopt more complex and fine algorithms for entity recognition and boundary recognition to achieve higher accuracy and recall rate.

Some researchers have used character-based approach to solve NER, which has achieved good performance, but can not use word boundary and word order information to determine the entity boundary. In recent years, with the introduction of deep learning, NER research mainly focuses on the feature that there is no clear boundary between Chinese words. In the process of research, it is found that the introduction of external resources can provide boundary information for the Chinese named entity recognition model which is not completely based on words. Thus, the model performance is improved, which is regarded as one of the auxiliary tools to improve the model performance. Therefore, the method of determining entity boundary can be roughly summarized into two aspects: word segmentation and Chinese word feature fusion. Gate Recurrent Unit (GRU) is a special kind of Recurrent Neural Network (RNN) [23,24]. GRU is similar to LSTM in performance but simpler and more efficient, and the training time is greatly shortened while retaining important timing features. Yu et al. [25] used GRU to capture the long-distance features of Chinese characters and applied them to Chinese named entity recognition, and achieved good results. Setiawan et al. [26] used BiGRU to extract character-level word vectors as part of the input in fine-grained sentiment analysis, and the results showed that character-level features extracted by BiGRU had a positive impact on model performance.

Relevant studies have shown that character-level features are effective in biomedical named entity recognition tasks, but different extractors have different character-level features, and the effect of using a single character-level feature extractor is limited. Therefore, this paper uses CNN and BiLSTM to extract different types of character-level features, adaptively integrates the two character-level features in different contexts, and proposes an adaptive named entity recognition model based on character-level features.

3. Proposed Adaptive Entity Recognition Method

3.1. Character Level Feature Extraction

Character-level features have been proved to be effective in various NLP tasks and can improve the performance of such tasks. References [27,28] shows that character-level features can significantly improve the performance of machine translation. Tran et al. [29] had applied character-level features to text classification to improve the performance to some extent. The advantage of using character-level features was that they could be extracted directly from the original text without designing additional manual features, and there was no need for complex preprocessing of the original corpus. In this paper, CNN, BiLSTM and Bidirectional Gate Recurrent Unit (BiGRU) are used to extract character-level features of words.

A. Character level CNN model

CNN is suitable for feature extraction of presuffixes and composition of words. The structure of CNN is shown in Figure 1, and the specific process is as follows. Each word in the original text is disassembled and filled to the maximum word length, so that the dimension size of the character vector matrix is consistent. The local features of the current word are extracted from the matrix formed by the characters of each word through convolution operation. The size of convolution kernel determines the range of local features

that can be extracted by CNN. The key information in the feature is extracted through the pooling process. The result is a 30-dimensional character vector.

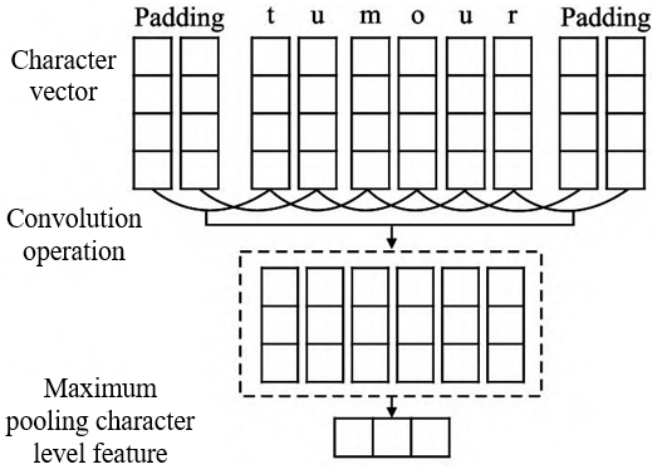


Fig. 1. Schematic diagram of character level feature extraction by CNN

B. Character level BiLSTM model

BiLSTM is good for spelling information about words. The structure of character features extracted by BiLSTM is shown in Figure 2. First, each character of the word is read from front to back to calculate the forward character vector V_f, and then the backward character vector is calculated by reading the characters of the word from back to front. The forward vector and the backward vector are connected in series to obtain the vector V_T of the word character level. V_T retains both forward and backward information and is ultimately represented as a 30-dimensional character vector.

C. Character-level BiGRU model

Different character-level feature extraction models have different characteristics. CNN is suitable for extracting local features and representing information such as presuffixes and compositions of words, while it lacks attention to long-distance dependent information. BiLSTM and BiGRU pay more attention to timing features, which are suitable for representing the character spelling information of words, and pay less attention to local features. Deng et al. [30] directly spliced the character-level features extracted by CNN and BiLSTM to form combined character features and took them as part of the input in biomedical named entity recognition. The experimental results showed that the effect of combined features was better than that of single features. Xu et al. [31] extracted character-level features from BiGRU and then extracted local features in the convolution process for Chinese named entity recognition, and achieved good results.

Different character-level feature extractors focus on different points. When using different extractors to extract features, if different weights can be assigned to different features, so that the model can make full use of information in character granularity, it will have a positive impact on the performance of the model. For a sentence

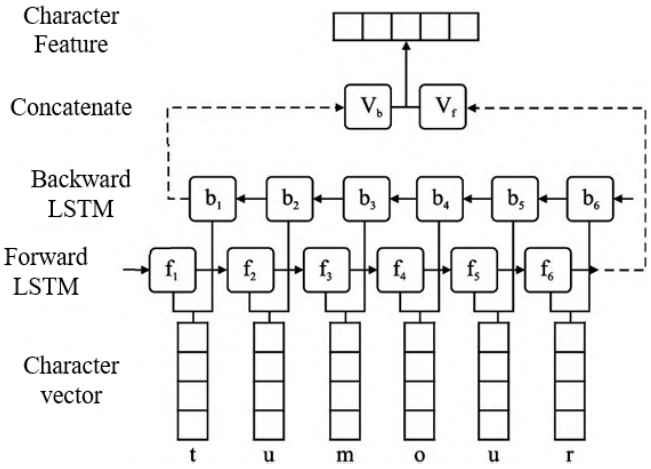


Fig. 2. Schematic diagram of character level feature extracted by BiLSTM

$S = W_1, W_2, \cdots, W_n$, CNN is used to extract the local feature V_P of the word character level in S, and one of BiLSTM or BiGRU is used to extract the sequence feature V_T of the character level. V_P is consistent with the dimension of V_T, and $\tilde{V}$ is used to represent the character-level combination feature.

The first strategy. According to the gradient method, the character features extracted by different extractors are given different weights on the macro level to test the degree of influence on the performance of the model. For example, the weight assigned to V_P is α_0, the weight assigned to V_T is β_0, and $\oplus$ represents the concatenation operation. The combination process satisfies formula (1), and $0.1 \leq \alpha_0 \leq 1.9, 0.1 \leq \beta_0 \leq 1.9$. When α_0 and β_0 are equal, it is equivalent to V_T, and $\tilde{V}$ and is directly concatenated. The process of generating character-level combination feature $\tilde{V_0}$ under this strategy can be expressed as:

$$\alpha_0 + \beta_0 = 2. \tag{1}$$

$$\tilde{V_0} = \alpha_0 V_P \oplus \beta_0 V_T. \tag{2}$$

Artificially controlling the weights of different types of character features may have positive or negative effects on model performance, which is highly uncertain. Therefore, the second strategy is proposed. To calculate dynamic weights in the process of model training, different types of extractors are selected to extract character level features of words. In the training process, the weight of each feature is dynamically calculated, so that the weight of important features becomes larger, the weight of unimportant features becomes smaller, the important part is strengthened, and the unimportant part is weakened. The process of generating character-level combination feature $\tilde{V_1}$ under the adaptive strategy can be expressed as:

$$Mish(x) = x \times (\tanh(\ln(1 + e^x))). \tag{3}$$

$$z_1 = \sigma(Mish(V_P)). \tag{4}$$

$$z_2 = \sigma(Mish(V_T)). \tag{5}$$

$$\alpha_1 = 2 \times \frac{z_1}{z_1 + z_2}. \tag{6}$$

$$\beta_1 = 2 - \alpha_1. \tag{7}$$

$$V'_P = \alpha_1 \times V_P. \tag{8}$$

$$V'_T = \beta_1 \times V_T. \tag{9}$$

$$\tilde{V_1} = V'_P \oplus V'_T. \tag{10}$$

In the above formula, $Mish$ represents the Mish activation function, which allows better information to penetrate into the neural network. σ is the sigmoid activation function. z_1 and z_2 are transition matrices after a series of nonlinear transformations. Each number in z_1 and z_2 ranges from 0 to 1. α_1 represents the weight matrix of the local eigenmatrix. β_1 represents the weight matrix of the temporal eigenmatrix. For each word in the input text, the model selectively strengthens or weakens the character-level local features corresponding to the word, and weakens or strengthens the character-level temporal features of the word. The operation trend of the two features is opposite. Finally, the two transformed character level features are concatenated to obtain character level combination features.

3.2. BERT

In the whole pre-training process, BERT adopts Transformer Encoder model to realize joint context constraint optimization among all layers, realize left-to-right bidirectional training, and can also carry out cross-information transfer between upper and lower layers, instead of just focusing on the text information of the current input words.

The Transformer Encoder consists of a multi-head self-attention mechanism layer and a feed-forward neural network layer. The multi-head attention mechanism layer is mainly composed of multiple attention mechanism units, and a single attention mechanism unit adopts the principle of scaled dot-product attention (SDPA) [32]. A fully bidirectional attention value is calculated for each input. After adding the input embedded word vector I to the positional encoding matrix representing each word position, the Q (Query) matrix, K (Key) matrix, and V (Value) matrix are generated by multiplying the matrix weights W_Q, W_K, and W_V. The Q matrix $[Q_1, Q_2, Q_3, \cdots]$ is generated by the current word. The K matrix $[K_1, K_2, K_3, \cdots]$ and V matrix $[V_1, V_2, V_3, \cdots]$ are generated by context words. As shown in formula (11), in a single attention mechanism unit, the input Q matrix of the current word is multiplied by $[K_1, K_2, K_3, \cdots]$ and then divided by the dimension d_k of the K matrix. After normalization by Softmax, the weight matrix is obtained and multiplied by $[V_1, V_2, V_3, \cdots]$ to obtain the self-attention unit Z_i. As shown in equation

(12), all self-attention units Z_i are splicing and linear mapping is performed through W to obtain multi-head attention matrix Z, which is used to reflect the relationship between current words and context words. Finally, Z is added and regularized with the original input embedding word vector I, and then added and regularized with the feedforward layer input matrix again through the feedforward neural network to obtain Transformer encoding vector.

$$Z_i = Softmax(\frac{Q \times K^T}{\sqrt{d_k}})V. \tag{11}$$

$$Z = Concat(Z_1, \cdots, Z_h)W. \tag{12}$$

3.3. Source Domain Pretrained LSTM Layer

LSTM neural network model has been widely used in feature extraction for sequence annotation tasks. LSTM can be used to train traditional news models with rich corpus sets. Reference [33] realized the fast training of the source domain model, as shown in Figure 3.

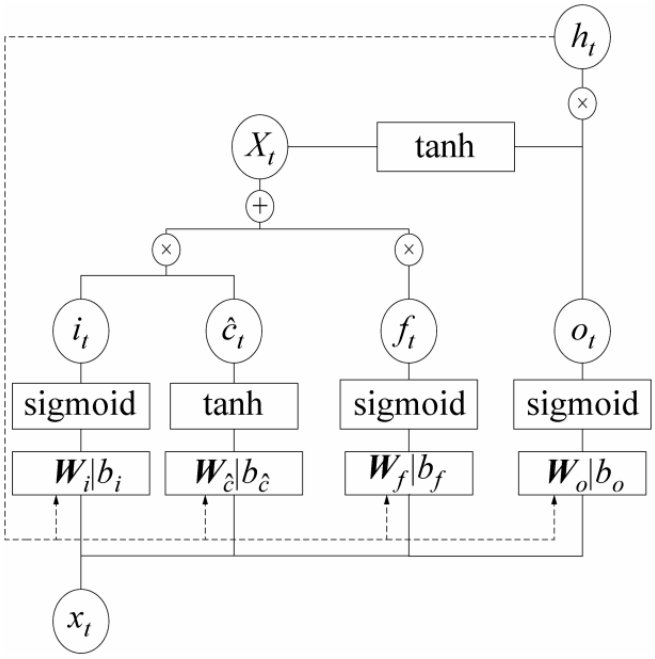


Fig. 3. LSTM structure

In the structure diagram of the LSTM unit, i_t, f_t and o_t respectively represent the states of the input gate, forget gate and output gate in the LSTM unit at time t. h_{t-1} indicates the hidden state at time $t-1$. c_t represents the cell state at time t. σ indicates

Sigmoid. $tanh$ represents the hyperbolic tangent excitation function, as shown in equations (13)-(18).

$$i_t = \sigma(W_{ih}h_{t-1} + W_{ix}x_t + b_i). \tag{13}$$

$$f_t = \sigma(W_{fh}h_{t-1} + W_{fx}x_t + b_f). \tag{14}$$

$$\tilde{c}_t = \sigma(W_{ch}h_{t-1} + W_{cx}x_t + b_c). \tag{15}$$

$$c_t = f_t c_{t-1} + i_t \tilde{c}_t. \tag{16}$$

$$o_t = \sigma(W_{oh}h_{t-1} + W_{ox}x_t + b_o). \tag{17}$$

$$h_t = o_t \tanh(c_t). \tag{18}$$

At t time, the word vector of BERT pre-training layer is used as the input of LSTM layer, and the hidden state of LSTM at $t-1$ time is combined to obtain the hidden state of t time. Finally, the output hiding state of BiLSTM can be obtained by combining the forward LSTM hiding state and the reverse LSTM hiding state.

3.4. BiLSTM Output Adaptation Layer Integrating Attention Mechanism

By introducing the attention mechanism to extract the feature vector in the context, the key words in the text will get more efficient feature extraction, so that the feature vector will integrate more contextual semantics. For the input sentence $L = [x_1, x_2, x_3, \cdots]$, it inputs the current word x_t of the BERT embedding word vector at time t. Firstly, the attention weight α_{tj} is calculated, where the greater the value of α_{tj} indicates that the current word x_t is more influenced by x_j, as shown in equation (19).

$$\alpha_{tj} = \frac{exp(score(x_t, x_j))}{\sum_{j=1}^{n} exp(score(x_t, x_j))}. \tag{19}$$

This paper uses the following four alignment algorithms to calculate $score(x_t, x_j)$ to measure the degree of association between the current word and the context word including Manhattan distance; Euclidean distance; Cosine distance; Perceptron mechanism. Here, the greater Manhattan distance and Euclidean distance denote the higher degree of correlation. The larger cosine distance and the perceptron denote the smaller correlation degree. The final result is $score(x_t, x_j) = a + b - c - d$, as shown in equation (20).

$$score(x_t, x_j) = \begin{cases} W_a|x_t - x_j| \\ W_a(x_t - x_j)^T(x_t - x_j) \\ \dfrac{W_a(x_t \cdot x_j)}{|x_t| \cdot |x_j|} \\ \tanh(W_a[x_t; x_j]) \end{cases} \tag{20}$$

Then, the output hidden state h_j and the attention weight α_{tj} at each moment of the BiLSTM frozen layer of the pre-trained source domain are weighted and summed to obtain the global variable g_t. After concatenating with the output hidden state h_t at the current moment, the linear mapping is carried out through matrix W_g, and the output M_t of the attention layer is obtained through $tanh$ activation function. Further feature extraction is carried out by randomly initialized BiLSTM, as shown in equations (21)-(22).

$$g_t = \sum_{j=1}^{N} \alpha_{tj} h_j. \tag{21}$$

$$M_t = \tanh(W_g[g_t; h_t]). \tag{22}$$

3.5. CRF Labeling Layer

From the context features extracted from the BiLSTM output adaptation layer that incorporates the attention mechanism, the CRF layer can further consider the dependencies between sequence labels. The emission matrix P_{x_t,y_t} is the x_t output probability given for the BiLSTM output adaptation layer. The transition matrix T_{y_{t-1},y_t} represents the probability of transferring the output y_{t-1} to the output y_t from time $t-1$ to time t. All preset labels are normalized to obtain the probability of the current output label. Given the input x_t, the score $s(x_t, y_t)$ of the output y_t is obtained. The probability of the current output label is obtained by normalization, as shown in equations (23) and (24).

$$s(x_t, y_t) = \sum_{i=1}^{n} (T_{y_{t-1},y_t} + P_{x_t,y_t}). \tag{23}$$

$$P(y_t|x_t) = \frac{exp(s(x_t, y_t))}{\sum_{y_t \in y} exp(s(x_t, y_t))}. \tag{24}$$

In the training process, Viterbi [34] algorithm is used for maximum likelihood estimation to maximize the probability of the model predicting the calibration label for the input text, and the expression of Loss is obtained, as shown in equation (25).

$$Loss = \underset{\tilde{y}}{argmax}\, s(x, \tilde{y}). \tag{25}$$

3.6. Training Process

In the process of transfer learning training, when pre-training the language model source domain BERT, it is inevitable to encounter some unknown words, which make the model have to generate new vocabularies and train them. In order to make the best use of the rich semantic features in the pre-trained word vector model, this paper only trains the high frequency unknown words and integrates them with the source language model. After initialization, freeze BERT language model parameters to avoid overfitting.

The LSTM model parameter θ_S trained by the source domain data set D_S is taken as the LSTM parameter θ_T of the target domain model. Because of the difference between

the feature semantic space of source domain and target domain, the radical training strategy will cause the whole model to lose the feature information of source domain language which is needed for the optimization of target domain. However, if the training source domain neural network is updated too cautiously, the iterative efficiency of the whole model will decrease, and overfitting will easily occur. In the transfer learning of this paper, the hyperparameter ξ is set as the ratio of the learning rate of the LSTM layer and the adaptive neural network layer that integrates the attention mechanism. The specific definitions are shown in equations (26) and (27).

$$lr_S = \sigma \times lr_T. \tag{26}$$

$$\sigma = \frac{Count_T}{Count_S} \times \xi. \tag{27}$$

Where $Count_T$ and $Count_T$ are the number of tokens in the source domain and target domain data set respectively, and ξ is the hyperparameter. In this paper, the new adaptive neural network layer is inserted into the last layer. Because this layer generally contains the least amount of knowledge, the source domain model dominates transfer learning.

4. Experiment and Analysis

To verify the validity of the proposed method, experiments are conducted on five English datasets and a specialized cross-domain dataset. The experimental analysis is carried out from four aspects, namely ablation experiment, saliency detection, and fine-grained analysis.

4.1. Experimental Data

The five English datasets are CoNLL-2003 (Conll03), Twitter (T), Broad Twitter(BT), BioNLP13PC (PC), and BioNLP13CG (CG). CoNLL-2003, Twitter, and Broad Twitter datasets are similar domains with roughly similar entity types, including person (PER), location (LOC), and organization (ORG). CoNLL-2003 contains more miscellaneous (MISC) entities than Twitter. The BioNLP13PC dataset and BioNLP13CG dataset belong to the medical and biological fields, and the entity types mainly include simple chemical (CHEM), cellular component (CC), gene and gene product (GGP). BioNLP13CG also includes species (SPE) and cell. The specific data set statistics are shown in Table 1. The cross-domain dataset [35] is a specialized cross-domain NER dataset CrossNER, which contains five domains, namely politics, naturalscience, music, literature, and artificial intelligence intelligence (AI). Each domain contains a specific entity type, and the specific data set statistics are shown in Table 2.

According to the different types of entities in the data set and the differences in related fields, the experiment can be divided into two groups. Group 1: The same experimental group is selected from 5 English data sets. For example, when Twitter and Broad Twitter are used as target domain data sets, CoNLL2003 is used as source domain data. When BioNLP13PC is used as the target domain data set, CoNLL-2003 is selected as the source domain data set, and experiments are conducted between different domains and similar

Table 1. English data set statistics

Data set	Type	Training number	Verification number	Testing number
Conll03	Sentence	15100	3500	3700
Conll03	Entity	23500	5900	5700
Conll03	Word	219600	55000	50300
T	Sentence	4300	1300	1500
T	Entity	7500	2500	2500
T	Word	68700	22900	23100
BT	Sentence	6300	1000	2000
BT	Entity	8800	1700	4300
BT	Word	106300	16000	37400
PC	Sentence	2500	900	1700
PC	Entity	7900	2700	5300
PC	Word	71700	24500	47900
CG	Sentence	3000	1000	1900
CG	Entity	10800	3600	6900
CG	Word	86500	28600	54700

Table 2. CrossNER data set statistics

Data set	Type	Training number	Verification number	Testing number
politics	Sentence	200	500	600
politics	Entity	1300	3400	4200
natural science	Sentence	200	400	500
natural science	Entity	1000	2500	3500
music	Sentence	100	300	400
music	Entity	600	2600	3300
literature	Sentence	100	400	400
literature	Entity	500	2100	2200
AI	Sentence	100	300	400
AI	Entity	500	1500	1800

domains to verify the effect of MSKECDNER's migration between different domains. Group 2: Five different domain-specific data are selected from the same experimental group as CrossNER as the target domain data set, and CONLL-2003 is selected as the source domain data.

4.2. Experimental Setup

For the five publicly available English data sets, it initializes the parameters of the model with reference NCRF++ [36]. As the data sets of source domain and target domain change in different experiments, the parameters of the model also change. For example, when the source domain is BioNLP13PC and the target domain is BioNLP13CG, the optimizer is SGD algorithm, the learning rate is set to 0.005, the learning rate decay is set to 0.01, the batch size is set to 10, the hidden state dimension is 250, and the hidden variable dimension is 200, and the dropout is set to 0.5 to prevent overfitting. When Twitter and Broad Twitter are used as the target domain data set, Glove 100-dim [37] is used for initialization to obtain the feature representation of word vector. When BioNLP13PC and BioNLP13CG are the target fields, PubMed 200-dim is used to initialize the word vector. The char vector adopts the form of random initialization, it extracts the character feature representation through the convolutional neural network, and finally splices the obtained word feature representation and character feature representation to get the final feature representation.

Based on the CrossNER data set, the parameters on five English data sets are initially set. After parameter adjustment, it is found that when the same parameters are used in most fields, the performance of MSKE-CDNER is better, which proves that the model has strong robustness. At the same time, the model performance is further improved when the model parameters in some domains are fine-tuned with changes in the target domain data set. For example, in the Music domain, the optimizer uses the SGD algorithm, the learning rate is set to 0.003, the learning rate decay is set to 0.03, the batch size is 32, the hidden state dimension is 250, and the hidden variable dimension is 200, and the dropout is set to 0.5 to prevent overfitting. In the experiments, Glove100-dim is used for initialization to obtain the feature representation of word vector, and BERT optimizes the feature representation of word vector obtained. The character vector is randomly initialized, the character feature representation is extracted by convolutional neural network, and the obtained word feature representation and character feature representation are spliced as the final feature representation.

In particular, during the experimental training of five English data sets, the end of a batch is marked when the end signal of the target domain is obtained. The read operation of the source domain data is not reset at the end of the batch; it continues to load the data until it reaches the end symbol of the source domain data. When performing experiments on the CrossNER dataset, the end of a batch is replaced by an end symbol that reads into both the source domain and the target domain data.

4.3. Evaluation Index

This paper adopts the evaluation index consistent with reference [38], which holds that the prediction is accurate only when the entity types and boundaries are correctly identified.

Precision (P), Recall (R) and F1 values are used to calculate the final score, which is calculated as follows:

$$P = \frac{TP}{TP + FP}. \tag{28}$$

$$R = \frac{TP}{TP + FN}. \tag{29}$$

$$F1 = \frac{2PR}{P + R}. \tag{30}$$

TP indicates the number of correctly identified entities, FP indicates the number of incorrectly identified entities, and FN indicates the number of unidentified entities.

4.4. Comparison Model

In order to verify the effect of the proposed method on cross-domain NER, comparative experiments with relevant models are conducted on different data sets.

1. BILSTM-CRF. BILSTM-CRF [39] combines two-way LSTM and conditional random fields for named entity recognition, combines source domain data with target domain data, and trains the model together.
2. Coach. Liu et al. [40] proposed a framework Coach with NER domain adaptation, which divided tasks into two stages, first detecting entities and then classifying entities to solve the problem of data scarcity in specific domains.
3. MULTI-TASK+PGN. Jia et al. [41] integrated language model tasks in the source domain and target domain to perform cross-domain knowledge transfer, thus solving the problem that the model could not be trained in an unsupervised environment.
4. MULTI-TASK+GRAD. Zhou et al. [42] proposed a new transmission method to integrate features under high and low resources through adversarial transmission networks, and introduced generalized resource adversarial discriminator to improve the generalization ability of the model.
5. MULTI-CELL+LSTM. Jia et al. [43] proposed a structure based on Bert representation, which was modeled for different entity types respectively and carried out cross-domain knowledge transfer at the entity level to solve the problem of different meanings of entities in different fields.

4.5. Experimental Result

The proposed method is compared with other relevant methods in five English data sets and CrossNER data sets, and the results are shown in Table 3 and Table 4. Overall, the proposed method achieves good results in different data sets.

As shown in Table 3, compared with BILSTM-CRF, F1 value with Coach method on Conll03 data, T, CG increases by 2.37%, 0.86%, 1.82% respectively. It shows that multi-task architecture can improve entity recognition capability in CD-NER. Compared with Coach, the F1 value of MULTI-TASK+PGN, MULTI-TASK+GRAD and MULTI-CELL+LSTM on CG increases by 0.11%, 0.57% and 0.95% respectively. It is suggested

Table 3. F1 value on English dataset/%

Method	Conll03	T	BT	PC	CG
BILSTM-CRF	77.29	73.09	82.14	76.47	79.35
Coach	79.66	73.95	83.20	77.84	81.17
MULTI-TASK+PGN	80.18	73.81	85.65	79.97	81.28
MULTI-TASK+GRAD	79.83	74.23	85.91	80.23	81.74
MULTI-CELL+LSTM	80.97	74.94	86.37	80.85	82.12
Proposed	81.09	74.56	86.72	82.01	83.04

that fully learning and use of domain invariant knowledge between features can alleviate the problem of poor model effect caused by lack of data resources. Compared with the current popular model MULTI-CELL+LSTM, the F1 value on CG is increased by 0.92%, indicating that structured semantic knowledge can promote cross-domain knowledge transfer and alleviate the problem of entity ambiguity. Among them, the Broad Twitter domain model is not effective, considering that Broad Twitter belongs to the news domain, Conl103 also belongs to the news domain, and there is little difference between fields. When cross-domain alignment is carried out, the constraint between graph matching and domain migration is poor, resulting in poor migration effect. In the BT group of experiments, BioNLP13PC belongs to the medical field, and the data between Conll03 and BioNLP13PC are quite different. Graph matching is a good constraint for cross-domain migration. The experimental comparison shows that the greater difference between domains shows the better model transfer effect, which also shows that structured information in semantic features can promote cross-domain knowledge transfer. The greater domain difference denotes the stronger constraining effect of structured knowledge and the better transfer effect. However, the existing research methods lack the mining and utilization of this kind of information. The multi-level structure transfer method in this paper can use structured information to enhance the cross-domain transfer ability of the model.

Table 4. F1 value on CrossNER dataset/%

Method	Politics	Science	Music	Literature	AI	Average
BILSTM-CRF	56.71	50.08	44.90	43.14	43.67	47.70
Coach	61.61	52.20	51.77	48.46	45.26	51.86
MULTI-TASK+PGN	68.55	64.42	63.67	59.70	53.81	62.03
MULTI-CELL+LSTM	70.67	66.53	70.63	67.07	58.39	66.66
Proposed	71.36	67.13	73.18	67.98	61.00	68.13

The results are shown in Table 4. Compared with BILSTM-CRF, the F1 value of MULTI-CELL+LSTM in five different fields has been improved, and the average F1 value has increased by 18.96%. Since BILSTM-CRF is a single-task model, it cannot make good use of the cross-domain invariant knowledge in the source domain, while MULTI-CELL+LSTM builds a network based on the multi-task architecture and can make full use of the cross-domain invariant knowledge in the source domain. Therefore, the multi-task learning paradigm is adopted as the basic framework when constructing the method in

this paper. Compared with MULTI-CELL+LSTM, the F1 value of the proposed method is significantly improved in five different fields. Among them, F1 value has increased by 0.69% in Politics, 0.60% in Science, 2.55% in Music, 0.91% in Literature, 2.61% in AI, and 1.47% in F1 average value. Since MULTI-CELL+LSTM only considers the feature information at the entity level, ignoring the structured knowledge in the feature information, the proposed method can effectively utilize the structured knowledge in the feature information, thus achieving better performance.

4.6. Ablation Experiment

In order to verify the effectiveness of the multi-level structured alignment mechanism, ablation experiments are conducted in the CG group, and the comparative results obtained are shown in Table 5. It can be seen that the three modules in the mechanism are helpful to the improvement of entity recognition performance. X represents the comparison difference of F1 values after different ablation methods.

Table 5. Ablation experiments on CG data set/%

Method	P	R	F1	X
Proposed	83.91	82.19	83.04	none
$\delta 1$	83.85	81.26	82.67	0.37
$\delta 2$	83.90	81.79	82.83	0.21
$\delta 3$	83.36	81.98	82.53	0.51

In Table 5, $\delta 1$ represents the result obtained when the boundary detection module is eliminated, P decreases by 0.06%, R decreases by 0.93%, and F1 value decreases by 0.37%. Among them, R is decreased the most in the three ablation experiments, indicating that learning common boundary information can help the model correctly identify entity types and improve the performance of the model. Adding this module can improve the effect of cross-domain named entity recognition. Similarly, $\delta 2$ represents the experimental results of the elimination of latent layer alignment module, P decreases by 0.01%, R decreased by 0.4%, and F1 value decreases by 0.21%, indicating that the common knowledge in the migration source domain and the target domain can promote the cross-domain migration of entity recognition ability. $\delta 3$ represents the result of removing the structured alignment module, P decreases by 0.55%, R decreases by 0.21%, and F1 value decreases by 0.51%. F1 value is decreased the most in the three ablation experiments, because the structured alignment module acquires structured knowledge while obtaining feature representation. During the migration, structured information can significantly improve the entity recognition performance of the target domain due to its cross-domain stability.

4.7. Significant Difference Test

The significance test is conducted in the CG experiment, and the results are shown in Table 6. The P value in the table is $Prob > F$, and when $P < 0.05$, it indicates that there is a significant difference between PC and CG. In Table 6, $P = 0.0278$ indicates

that there are significant differences between the methods in this paper. F is the statistic of the test; P is the P value used for testing.

Table 6. Analysis of variance on $PC \rightarrow CG$ dataset

Source	Quadratic sum	Degree of freedom	Mean square error	F	P
Group	1.0087	1	1.0087	11.45	0.0278
Error	0.35254	4	0.08814		
Total	1.36115	5			

4.8. Fine-grained Analysis

Table 7 shows the fine-grained experimental results of the proposed method in CG experiments. The P, R and F1 values of the relevant entities are recorded in Table 7. Because there are many types of entities in the relevant data set, for the sake of discussion, the entity types are randomly selected for example. Compared with the current cross-domain models, F1 values of all models are improved, and the overall F1 value is increased by nearly 0.92%, which confirms the effectiveness of the proposed method.

Table 7. Fine-grained analysis on CG data set/%

Entity Type	P	R	F1
Cell component	82.69	81.78	82.23
Multi-organization structure	77.29	76.02	76.65
Biology	87.45	80.03	83.58
Simple chemistry	81.16	74.25	77.55
Organization	65.31	80.54	72.13

In order to clearly compare the method presented in this paper to obtain better results at the entity type level, it is compared with the original model for fine-grained analysis in different types of entities, and the results are shown in Table 8. Under the same entity type, the results of the proposed method are significantly better than the other two methods, which is due to the fact that the structured knowledge inside the entity effectively alleviates the problem of entity ambiguity in different domains when carrying out cross-domain migration.

5. Conclusion

This paper proposes an English text named entity recognition model based on character-level feature adaptive network. The model uses CNN to extract local features of word character sequences, and BiLSTM to extract sequential features of word sequences. In the process of model training, different weights are dynamically assigned to the two

Table 8. Comparison of fine-grained F1 values between different models/%

Entity Type	Baseline	MULTI-CELL+LSTM	Proposed
Cell component	78.81	79.67	82.23
Multi-organization structure	75.45	75.15	76.65
Biology	81.21	81.52	83.58
Simple chemistry	76.23	73.57	77.55
Organization	70.14	70.45	72.13

character-level features, which further strengthens important character-level features, weakens unimportant character-level features, and makes full use of information in character granularity. After the combined character-level features are obtained, the part-of-speech information and block analysis features are used as additional features to assist the model to judge the entity boundary, which further improves the performance of the model. Experiments are carried out on five English datasets and CrossNER datasets, and compared with current cross-domain methods, the results show that the proposed method achieves better results in cross-domain tasks, indicating that learning and using structured knowledge can better promote cross-domain knowledge transfer. In future work, we will better decouple domain invariant knowledge and domain specific knowledge to obtain better feature representation.

Acknowledgments. This work was supported by Henan Provincial Education Reform Project "Reform and Practice of Practical Teaching System for Applied Translation Undergraduate Majors from the Perspective of Technology Hard Trend" in 2024 (Project number: 2024SJGLX0581) and Zhengzhou University of Science and Technology Key Education Reform project "Innovation Research on Practical Teaching of Translation with Digital-Intelligence Technology Enabling Production and Education" in 2024 (Project number: 2024JGZD11) of phased research results.

References

1. Ehrmann M, Hamdi A, Pontes E L, et al. Named entity recognition and classification in historical documents: A survey[J]. ACM Computing Surveys, 2023, 56(2): 1-47.
2. Liu P, Guo Y, Wang F, et al. Chinese named entity recognition: The state of the art[J]. Neurocomputing, 2022, 473: 37-53.
3. Zhao Y, Li H, Yin S. A multi-channel character relationship classification model based on attention mechanism[J]. Int. J. Math. Sci. Comput.(IJMSC), 2022, 8: 28-36.
4. Ravikumar J, Kumar P R. Machine learning model for clinical named entity recognition[J]. International Journal of Electrical and Computer Engineering, 2021, 11(2): 1689-1677.
5. Khan W, Daud A, Shahzad K, et al. Named entity recognition using conditional random fields[J]. Applied Sciences, 2022, 12(13): 6391.
6. Zhang R, Zhao P, Guo W, et al. Medical named entity recognition based on dilated convolutional neural network[J]. Cognitive Robotics, 2022, 2: 13-20.
7. Su S, Qu J, Cao Y, et al. Adversarial training lattice lstm for named entity recognition of rail fault texts[J]. IEEE Transactions on Intelligent Transportation Systems, 2022, 23(11): 21201-21215.
8. Carbonell M, Riba P, Villegas M, et al. Named entity recognition and relation extraction with graph neural networks in semi structured documents[C]//2020 25th International Conference on Pattern Recognition (ICPR). IEEE, 2021: 9622-9627.

9. Uzair M, Mian A. Blind domain adaptation with augmented extreme learning machine features[J]. IEEE transactions on cybernetics, 2016, 47(3): 651-660.
10. Liang Y, Lv H, Li Y, et al. Tibetan-BERT-wwm: A Tibetan Pretrained Model With Whole Word Masking for Text Classification[J]. IEEE Transactions on Computational Social Systems, 2024.
11. Chen X, Cong P, Lv S. A long-text classification method of Chinese news based on BERT and CNN[J]. IEEE Access, 2022, 10: 34046-34057.
12. Teng L, Qiao Y. BiSeNet-oriented context attention model for image semantic segmentation[J]. Computer Science and Information Systems, 2022, 19(3): 1409-1426.
13. Pokharel S, Sidorov E, Caragea D, et al. NLP-based encoding techniques for prediction of post-translational modification sites and protein functions[M]//Machine Learning in Bioinformatics of Protein Sequences: Algorithms, Databases and Resources for Modern Protein Bioinformatics. 2023: 81-127.
14. van Erkelens A M, Thompson N A, Chalmers D. The dynamic construction of an incubation context: a practice theory perspective[J]. Small Business Economics, 2024, 62(2): 583-605.
15. Lambooy J. The transmission of knowledge, emerging networks, and the role of universities: an evolutionary approach[J]. European Planning Studies, 2004, 12(5): 643-657.
16. Juarrero A. Dynamics in action: Intentional behavior as a complex system[J]. Emergence, 2000, 2(2): 24-57.
17. Wang Y, Sun Y, Ma Z, et al. An ERNIE-based joint model for Chinese named entity recognition[J]. Applied Sciences, 2020, 10(16): 5711.
18. Peng D L, Wang Y R, Liu C, et al. TL-NER: A transfer learning model for Chinese named entity recognition[J]. Information Systems Frontiers, 2020, 22(6): 1291-1304.
19. Xi Q, Ren Y, Yao S, et al. Chinese named entity recognition: applications and challenges[J]. MDATA: A New Knowledge Representation Model: Theory, Methods and Applications, 2021: 51-81.
20. Chen Y, Wu L, Zheng Q, et al. A boundary regression model for nested named entity recognition[J]. Cognitive Computation, 2023, 15(2): 534-551.
21. Vashishth S, Newman-Griffis D, Joshi R, et al. Improving broad-coverage medical entity linking with semantic type prediction and large-scale datasets[J]. Journal of biomedical informatics, 2021, 121: 103880.
22. Zhang D, Wei S, Li S, et al. Multi-modal graph fusion for named entity recognition with targeted visual guidance[C]//Proceedings of the AAAI conference on artificial intelligence. 2021, 35(16): 14347-14355.
23. Yin S, Li H, Laghari A A, et al. An Anomaly Detection Model Based On Deep Auto-Encoder and Capsule Graph Convolution via Sparrow Search Algorithm in 6G Internet-of-Everything[J]. IEEE Internet of Things Journal, 2024.
24. Yin S. Object Detection Based on Deep Learning: A Brief Review[J]. IJLAI Transactions on Science and Engineering, 2023, 1(02): 1-6.
25. Yu C, Wang S, Guo J. Learning Chinese word segmentation based on bidirectional GRU-CRF and CNN network model[J]. International Journal of Technology and Human Interaction (IJTHI), 2019, 15(3): 47-62.
26. Setiawan E I, Ferry F, Santoso J, et al. Bidirectional GRU for Targeted Aspect-Based Sentiment Analysis Based on Character-Enhanced Token-Embedding and Multi-Level Attention[J]. International Journal of Intelligent Engineering & Systems, 2020, 13(5).
27. Wang F, Chen W, Yang Z, et al. Hybrid attention for Chinese character-level neural machine translation[J]. Neurocomputing, 2019, 358: 44-52.
28. Lee J, Cho K, Hofmann T. Fully character-level neural machine translation without explicit segmentation[J]. Transactions of the Association for Computational Linguistics, 2017, 5: 365-378.
29. Tran P, Dinh D, Nguyen H T. A character level based and word level based approach for Chinese-Vietnamese machine translation[J]. Computational intelligence and Neuroscience, 2016, 2016.

30. Deng J, Cheng L, Wang Z. Attention-based BiLSTM fused CNN with gating mechanism model for Chinese long text classification[J]. Computer Speech & Language, 2021, 68: 101182.
31. Xu C, Wang F, Han J, et al. Exploiting multiple embeddings for chinese named entity recognition[C]//Proceedings of the 28th ACM international conference on information and knowledge management. 2019: 2269-2272.
32. Leelaluk S, Minematsu T, Taniguchi Y, et al. Scaled-Dot Product Attention for Early Detection of At-risk Students[C]//2022 IEEE International Conference on Teaching, Assessment and Learning for Engineering (TALE). IEEE, 2022: 316-322.
33. Lin J C W, Shao Y, Djenouri Y, et al. ASRNN: A recurrent neural network with an attention model for sequence labeling[J]. Knowledge-Based Systems, 2021, 212: 106548.
34. Di Y, Li R, Tian H, et al. A maneuvering target tracking based on fastIMM-extended Viterbi algorithm[J]. Neural Computing and Applications, 2023: 1-10.
35. Chen Y, Zhong H, He X, et al. Real20M: A Large-scale E-commerce Dataset for Cross-domain Retrieval[C]//Proceedings of the 31st ACM International Conference on Multimedia. 2023: 4939-4948.
36. Thu Y K, Aung T, Supnithi T. Neural Sequence Labeling Based Sentence Segmentation for Myanmar Language[C]//Conference on Information Technology and its Applications. Cham: Springer Nature Switzerland, 2023: 285-296.
37. Sarang P. ANN-Based Applications: Text and Image Dataset Processing for ANN Applications[M]//Thinking Data Science: A Data Science Practitioner‘s Guide. Cham: Springer International Publishing, 2023: 289-327.
38. Yin S, Li H, Teng L, et al. Attribute-based multiparty searchable encryption model for privacy protection of text data[J]. Multimedia Tools and Applications, 2023: 1-22.
39. Knight K, Nenkova A, Rambow O. Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies[C]//Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2016.
40. Liu J, Yu M, Chen Y, et al. Cross-domain slot filling as machine reading comprehension: A new perspective[J]. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2022, 30: 673-685.
41. Rasmussen N F, Jensen K N, Placenti M, et al. Cross-Domain Sentiment Classification using Vector Embedded Domain Representations[C]//Proceedings of the First NLPL Workshop on Deep Learning for Natural Language Processing. 2019: 48-57.
42. Zhou J T, Zhang H, Jin D, et al. Dual adversarial neural transfer for low-resource named entity recognition[C]//Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. 2019: 3461-3471.
43. Jia C, Zhang Y. Multi-cell compositional LSTM for NER domain adaptation[C]//Proceedings of the 58th annual meeting of the association for computational linguistics. 2020: 5906-5917.

Liuxin Gao is with the School of Foreign Languages, Zhengzhou University of Science and Technology. Several papers had been published related to the work in this paper. Research direction is English text analysis, English data analysis..

Received: April 18, 2024; Accepted: June 29, 2024.

Computer Science and Information Systems 21(4):1887–1911 https://doi.org/10.2298/CSIS240301064R

Identifying Occurrences of the Cnidarian *Physalia physalis* in Social Media Data*

Heloisa F. Rocha[1], Lorena S. Nascimento[2], Leonardo Camargo[1], Mauricio A. Noernberg[2], Aurora T. Ramirez Pozo[1], and Carmem S. Hara[1]

[1] Universidade Federal do Paraná, Departamento de Informática, Curitiba-PR, Brazil
heloisaferr@gmail.com
{camargo.s.leonardo, aurora.pozo, carmemhara}@ufpr.br
[2] Universidade Federal do Paraná, Centro de Estudos do Mar, Pontal do Paraná-PR, Brazil
{lorena.sn, m.noernberg}@ufpr.br

Abstract. The Portuguese man-of-war (*Physalia physalis*), though beautiful, poses a risk to the population due to its potential to cause severe burns. Tracking their occurrences can prevent accidents through alerts to the population and predictive simulation models. However, traditional sources do not always provide records of their sightings. On the other hand, previous studies indicate that social media can be an effective source of information for conservation science. This work uses natural language processing and computer vision to obtain machine learning models to classify data extracted from Instagram. Such models can be used as part of an automated Extract-Transform-Load process to build a database on occurrences of *Physalia physalis* on the Brazilian coast. In preparation for training the models, we collected and manually labeled Instagram posts in order to distinguish the ones about the animal from other subjects, such as ships and tattoos. Given the nature of the problem, the spatial and temporal information associated with the sightings are essential for biologists. Thus, the absence or nonvalidity of such data is often used as a rationale to reject the post. However, the same criteria may not be suitable for training machine learning models to classify new posts automatically. The main goal of this article is to highlight the importance of choosing appropriate labels to train both text and image models, as well as to take into consideration the rejection criteria of the biologist before using a classification model. An experimental study is presented to show the effect of unquestioning adoption of labels given by a specialist, compared to labels adapted for machine learning training.

Keywords: Physalia physalis, dataset construction, labeling, machine learning.

1. Introduction

Portuguese man-of-war (cnidarian *Physalia physalis*) are multicellular organisms whose tentacles have stinging cells that can release harmful toxins. They often appear along the Brazilian coast, and due to their beautiful colors, people tend to touch them. As a result, there are many cases of envenomation, which may lead to severe injuries [6].

* This work is an extended version of the conference paper [26]

Although it is mandatory to be reported, SINAN (the Brazilian injury reporting system) lacks specific indications for *Physalia physalis* poisoning. Moreover, evidence suggests significant underreporting of cases [6]. The limited information about these animals in official public databases lead researchers to look for alternative sources of information.

One such alternative is to use involuntary data from social media. This approach, called passive citizen science, consists of using data generated by non-professionals, collected and shared on the Internet, mainly on social media [12].

Among the advantages resulting from using social media data are [12]: abundant and almost always public content; less labor intensive, less time consuming, and with lower cost, especially if automated, compared to traditional methods; data available almost in real-time and continuously; and data acquisition with wide geographic scope [22]. In addition, the use of information technology tools, such as machine learning, makes it possible to explore the potential of social media data and turn it into useful information.

In the *Caravela* project [23], approved by the Federal University of Paraná ethics committee, one of the specific goals is to compile data extracted from social media about occurrences of *Physalia physalis* on the Brazilian coast. This process was performed manually [24] and the results showed that the posts collected from Instagram represented 60% of the entire dataset, which included other sources such as newspapers (7%), iNaturalist[3] (19%), reports in the literature (8%), and personal observations (6%). The goal of the study was to determine the effectiveness of Instagram as a data source for spatial and temporal information of their sightings, as well as obtain the number, size and types of interactions with other species and human beings.

In this article, we report on the annotation process to show the importance of properly designing the labels that will be associated with each entry, in order to prepare the dataset for training machine learning models. The labeling process is required to distinguish between posts that are legitimate sightings of the species from others, such as tattoos and Portuguese boats.

It has been a rich real-world application experience. It contrasts with other machine learning works that consider benchmark datasets for which labels were already given. The design of the resulting labels involved a close interaction between a computer scientist and an oceanographer, with discussions of the requirements for the end user and for the training process to automate the classification task. Such a classifier can later be inserted into a search, classification and data storage system, which can continuously monitor the species.

During the annotation process, the oceanographer adopted some criteria for accepting a post as a legitimate occurrence, which included the requirement that its location is on the Brazilian coast. However, for the computer scientist, the spatial information may not be important for a model that classifies posts based only on their text and/or image. More than that, the dataset annotated by the specialist may introduce noise for training these models. To this end, the experiments presented in this article show that models trained with the adapted labels perform better than models trained with the labels originally given by the oceanographer. This work is an extended version of a conference paper [26]. The extension includes experiments with two adapted labels instead of only one, as presented in the original work. In the textual analyses, we added experiments with multilingual BERT, as a classifier and also as a feature extractor to feed a Logistic Regression model. In all

[3] https://www.inaturalist.org/

experiments, we used class weighting to deal with imbalanced data and cross-validation. In the original paper, we used these techniques only in textual analyses.

Moreover, considering that in the future the trained model could be used to develop a system to continuously extract posts from Instagram and automatically classify them as legitimate occurrences of *Physalia physalis*, we considered two different flows of this process. In the first one, all the collected posts are given as input to the trained model. In the second, posts that do not meet the spatial requirement are filtered out before being evaluated by the classifier. The latter closely resembles what is expected to be adopted in the real production chain. Our experiments show that it improves all metrics of the classifier compared to the former approach.

In summary, the contributions of the article are threefold:

- Show the importance of the knowledge on the part of the computer scientist of which criteria were used to label the data: our experiments show that adapting labels can produce better classification models;
- Compare the effect of the adapted labels on different modals (text and image), in the context of conservation science and social media data;
- Determine the performance of the classifier adopting two different flows of tasks: with and without prior filtering of posts based on the spatial criteria.

The rest of the article is organized as follows. Section 2 presents some related works that apply machine learning on data extracted from social media. The *Physalia physalis* dataset construction is described in Section 3, followed by details on the annotation process in Section 4. Experiments with adapted labels are presented in Section 5, while experiments simulating the production flow are presented in Section 6. Section 7 concludes the article by presenting some future works.

2. Related Work

Several works highlight the advantages and possibilities of using involuntary data, such as those available on social media, for tasks related to conservation science [12, 22]. In addition, there are some studies that apply machine learning to detect aspects of the environment, such as estimating the number of animals of a given species [14] and classifying land cover [13]. One can also find studies using Instagram to monitor other marine species [29, 19], but without the use of automated tools.

However, as already noted in other studies [15, 11], there are few works that applied machine learning to involuntary data for wildlife classification tasks. In particular, when searching for works that applied Natural Language Processing (NLP), it was not possible to find works that used texts in Portuguese extracted from Instagram to identify wildlife. In this Section, we report 6 works related to conservation science that used machine learning for wildlife classification tasks using involuntary data.

Mazars-Simon [20] developed a system that searches, classifies, identifies, and stores data on sea turtles. He used a dataset with 22,500 images, divided into six categories, which were shared by conservationists for the project. The author did not explain how the categories were defined.

In [18], the authors used 5,464 texts extracted from Google News and Twitter[4] to train a classifier to identify news about endangered species. They created the dataset automatically considering all data extracted from Twitter as positive and the data extracted from Google News as negative.

Another work [11] experimented with different combinations of representation techniques and machine learning algorithms to identify wildlife observations in Twitter texts. A dataset with 2,798 tweets was used. In their work, they described the collection and labeling process.

Convolutional neural networks to identify *Physalia physalis* in images extracted from Instagram were reported in [5]. A dataset with 12,300 images was used. Part of the training images were obtained from Instagram and labeled by an expert, and part was downloaded through search engines and specialized websites.

Cardoso et al. [4] investigated whether pre-trained and freely available deep learning models could support the identification of potential wildlife trade on online platforms. To train the models they collected 2.634 images of pangolins from iNaturalist, Flickr[5] and Google images. In their work, they described the collection and labeling process.

In [17], the authors implemented and evaluated a hierarchical text classification pipeline aiming to investigate the utility of machine learning in automating online analyses of wildlife exploitation. To evaluate the proposed pipeline they collected data on bats (Chiroptera) from Facebook[6], Twitter, Google, and Bing search engines. After filtering and deduplication they had a dataset of 413,937 unique social media posts and 104,279 web articles. The authors described the collection and labeling process.

Although the works listed in this section bear some resemblance to the problem studied in this article, only one of the models obtained by the authors can be used in the classification problem presented here [5]. The others have characteristics and objectives different from the problem addressed in this article. In [20] the goal is to classify images of turtles. In [18] the objective is to classify news as news about endangered species or not, which does not include the *Physalia physalis*. In addition to the text used in the training being in English, the characteristics of the text itself, in this case, news (usually long and formal texts), are different from the characteristics of social media texts (usually short and informal texts). In [11] the goal is to classify Twitter texts as being or not about wildlife observations. Furthermore, the authors used only texts in English to train the models, and there were no texts on our species of interest in their training database. In the work of [4] the goal was to investigate the use of deep learning to identify animal trade using pangolin images as a test case. Although the species of interest is different from ours, this work reinforces that the use of deep learning to identify species in data extracted from social media can present good results. In [17], the author proposed a pipeline for text classification using text on bats as a test case. In addition, the text used in the experiments was in English. In summary, as the main differentiator compared to works that propose to identify wildlife in social media data, we trained models with the caption of the posts and also with one of the images of the post. Moreover, none of these works focus on the impact of choosing appropriate labels for training a machine learning model

[4] Twitter is a social network and microblogging service. In July 2023, Twitter was renamed X, however, in this article we will use the original name: Twitter.

[5] https://www.flickr.com/

[6] https://www.facebook.com/

or compare the direct use of the dataset with adaptations that could increase the model performance. In the case of [5], the goal is to classify images of the species. In fact, we used the best architecture reported by the authors in our experiments (Sections 5 and 6).

3. Dataset Construction

This section describes how the data used to train the machine learning models proposed in this work were collected and labeled. It is important to recall that our research interest is in occurrences of *Physalia physalis* on the Brazilian coast. Therefore, the construction of the dataset followed this premise.

In this work, we used data extracted from Instagram. Instagram is a social network developed by Meta. Among its features is the sharing of media (i.e., photos and videos). Users can upload up to 10 media in a single post. They can also write a caption for the post and add its location. In addition, users can add hashtags to the caption. A hashtag can be written with text, emojis, and numbers. Spaces and special characters will not work. Figure 1 shows an example of an Instagram post about a Portuguese man-of-war.



Fig. 1. Example of an Instagram post about a Portuguese man-of-war. Source: Instagram

Instagram posts have some characteristics that should be considered:

- A post can have one or more images or videos. It is common to find posts with different media, for example: a picture of a *Physalia physalis*, a video showing the waves, and a picture of trash, all in the same post;
- As Instagram is an image-focused platform, posts do not always have caption;
- Although it is possible for the user to add location metadata to the post, it is not mandatory and sometimes the location of the occurrence is informed in the caption or comments of the post;
- There are linguistic variations in the caption. While some posts show the correct use of grammar rules, social media text is informal in nature, with many grammatical errors, slang, abbreviations, neologisms, internet jargon, and language mixture. Also,

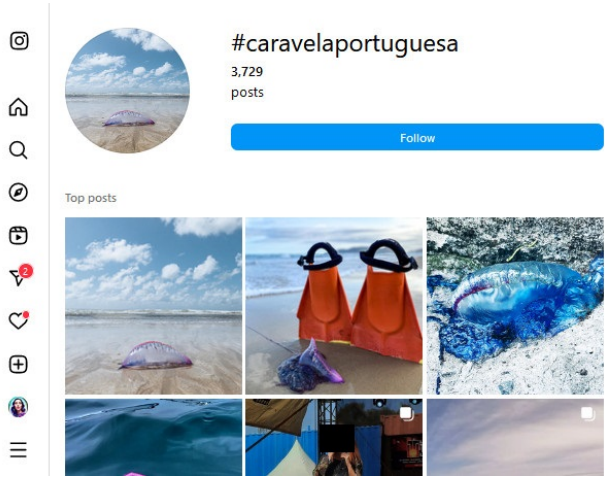


Fig. 2. Screenshot of the #caravelaportugesa page. Source: Instagram

some posts are formed almost exclusively by hashtags and emojis. Furthermore, the use of compound hashtags (e.g., #caravelaportuguesa, #goprobrasil) generates words that do not exist in the lexicon.

When a user has a public account and adds hashtags to a post, the post will be visible on the corresponding Hashtag Page. A Hashtag Page shows a *Top* section where the most popular posts tagged with the hashtag appear. The page also shows a *Recent* section, containing the most recent posts tagged with the hashtag. In this section, the posts appear in the order in which they were posted. Figure 2 shows a screenshot of the page #caravelaportugesa. It is important to know that each hashtag is unique: #cnidario is different from #cnidarios, #aguaviva is different from #águaviva, and so on. This means that one should not expect to find posts that have the hashtag #cnidarios in the #cnidario page, for example.

The first option considered to obtain Instagram data was the use of Instagram's API. However, this API does not return location metadata. So we chose to use the Instaloader library[7]. This library uses scraping to get data from Hashtag Pages and returns data in .json format. More details on the data extraction process were published in [3]. The data of interest for our research are the caption of the post, media, posting date (timestamp), and location (latitude, longitude, city, location name, and country). We did not collect data on users' profiles. Figure 3 illustrates the data of interest for this research.

We chose to use the data obtained through the following hashtags: #aguaviva, #caravelaportuguesa, #cnidarios, #cnidários, #cnidario, #cnidário and #physaliaphysalis.

The #aguaviva was chosen because it is a generic term sometimes used in posts about *Physalia physalis*. The search result for this hashtag returned about 151,000 posts, which include, in addition to *Physalia physalis*, tattoos, people, clothes, other cnidarians, among other things. The #caravelaportuguesa was chosen because it is the popular name of *Physalia physalis* in the Portuguese language and thus more assertive than

[7] https://github.com/instaloader/instaloader

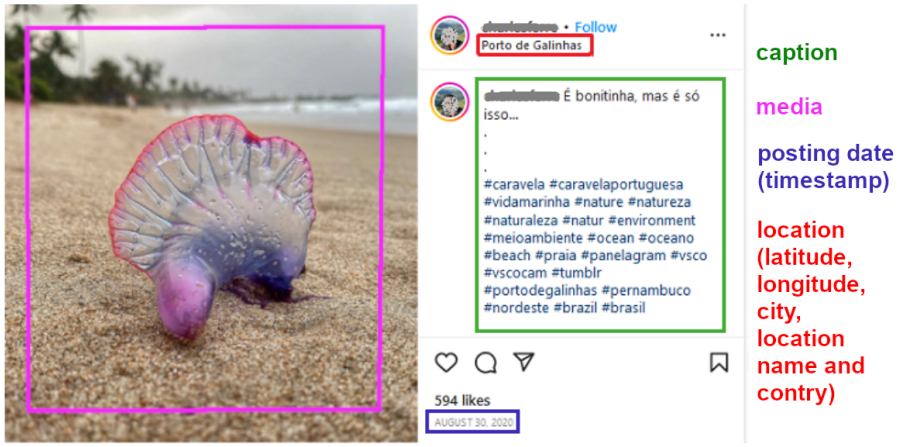


Fig. 3. Illustration the data of interest for this research. In green is the caption of the post, pink is for media, blue is for posting date, and red is for location. Source: adapted from Instagram

#aguaviva. The result of the search for this hashtag returned around 3,300 posts, which included, in addition to *Physalia physalis*, boats, handicrafts, tattoos, clothes, among other things. The #physaliaphysalis, which is the scientific name of the species, was chosen because it has a high potential to obtain positive posts about *Physalia physalis*. A search for this hashtag returned around 2,500 posts. Cnidaria is the name of the phylum to which *Physalia physalis* belongs. Thus the hashtag #cnidarios, and some variations like #cnidário, #cnidario and #cnidários, were chosen because they are common names of this phylum in the Portuguese language and for being terms with medium potential to return positive posts about *Physalia physalis*. A search for #cnidarios returned around 3,400 posts which include *Physalia physalis* as well as other cnidarians such as *Velella velella* and medusa. All searches were performed on August 27, 2022. In total, we collected 6,204 posts dated between 2012 and 2021.

After downloading, the data were saved in spreadsheets to facilitate the annotation process. All data collected were stored in a password-protected file system. For posts that had geolocation (that is, latitude and longitude), the BigDataCloud API[8] was used to obtain additional location data (that is, city, location name, state, country). This data was used as a complement to existing location metadata.

Data were also enriched by identifying the caption language using the Langdetect library[9]. This library has a function that returns a vector indicating which languages were detected in the text along with the probability of the text being written in the language in question. The Portuguese language was assigned to the post whenever it appeared with some probability, even if small. As previously mentioned, among the characteristics of the data extracted from Instagram are the existence of posts using words from more than one language, in addition to the use of compound hashtags that generate words that do not exist in the lexicon. These characteristics make the task of assigning the language for the

[8] https://www.bigdatacloud.com/
[9] https://github.com/Mimino666/langdetect

Table 1. Caption instances that the library did not detect the Portuguese language

Caption	Languages and Probabilities
Sou imortal!	Catalan 99%
#aguaviva Por USER #nemdói TELEFONE para orçamentos #tattoo #inked #inkwork #inkworld #inklife #inkedgirls #tatuagens #art #blackwork #sketch #brunotattoo #omegainkstudio	English 42%, Swedish 28% and Afrikaans 28%

caption difficult. For this reason, the language assigned by the library was reviewed by the computer scientist, with the goal of identifying the presence of terms used in Portuguese that could characterize the caption as written by a speaker of that language. After the review, the number of posts considered in Portuguese increased from 3,440 (according to the library) to 3,849 (after the review).

Table 1 shows 2 caption instances in which the library did not detect the Portuguese language. Figure 4 shows a screenshot of the spreadsheet created with the downloaded and enriched data.

TIMESTAMP	SHORTCODE	LAT	LONG	COUNTRY	CITY	LOCNAME	TEXT	LANGUA	MEDIA1
2016-07-11_12	BHuL2lLAM1	-14	-39		Itacaré	Praia de Jeribucaçu	Para quem não conhece, apresentamos a vocês as Caravellas! Atenção sempre, no mar existe vida e vc é visita, respeite! #VemParaItacaré --- Repost from @[redacted] using	pt	
2016-07-01_18	BHVGJHCj1F	-7	-35	BR	Cabedelo	praia formosa-cabedelo	⚠⚠⚠ATENÇAO BANHISTAS!!!!⚠⚠⚠ Semelhante à água-viva, a CARAVELA-PORTUGUESA e tem um corpo oval, de cor azul, violeta ou vermelha e mede de 10cm a 30 cm. Vamos redobrar os cuidados pois nesse período do ano ficam mas próximas da faixa de areia . "Ela tem veneno com efeito	pt	

Fig. 4. Screenshot of a spreadsheet created with the downloaded and enriched data

4. Data Annotation

The data collected were manually annotated by a computer scientist and an oceanographer. However, the purpose of the annotation differed for each person. The oceanographer was interested in determining the effectiveness of Instagram in providing temporal and spatial information of sightings on the Brazilian coast. On the other hand, the computer scientist was interested in generating an annotated dataset for training a machine learning model in order to automatically classify new Instagram posts, and thus continually monitor the species.

This distinction resulted in a large number of posts classified as legitimate occurrences of the species by one annotator and negative by the other.

The oceanographer used the following criteria to accept a post as a legitimate occurrence of *Physalia physalis*. It had to satisfy three conditions:

– Taxonomic identification: a media that clearly shows a *Physalia physalis*;
– Spatial information: a location on the Brazilian coast;
– Temporal information: include a timestamp associated with the post.

For the computer scientist, on the other hand, the lack of spatial and/or temporal information may not be important for a model that classifies posts based on their text and/or media. Thus, the dataset annotated by the specialist can include noise for training machine learning models.

Figure 5 illustrates the problem. Two posts extracted from Instagram are displayed. Both have an image of *Physalia physalis*, but post A does not have spatial information, while post B does. Following the expert's criteria, only post B is accepted as a legitimate occurrence of *Physalia physalis* on the Brazilian coast, while post A is rejected.

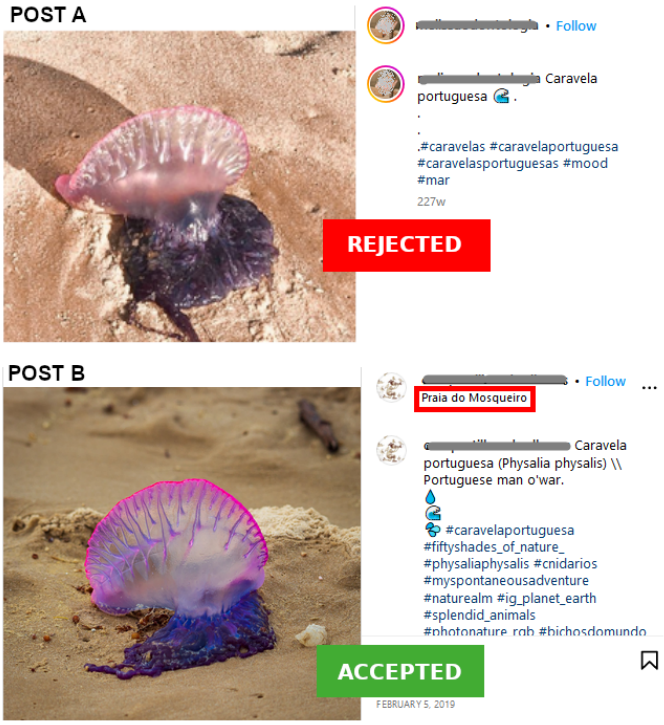


Fig. 5. Illustration of the problem with using the original label. Both posts have an image of *Physalia physalis*, but post A does not have spatial information, while post B does. Following the expert's criteria, only post B is accepted as a legitimate occurrence of *Physalia physalis* on the Brazilian coast, while post A is rejected. Source: adapted from Instagram

To overcome this problem, we decided to adapt the labels, accepting posts that were rejected because of spatial information. However, before assigning a binary label to the entire post, we needed to annotate their taxonomic and spatial information. These new labels weren't given by the oceanographer but were given by the computer scientist with

machine learning in mind. This annotation helped us reach a consensus on the binary labels, solve doubts and ambiguities, increase transparency, and promote the reproducibility of the annotation process.

These taxonomic and spatial classes allowed the adaptation of the binary label in order to reduce noise and generate better classification models, as reported in experiments comparing original and adapted labels (Section 5). In addition, they make it possible to carry out other experiments and studies using the enriched data. As an example, they can be used to train a model to identify accidents with *Physalia physalis* using the taxonomic information.

Spatial Criterion For the spatial criterion, we evaluated whether the post contained spatial information, and in particular, if the location was on the Brazilian coast. Note that a post may have several metadata about location: latitude, longitude, city, location name, and country. In addition, we enriched the metadata using latitude and longitude as we described in Section 3, and sometimes the user informed the location in the caption. There may be a difference between the location obtained by latitude and longitude and the location informed by the user on the metadata (city, location name, and country) or in the caption. The reason for this is that the user can post while being in one location, but referring to something that happened in another location.

Therefore, for the evaluation of spatial information, the following order of precedence was considered: first, the data contained in the caption, second the data from the metadata: city, location name, and country, and lastly location obtained by latitude and longitude. It should be noted that the word or hashtag `portugal` alone was not used as location information, as it often appears as a reference to Portuguese culture, such as the post that talks about a visit to the Museu Paranaense[10]. It says: *"Navegar é preciso #caravelaportuguesa #portugal #brasil #museuparanaense"*[11]. Also, just the state (e.g., #bahia, Ceará), #beach or names of very common beaches (e.g., *Praia Grande*) were not used alone to determine the location.

The spatial information was classified as follows:

- COAST-TEXT: The caption contains some information that allows identifying the location as being on the Brazilian coast (even if there is metadata indicating otherwise).
- COAST-GEO: The post has metadata that allows identifying the location as being on the Brazilian coast.
- COUNTRYSIDE-TEXT: The caption contains some information that indicates that the post did not take place on the Brazilian coast, but still the location is in Brazil.
- COUNTRYSIDE-GEO: The post has metadata indicating that the post did not take place on the Brazilian coast, but still the location is in Brazil.
- FOREIGNER: The post has information in the caption or metadata that indicates that the post did not take place in Brazil.
- NOTHING: There is no information that allows the post location to be identified.

In some cases the location has been identified as being in a coastal city, but it is clear that it is not on the beach, as in the examples: *"Aquário Marinho do Rio de Janeiro"* and

[10] Museu Paranaense is a museum located in the Brazilian city of Curitiba

[11] Translated as: "Sailing is necessary #caravelaportuguesa #portugal #brasil #museuparanaense"

"Centro Acadêmico de Vitória - CAV - UFPE"[12]. These posts are classified as COUNTRYSIDE-TEXT when the information is part of the caption and as COUNTRYSIDE-GEO when the information is part of the metadata.

Taxonomic Criterion For the taxonomic criterion, we evaluated the media of the posts, especially if the images are photos of *Physalia physalis*. However, the existence of a *Physalia physalis* picture does not always mean the post will be accepted as a legitimate occurrence. It is important to recall that a single post can have up to 10 media, so each media can have a different classification. The media of the posts were classified as follows.

- REALISTIC: The media is a realistic picture of *Physalia physalis*, either on the beach sand or in the sea. In addition, when a post contains several images, they appear to be from the same occurrence.
- CLOSE: The media is a realistic picture of *Physalia physalis*, but it is a close-up image, showing only parts of *Physalia physalis*.
- DISPLACED: The media is a realistic picture of *Physalia physalis*, but displaced from its habitat. For example, it is in an aquarium or inside a plastic bag.
- EDITED: The media is a realistic picture of *Physalia physalis* but with minor edits such as the inclusion of authorship, text, or frame, but that do not disturb the visualization of *Physalia physalis*.
- COLLECTION: The images in the post are a collection of images from *Physalia physalis*, but it is possible to notice that they are not from the same occurrence and are possibly images taken from the internet.
- ART: The image is a realistic representation of *Physalia physalis*, such as: compositions, drawings, tattoos. It could be a realistic image of *Physalia physalis* with edits that reveal this is not a sighting of the species.
- ACCIDENT: The media is a picture of a body part "burned" by a *Physalia physalis*.
- CNIDARIA: The media is a picture of another cnidarian, for example: *Velella velella*.
- VIDEO: The media is a video and therefore was not evaluated for this criterion.
- NOTHING: The media cannot be sorted in any other way described above.

Binary Label Based on the spatial and taxonomic annotations, the posts were labeled as ACCEPTED or REJECTED as legitimate occurrences of *Physalia physalis* on the Brazilian coast. For Instagram posts, the temporal criterion is always satisfied, because every post has an associated timestamp. The given label is also associated with a justification as described next.

To be *accepted* as a legitimate occurrence of *Physalia physalis* on the Brazilian coast, the post must meet taxonomic and spatial criteria. It means that: the post must have at least one image classified as REALISTIC, COLLECTION, EDITED, DISPLACED, or CLOSE (even if the post has other media that do not fit these five classes), and have the spatial information classified as COAST-GEO or COAST-TEXT.

One post that does not meet one or more criteria is considered *rejected* as a legitimate occurrence of *Physalia physalis* on the Brazilian coast. The rejection can be justified as:

[12] Translated as: "Marine Aquarium of Rio de Janeiro" and "Vitoria Academic Center - CAV - Federal University of Pernambuco"

- BECAUSE OF THE MEDIA: The post has spatial information classified as COAST-GEO or COAST-TEXT, but no media were classified as REALISTIC, COLLECTION, EDITED, DISPLACED, or CLOSE.
- BECAUSE OF THE LOCATION: The post has at least one media classified as REALISTIC, COLLECTION, EDITED, DISPLACED, or CLOSE, but the spatial information is classified as COUNTRYSIDE-TEXT, COUNTRYSIDE-GEO, FOREIGNER or NOTHING.
- BECAUSE OF THE MEDIA AND LOCATION: The post does not meet both criteria.

4.1. Distribution by Label

Table 2 shows the number of accepted and rejected posts per hashtag. To simplify the analysis, we combined all the data from the hashtags: #cnidario, #cnidarios, #cnidario and #cnidarios. So from now on, when we write #cnidario, we are considering all posts with these hashtags.

In general, the number of positive posts for the occurrence of *Physalia physalis* on the Brazilian coast was small, mainly for the hashtags: #aguaviva, #cnidarios, and #physaliaphysalis. We expected to find more positive posts in the search for #physaliaphysalis, but only 8% of the collected posts were accepted as legitimate occurrences of *Physalia physalis* for this hashtag. Despite the small number of positive occurrences found with #aguaviva (1%), the period surveyed was also the shortest (175 days), which makes room to invest more effort in the search for the species in posts that have this hashtag. The search for occurrences using the hashtag #caravelaportuguesa had the best result, with 422 posts accepted by the specialist.

The table also shows that the number of rejected posts is much larger than the number of accepted ones.

Table 2. Number of posts per Hashtag and Label. The percentage is based on the number of posts by hashtag

Hashtag	Label	No. Posts	%
#aguaviva	ACCEPTED	23	1%
	REJECTED	1,657	99%
#caravelaportuguesa	ACCEPTED	422	24%
	REJECTED	1,364	76%
#cnidario	ACCEPTED	77	3%
	REJECTED	2,472	97%
#physaliaphysalis	ACCEPTED	15	8%
	REJECTED	174	92%

5. Experiments with Adapted Labels

As discussed earlier, posts rejected because of location may represent noise for training machine learning models. In this section, we investigate whether there is any impact on

the trained model if we consider as accepted posts rejected because of spatial information. We performed experiments with the caption and one of the images of the posts and built three annotated datasets:

- `original`: the one annotated with the oceanographer criteria.
- `adapted-lack-location`: considers as ACCEPTED posts rejected only for lack of spatial information (i.e., spatial information equals NOTHING).
- `adapted-not-in-coast`: considers as ACCEPTED posts rejected because of spatial information (i.e., spatial information equals NOTHING, COUNTRYSIDE-TEXT, COUNTRYSIDE-GEO OR FOREIGNER).

For the training, validation, and testing of the machine learning models we first filtered the dataset as follows. We deleted posts that have the following characteristics: empty text (12 posts); duplicate posts identified by having the same ID (164 posts); location outside Brazil (1,991 posts); and with video only (643 posts). The latter refers to posts that do not have images, but just videos. We removed them because we chose to do standard image classification. Moreover, we kept posts identified as Portuguese language and those with only emojis in the caption because emojis can be translated into Portuguese. After applying the filters, 2,610 posts remained in the dataset, which were used in the experiments described next. Table 3 shows the number of remaining posts in the original and adapted datasets.

Table 3. Number of remaining posts per Label

Label	original		adapted-lack-location		adapted-not-in-coast	
	No. Posts	%	No. Posts	%	No. Posts	%
ACCEPTED	470	18%	648	25%	658	25%
REJECTED	2,140	82%	1,962	75%	1,952	75%

Table 4 shows the number of rejected posts per reason for rejection. There were 188 posts rejected because of location, that were considered accepted in the adapted datasets.

Table 4. Number of remaining rejected posts per reason for rejection

Reason for Rejection	No. Posts
LOCATION	188
MEDIA AND LOCATION	1,549
MEDIA	403
TOTAL	2,140

Table 5 details these 188 posts per type of spatial information. There are 178 posts rejected because of lack of location (NOTHING). They are considered accepted in the adapted-lack-location dataset, resulting in 648 (470+178) accepted posts in this dataset. There are additional 10 posts rejected because they have location information but not in

the coast (COUNTRYSIDE-GEO and COUNTRYSIDE-TEXT). These posts are considered accepted in the adapted-not-in-coast dataset, in addition to those labeled as NOTHING. Thus, the number of accepted posts in the adapted-not-in-coast dataset is 658.

Table 5. Number of remaining posts rejected because of location per type of spatial information

Spatial info. class	No. Posts
NOTHING	178
COUNTRYSIDE-GEO	9
COUNTRYSIDE-TEXT	1
TOTAL	188

Comparing the original and adapted labels, the number of accepted posts in both adapted datasets is not much bigger than in the original dataset (see Table 3). Although the difference is small, it already affects the quality of the machine learning models obtained, as we report in Sections 5.1 and 5.2.

To evaluate the machine learning models trained in this work, the metrics precision, recall and F1 Score were used. They were chosen because they are indicated to evaluate the resulting models in problems whose data are imbalanced and when there is a need to keep the number of false positives low.

Furthermore, considering that in the future the model can be integrated into an automated ETL process of a database on occurrences of *Physalia physalis* on the Brazilian coast, the ideal is to minimize the number of false positives. This prevents false posts from being registered in the database, even with the risk that legitimate posts are discarded, thus guaranteeing credibility and trust in the data included in the database. This means that precision is an important metric for our study, and it will be considered as the main measure to compare models.

We divided the dataset into 30% (test) and 70% (development). Then, we applied cross-validation across 5 folds, where the development set was divided into 70% (train) and 30% (validation). The result was a test set with 783 samples, a train set with 1279 samples, and a validation set with 548 samples. Furthermore, we used class weighting to handle data imbalance.

The experiments were performed using Python and libraries for NLP and image classification, such as NLTK [2], Scikit-learn [25], TensorFlow [1] and Keras [7].

5.1. Textual Analysis

We trained models with the posts' captions, using original and adapted labels. In the experiments, we trained a classic machine learning model available in the Scikit-learn library [25]: Logistic Regression (LR). We also considered a state-of-the-art deep learning method for NLP tasks: Multilingual BERT (mBERT) [8], which is available in the Tensorflow Hub[13]. It is a pre-trained model based on the weights of the original BERT [9], available only in the *base* size. It was trained in 104 languages, including Portuguese.

[13] https://tfhub.dev/tensorflow/bert_multi_cased_L-12_H-768_A-12/4

mBERT was refined with an output layer adapted to our problem (with a sigmoid function), and a dropout layer (with 10% of probability). In addition, we unfroze the model and retrained it with our data. We used as hyperparameters: 50 epochs with early stopping based on the loss calculated in the validation set, batch size of 16, AdamW optimizer, learning rate of 3e-5 with warmup over the first 10% steps and linear decay, and a sentence length of 128 tokens, since this sentence length is the default value from the mBERT preprocessing model[14] provided by TensorFlow Hub. For other hyperparameters, we used the same ones as used by [9].

For training with LR we kept its default parameters except the parameter `class_weight` which was set to `'balanced'`. This option automatically adjusts the weights inversely proportional to the class frequencies in the input data. It is a strategy to deal with imbalanced data. As vectorization methods, TF-IDF and mBERT without fine-tuning were used. For TF-IDF, we kept the default values of the method. By default, this method converts data to lowercase, and ignores tokens less than 2 characters long, emojis, punctuation, and signs, except underline. In the end, the method normalizes the resulting TF-IDF vectors by the Euclidean norm. For mBERT, we used the pooled-output, the contextual embedding that represents the sentence.

The results obtained for classifying posts according to their captions are presented in Table 6.

Table 6. Results of Experiments with Posts' Caption. Showing: Label, Precision, Recall and F1 Score with their respective standard deviation

Label	TF-IDF + LR			mBERT + LR			mBERT		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
original	0.667 (0.022)	0.831 (0.040)	0.740 (0.021)	0.478 (0.033)	0.760 (0.044)	0.585 (0.015)	0.708 (0.040)	0.740 (0.080)	0.722 (0.044)
adapted-lack-location	0.791 (0.009)	0.899 (0.027)	0.842 (0.015)	0.590 (0.020)	0.800 (0.022)	0.679 (0.019)	0.823 (0.041)	0.886 (0.046)	0.852 (0.014)
adapted-not-in-coast	0.805 (0.024)	0.893 (0.032)	0.846 (0.026)	0.618 (0.018)	0.787 (0.036)	0.692 (0.019)	0.841 (0.052)	0.876 (0.048)	0.856 (0.022)

It is possible to observe that there was an increase in all metrics when we compare the results of the models trained with adapted labels with the results of the models trained with original labels.

Another point is that models trained with the label adapted-lack-location showed a small advantage in the recall, around 1% greater than models trained with the label adapted-not-in-coast. However, models trained with adapted-not-in-coast labels showed a small advantage in precision compared to models trained with adapted-lack-location.

5.2. Image Analysis

We trained CNNs with the posts' images, using original and adapted labels. As the posts can have more than one image, we randomly selected one image per post to perform the

[14] https://tfhub.dev/tensorflow/bert_multi_cased_preprocess/3

training. In particular, for positive posts, we selected one image that met the taxonomic criterion.

We trained the CNNs by transferring the learning from a ResNet50 [16] pre-trained with Imagenet [27] and adapted the last layer to our problem. We chose the same architecture described as the best model in [5].

ResNet50 was adapted by adding: an average pooling layer, a flattening layer, a fully connected layer with ReLu, a dropout regularization layer with a percentage of 30%, and an output layer with one neuron with sigmoid. To increase the amount of training images, we used data augmentation. We have applied random rotation, random zoom, and random horizontal flip of the images. Note that data augmentation is only applied to the development set. All images have been resized to a 224x224 size. This data preparation is actually done by the neural network, through the addition of a preprocessing layer. These preprocesses and hyperparameters were chosen based on the work of [5]. We also used the following hyperparameters: 50 epochs with early stopping based on the loss calculated in the validation set, batch size of 32, and Adam optimizer with a learning rate of 0.001.

The results for classifying posts based on their images are presented in Table 7.

Table 7. Results of Experiments with Posts' Image. Showing: Label, Precision, Recall, and F1 Score with their respective standard deviation

Label	Precision	Recall	F1
original	0.844 (0.049)	0.938 (0.066)	0.889 (0.056)
adapted-lack-location	0.965 (0.024)	0.957 (0.059)	0.961 (0.042)
adapted-not-in-coast	0.967 (0.049)	0.986 (0.020)	0.976 (0.035)

As in the experiments with captions, the experiments with images showed that the adapted labels produced better classification models than the original label, with the adapted-not-in-coast label achieving a 9% F1 Score increase compared to the original label.

5.3. Discussion

The experiments for both text and image show that the adapted labels produced better classification models than the original labels since there was an increase of all metrics when we compare the results of the models trained with adapted labels with the results of the models trained with original labels.

Despite the small increase in the number of ACCEPTED posts when we adapted the labels (18% to 25%), as presented in Table 3, the results with the adapted labels are significantly higher than results with the original labels.

6. Simulating the Production Flow

To find out if the models are really applicable to generate a base on occurrences of *Physalia physalis* in a real scenario. In this section, we trained models using the entire development set and evaluated them on two different test sets: the `full test set` (as

described at the beginning of Section 5) and a `partial test set`. Using this partial test set, we intended to simulate the flow in production, in which, before being evaluated by the model, the post is evaluated for location data. When the post meets the spatial criteria, it is evaluated by the model. If not, it is rejected before being evaluated by the model. In practice, we create a partial test set with the posts `accepted` and `rejected because of media`, resulting in 257 posts. Figure 6 illustrates the production scenario.

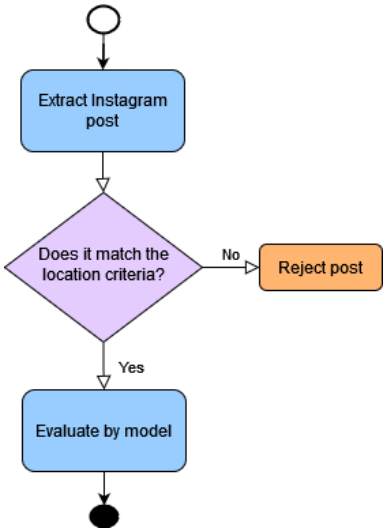


Fig. 6. Illustration of the production scenario. Before being evaluated by the model, the post is evaluated for location data. When the post meets the spatial criteria, the post is evaluated by the model. If not, it is rejected before being evaluated by the model. Source: the Author.

6.1. Textual Analysis

In this part of the experiments, we chose to train only mBERT, since it presented the best precision and F1 Score when trained with adapted labels. Five training rounds with different random states were performed. We also applied hyperparameter optimization, where we obtained the best results with 15 epochs and a sentence length of 344. The results are presented in Table 8.

When we observe the results presented in Table 8, it is possible to notice an increase in all metrics when we evaluate the models by simulating the flow in production.

Error Analysis We compared the performance of the best-performing classifiers. As we executed 5 runs for each mBERT configuration, we present here the results of the models that achieved the best precision among the 5 runs. As we mentioned at the beginning of Section 5, the ideal for our application is to minimize the number of false positives.

Table 8. Results of mBERT trained with entire development set and evaluated with the partial and full test sets. They report the mean performance over 5 runs with the standard deviation

	Full Test Set			Partial Test Set		
Label	Precision	Recall	F1	Precision	Recall	F1
adapted-lack-location	0.868 (0.012)	0.809 (0.051)	0.837 (0.024)	0.925 (0.008)	0.840 (0.050)	0.880 (0.027)
adapted-not-in-coast	0.860 (0.018)	0.858 (0.025)	0.859 (0.004)	0.926 (0.013)	0.879 (0.027)	0.902 (0.015)

In Figure 7 we present the confusion matrices achieved by the mBERT models evaluated with the full test set, while in Figure 8 we present the confusion matrices achieved by the mBERT models evaluated with the partial test set.

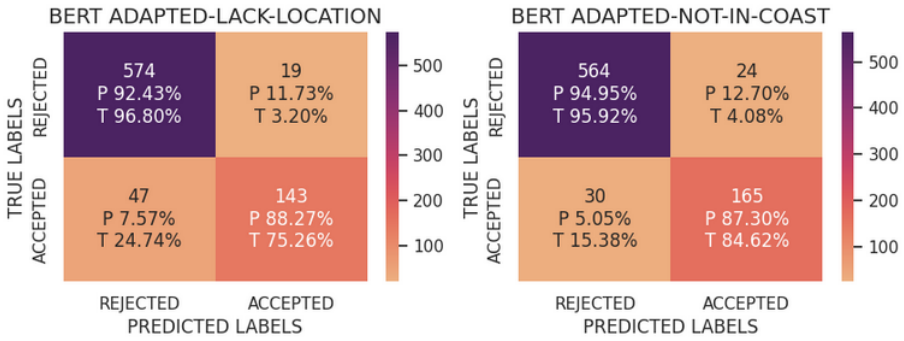


Fig. 7. Confusion Matrices of mBERT models evaluated with the full test set. BERT ADAPTED-LACK-LOCATION shows the confusion matrix of the model trained with adapted-lack-location and achieved the best precision among the 5 runs, this model has an F1 Score of 81%, while BERT ADAPTED-NOT-IN-COAST shows the confusion matrix of the model trained with adapted-not-in-coast and achieved the best precision among the 5 runs, this model has an F1 Score of 86%. Beyond the classifier's errors and successes in absolute numbers, they show normalized numbers over the predicted label (P), and normalized numbers over the true label (T)

It is possible to observe that all models have more difficulty in recognizing positive examples. Note the normalized numbers over the true label (T), the percentage of false negatives is greater than the percentage of false positives.

There is a decrease in false positives in BERT ADAPTED-LACK-LOCATION when evaluated with the full test set. On the other hand, when the models are evaluated with the partial test set, beyond the general increase of all metrics, the model trained with ADAPTED-NOT-IN-COAST presented a decrease of false positive and false negative, and also an increase of precision, resulting in an F1 Score of 91%.

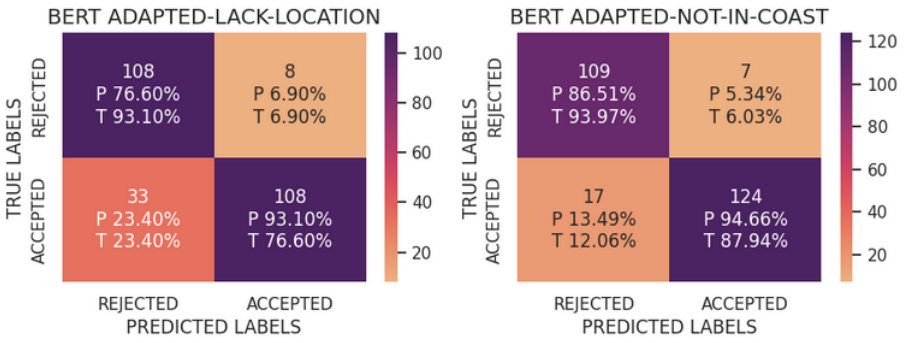


Fig. 8. Confusion Matrices of mBERT models evaluated with the partial test set. BERT ADAPTED-LACK-LOCATION shows the confusion matrix of the model trained with adapted-lack-location and achieved the best precision among the 5 runs, this model has an F1 Score of 84%, while BERT ADAPTED-NOT-IN-COAST shows the confusion matrix of the model trained with adapted-not-in-coast and achieved the best precision among the 5 runs, this model has an F1 Score of 91%. Beyond the classifier's errors and successes in absolute numbers, they show normalized numbers over the predicted label (P), and normalized numbers over the true label (T)

6.2. Image Analysis

As in the experiments with captions, we also trained models using the entire development set and evaluated them on two different test sets: the `full test set` and the `partial test set`. Five training rounds with different random states were performed. The hyperparameters were optimized and 2 techniques were experimented to deal with data imbalance: undersampling and class weighting. We obtained the best results with undersampling, a learning rate of 1e-5, and 512 neurons with ReLu. The results are presented in Table 9.

Table 9. Results of ResNet50 trained with entire development set and evaluated with the partial and full test sets. Showing Label, Precision, Recall and F1 Score. We show the mean performance over 5 runs with the standard deviation

	Full Test Set			Partial Test Set		
Label	Precision	Recall	F1	Precision	Recall	F1
adapted-not-in-coast	0.933 (0.026)	0.895 (0.052)	0.913 (0.019)	0.978 (0.015)	0.926 (0.045)	0.951 (0.020)
adapted-lack-location	0.922 (0.023)	0.866 (0.058)	0.892 (0.021)	0.985 (0.008)	0.896 (0.050)	0.938 (0.025)

Observing the results presented in Table 9, it is possible to notice an increase in all metrics when we evaluate the models by simulating the flow in production.

Error Analysis We compared the performance of the best-performing classifiers. As we performed 5 runs for each configuration, we present here the results of the models

that achieved the best precision among the 5 runs. As we mentioned at the beginning of Section 5, the ideal for our application is to minimize the number of false positives.

In Figure 9 we present the confusion matrices achieved by the ResNet50 models evaluated with the full test set, while in Figure 10 we present the confusion matrices achieved by the ResNet50 models evaluated with the partial test set.

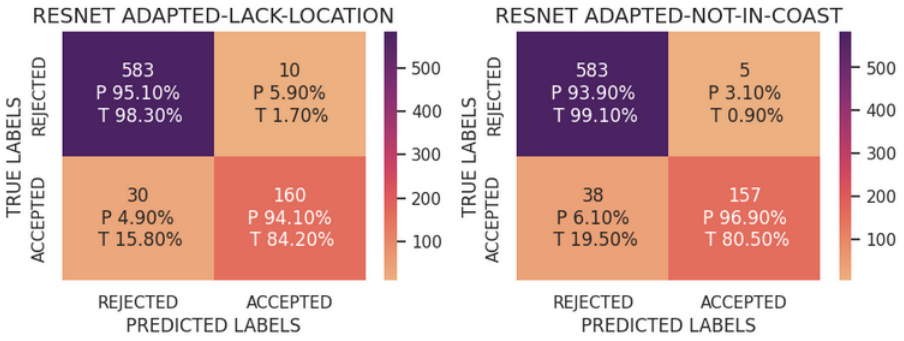


Fig. 9. Confusion Matrices of ResNet50 models evaluated with the full test set. RESNET ADAPTED-LACK-LOCATION shows the confusion matrix of the model trained with adapted-lack-location and achieved the best precision among the 5 runs, this model has an F1 Score of 89%, while RESNET ADAPTED-NOT-IN-COAST shows the confusion matrix of the model trained with adapted-not-in-coast and achieved the best precision among the 5 runs, this model has an F1 Score of 88%. Beyond the classifier's errors and successes in absolute numbers, they show normalized numbers over the predicted label (P), and normalized numbers over the true label (T)

Like in the models trained with captions. It is possible to observe that all models have more difficulty in recognizing positive examples. Note the normalized numbers over the true label (T), the percentage of false negatives is greater than the percentage of false positives.

There is a decrease in false positives in RESNET ADAPTED-NOT-IN-COAST when evaluated with the full test set. On the other hand, when the models are evaluated with the partial test set, beyond the general increase of all metrics, both models presented the same number of false positives and precision. The model trained with ADAPTED-LACK-LOCATION presented a decrease of false negatives and also an increase in recall, resulting in an F1 Score of 94%.

6.3. Discussion

The model trained with text that achieved the best result was a multilingual BERT model trained with the adapted-not-in-coast label, which achieved a precision of 95% and an F1 Score of 91%, when evaluated with the partial test set (see Figure 8). The model trained with images that obtained the best result was the model trained with adapted-lack-location, which achieved a precision of 99% and an F1-Score of 94% when evaluated with the partial test set (see Figure 10).

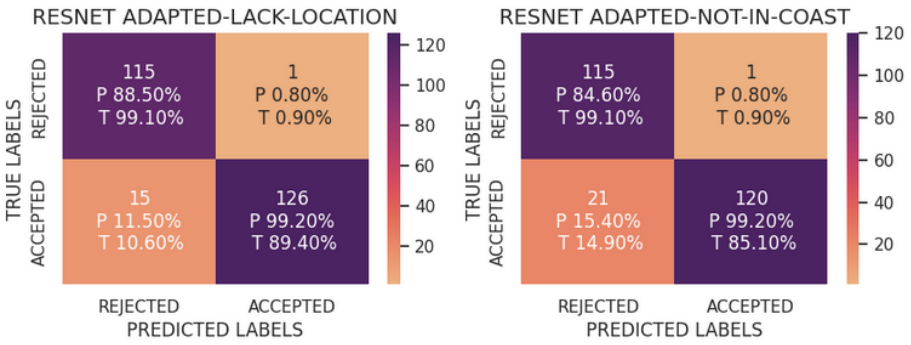


Fig. 10. Confusion Matrices of ResNet50 models evaluated with the partial test set. RESNET ADAPTED-LACK-LOCATION shows the confusion matrix of the model trained with adapted-lack-location and achieved the best precision among the 5 runs, this model has an F1 Score of 94%, while RESNET ADAPTED-NOT-IN-COAST shows the confusion matrix of the model trained with adapted-not-in-coast and achieved the best precision among the 5 runs, this model has an F1 Score of 92%. Beyond the classifier's errors and successes in absolute numbers, they show normalized numbers over the predicted label (P), and normalized numbers over the true label (T)

These experiments also show that between the two best models, the one trained with images outperformed the model trained with text in 4% of precision and 3% of F1 Score. Given the nature of Instagram, which is based primarily on images and video, this is an expected result. However, the good results obtained with text, show that it can also play an important role in the classification process.

Moreover, the results demonstrate that the spatial information had indeed an impact on the results, being actually a source of noise for the problem investigated. It reinforces that adapting the labels was a good choice. Moreover, results presented in this Section show that filtering out posts that do not meet the spatial criteria before evaluating them by the classifier can further improve all metrics.

7. Conclusion

In this article, we explored the problem of identifying legitimate occurrences of *Physalia physalis* in posts extracted from Instagram using a classification task. Classifiers were trained using only the text of the post and only one of the images of the post. We also presented an experience of constructing a dataset on the cnidarian *Physalia physalis* with posts extracted from Instagram. In particular, we showed that even for a binary classification problem, to distinguish between legitimate and false sightings of the species, the design of appropriate labels can affect the quality of the machine learning model trained with the dataset.

Among the models experimented with the text are Logistic Regression and Multilingual BERT. The latter was also experimented as a feature extractor to feed the Logistic Regression model and also retrained with our data and used as a classifier. TF-IDF was also used in conjunction with Logistic Regression.

ResNet50 pre-trained with ImageNet was chosen for image experiments. They were carried out with different approaches to deal with imbalanced data, in addition to retraining the network with our data and optimizing the hyperparameters.

The best models were evaluated on a test set that simulates the flow in production, in which before being evaluated by the model the post is evaluated for location data. It was possible to observe an increase in all metrics, achieving a precision of 95% for text and 99% for images, indicating that these models can be used as part of an automated ETL process of a database on occurrences of *Physalia physalis* on the Brazilian coast from data extracted from Instagram.

One of the contributions of this work is a comparison between the use and performance of different modals (i.e., text and image) as resources for training machine learning models, in the context of conservation science and social media data. To the best of our knowledge, there is no other work using text and image for species recognition in data extracted from social media. Furthermore, there is a small number of works that explore the text for tasks related to conservation science, as already mentioned by [11]. In fact, the use of images and video are more common in this context. Our experiments reinforce the importance of images in these applications, showing that for our target species models trained with images outperformed models trained with text by 4% of precision.

Another contribution of this article was to show the importance of knowledge on the part of the computer scientist of which criteria were used to label the data, since the unquestionable use of the label assigned by the specialist can be a source of noise. This is especially true for works involving species occurrence, where spatial and temporal data are essential but may not be present in all samples. Knowing how the data was labeled can allow the labels to be adapted and therefore produce better results, as demonstrated in this work.

The dataset that was constructed and labeled manually by the oceanographer and the computer scientist, can be extended with new data through the application of the final models. Furthermore, taxonomic and spatial information labels can be used to refine the model itself or used for training new models. Examples of refinements include: recognizing the spatial information contained in the caption and differentiating sightings from accidents using taxonomic information.

As a final contribution, as the best models are neural networks, they can be adapted to new problems, taking advantage of the transfer learning paradigm, which makes it possible to automate the identification of other species.

This work is part of a project that has the goal of monitoring *Physalia physalis* on the Brazilian coast. There are many interesting directions that this work can be extended to contribute to this objective. Among them, we are planning to train multimodal machine learning in order to produce a classifier that takes into consideration both text and images in the same model. We also plan to explore preprocessing techniques for the text and train classification models to automatically extract data from other sources such as newspapers, and other social media platforms.

Other future lines of investigation include: experiments with other language models, such as BERTimbau [28] and XML-RoBERTa[15]; interpretation of decisions made by the models by applying methods such as Feature Visualization and Pixel Attribution [21];

[15] https://huggingface.co/xlm-roberta-large

experiments with new techniques such Vision Transformer (ViT) [10] to image classification.

Acknowledgments. This work was partially supported by CNPq (Process 407644/2021-0), and by CAPES/PrInt-UFPR.

References

1. Abadi, M., et al.: TensorFlow: Large-scale machine learning on heterogeneous systems (2015), https://www.tensorflow.org/
2. Bird, S., Klein, E., Loper, E.: Natural Language Processing with Python. O'Reilly Media Inc. (2009), https://www.nltk.org/
3. Camargo, L., Rocha, H., Nascimento, L., Hara, C.: Coleta de dados do instagram sobre ocorrências de caravelas-portuguesas na costa brasileira. In: Anais da XVIII Escola Regional de Banco de Dados. pp. 51–59. SBC, Porto Alegre, RS, Brasil (2023)
4. Cardoso, A.S., Bryukhova, S., Renna, F., Reino, L., Xu, C., Xiao, Z., Correia, R., Di Minin, E., Ribeiro, J., Vaz, A.S.: Detecting wildlife trafficking in images from online platforms: A test case using deep learning with pangolin images. Biological Conservation 279, 109905 (2023)
5. Carneiro, A., Nascimento, L.S., Noernberg, M.A., Hara, C.S., Pozo, A.T.R.: Social media image classification for jellyfish monitoring. Aquatic Ecology 58, 3–15 (2024)
6. Cavalcante, M.M.E., Rodrigues, Z.M.R., Hauser-Davis, R.A., Siciliano, S., Haddad Júnior, V., Nunes, J.L.S.: Health-risk assessment of portuguese man-of-war (physalia physalis) envenomations on urban beaches in são luís city, in the state of maranhão, brazil. Revista da Sociedade Brasileira de Medicina Tropical 53 (2020)
7. Chollet, F., et al.: Keras. https://keras.io (2015)
8. Devlin, J.: Multilingual bert. https://github.com/google-research/bert/blob/master/multilingual.md (2019), acessado em 10/03/2022
9. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: "BERT: Pre-training of deep bidirectional transformers for language understanding". In: Burstein, J., Doran, C., Solorio, T. (eds.) Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019), https://aclanthology.org/N19-1423
10. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale (2021)
11. Edwards, T., Jones, C.B., Corcoran, P.: Identifying wildlife observations on twitter. Ecological Informatics 67, 101500 (2022)
12. Edwards, T., Jones, C.B., Perkins, S.E., Corcoran, P.: Passive citizen science: The role of social media in wildlife observations. PLOS ONE 16(8), e0255416 (08 2021)
13. ElQadi, M.M., Lesiv, M., Dyer, A.G., Dorin, A.: Computer vision-enhanced selection of geotagged photos on social network sites for land cover classification. Environmental Modelling & Software 128, 104696 (2020)
14. Foglio, M.: Animal Wildlife Population Estimation Using Social Media Images Collections. Master's thesis, University of Illinois, Chicago, Illinois, USA (2019)
15. Ghermandi, A., Sinclair, M.: Passive crowdsourcing of social media in environmental research: A systematic map. Global Environmental Change 55, 36–47 (2019)
16. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770–778 (June 2016)

17. Hunter, S.B., Mathews, F., Weeds, J.: Using hierarchical text classification to investigate the utility of machine learning in automating online analyses of wildlife exploitation. Ecological Informatics 75, 102076 (2023)
18. Kulkarni, R., Di Minin, E.: Automated retrieval of information on threatened species from online sources using machine learning. Methods in Ecology and Evolution 12(7), 1226–1239 (2021)
19. Leitão, A.T.T.S., de O Alves, M.D., dos Santos, J.C.P., Bezerra, B.: Instagram as a data source for sea turtle surveys in shipwrecks in brazil. Animal Conservation 25(6), 736–747 (2022)
20. Mazars-Simon, A.E.: The Wild in Live Project: A Human/Algorithm learning network to help citizen science in wildlife conservation. Master's thesis, Universidade de Coimbra (2019)
21. Molnar, C.: Interpretable Machine Learning. Independently published, 2 edn. (2022), https://christophm.github.io/interpretable-ml-book
22. Morais, P., Afonso, L., Dias, E.: Harnessing the Power of Social Media to Obtain Biodiversity Data About Cetaceans in a Poorly Monitored Area. Frontiers in Marine Science 8 (2021)
23. do Nascimento, L.S.: Monitoring jellyfish population by social media. Tech. rep., Universidade Federal do Paraná (2020), technical report, Pós-Graduação em Sistemas Costeiros e Oceânicos
24. Nascimento, L.S., Hara, C.S., Jr., M.N., Noernberg, M.: Instagram como fonte de dados alternativa no monitoramento da #caravelaportuguesa (physalia phisalis, cnidaria). In: Livro de Memórias do IV SUSTENTARE e VII WIPIS: Workshop internancional de Sustentabilidade, Indicadores e Gestão de Recursos Hídricos (2022)
25. Pedregosa, F., et al.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12, 2825–2830 (2011)
26. Rocha, H.F., Nascimento, L.S., Camargo, L., Noernberg, M., Hara, C.S.: Labeling portuguese man-of-war posts collected from instagram. In: Abelló, A., Vassiliadis, P., Romero, O., Wrembel, R., Bugiotti, F., Gamper, J., Vargas Solar, G., Zumpano, E. (eds.) New Trends in Database and Information Systems. pp. 369–381. Springer Nature Switzerland, Cham (2023)
27. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. International journal of computer vision 115, 211–252 (2015)
28. Souza, F., Nogueira, R., Lotufo, R.: BERTimbau: pretrained BERT models for Brazilian Portuguese. In: Cerri, R., Prati, R.C. (eds.) Intelligent Systems. pp. 403–417. Springer International Publishing, Cham (2020)
29. Sullivan, M., Robinson, S., Littnan, C.: Social media as a data resource for #monkseal conservation. PLoS ONE 14(10) (2019)

Heloisa Fernanda Rocha is a Software Engineer with over 15 years of experience. She holds a master's degree in Informatics from the Federal University of Paraná in 2023. She received a postgraduate degree in Database Administration from Ponta Grossa State University in 2004 and Software Engineering from the Pontifical Catholic University of Paraná in 2015.

Leonardo S. Camargo is a Major in Science Computer graduation student at Universidade Federal do Parana (UFPR) in Brazil and an intern at Telefónica Brasil S.A. He has researched social media data mining and presented the article "Coleta de Dados do Instagram sobre Ocorrências de Caravelas-Portuguesas na Costa Brasileira" at Encontro Regional de Banco de Dados (ERBD,2022). His interests reside in Data Visualization, Data Mining and Data Science.

Maurício Almeida Noernberg recived the PhD degree in environmental geology from the Federal University of Paraná in 2001. He is currently a Full Professor at Center for

Marine Studies and the Head of Coastal Oceanography and Remote Sensing Lab. at the Federal University of Paraná. His research interests include land-sea interactions, coastal circulation, environmental monitoring, numerical modeling, remote sensing and image processing.

Aurora Pozo is currently a full Professor of Computer Science at Universidade Federal do Paraná, where she heads the interdisciplinary Evolutionary Computation Research Group. Her scientific contributions are in the area of computational intelligence, mainly focused on machine learning, and particularly in evolutionary computation and learning (using genetic programming, particle swarm optimization, and learning classifier systems), multi-objective optimization, and evolutionary deep learning. In addition to technical contributions in the area of computational intelligence, Aurora works to consolidate the area in Brazil, actively participating in events related to the area (BRACIS, ENIAC, CTDIAC). The bidder has coordinated CNPq Universal and Fundação Araucária projects in the area of bioinspired computing and applications. The projects have the collaboration of professors from different National Universities. Internationally, she coordinated projects which allowed fruitful cooperation with the University of the Vasco Country/Spain. In addition to these activities, during this period, she has been dedicated to the qualification of human resources, providing ongoing supervision to both master's and doctoral candidates. Furthermore, she has contributed to numerous high-quality articles published in esteemed journals and conferences, cementing her presence in the academic and research spheres.

Carmem S. Hara is a Full Professor at Universidade Federal do Parana (UFPR) in Brazil, head of the Information and Data Management Group (GDAI), and member of the Wireless and Advanced Networks Group (NR2). Her main research interests include management of complex and graph data models. She holds a PhD degree in Computer and Information Sciences from the University of Pennsylvania, USA (2014). From 2015 to 2019, she served as a member of the Database Special Council of the Brazilian Computer Society (CEBD-SBC), and is currently serving the Council as its senior member (2022-2024). In 2022, she received a "Distinguished Researcher Award" from the Brazilian Database Symposium. In the graduate program in Informatics at UFPR, she has supervised 4 PhD students, and 20 master's thesis. Her current projects involves Data Management for Environmental Monitoring and Modelling (REsMA-PrInt - Funding: CAPES) and Intelligent Systems for Marine Species Monitoring (Funding: CNPq).

Lorena Silva Nascimento is an oceanographer with a PhD in Biology and Ecology of Coastal and Oceanic Systems (Federal University of Paraná – UFPR, Brazil). She has expertise in the taxonomy and ecology of marine zooplankton, with an emphasis on copepods and gelatinous organisms. Her main interest involves the application of citizen science and iEcology in the marine monitoring and analysis of species distribution. She is also enthusiastic about scientific communication, working especially with social media use in environmental education. From February 2022 to February 2024, she served as the journal Aquatic Ecology editor. Currently, Lorena is a postdoctoral fellow in Zoology (Federal University of Paraíba – UFPB, Brazil) and works as associate editor of the journal Ocean and Coastal Research.

Received: March 01, 2024; Accepted: August 06, 2024.

Computer Science and Information Systems 21(4):1913–1961 https://doi.org/10.2298/CSIS240229065B

Automatic Conceptual Database Design based on Heterogeneous Source Artifacts*

Goran Banjac, Drazen Brdjanin, and Danijela Banjac

Faculty of Electrical Engineering, University of Banja Luka
Patre 5, 78000 Banja Luka, Bosnia and Herzegovina
{goran.banjac,drazen.brdjanin,danijela.banjac}@etf.unibl.org

Abstract. The article presents an approach to the automatic derivation of conceptual database models from heterogeneous source artifacts. The approach is based on the integration of conceptual database models that are derived from source artifacts of one single type by already existing tools, whereby those models possess limited certainty given their limited completeness and correctness. The uncertainty of the automatically derived models from specific source artifacts is expressed and managed through the effectiveness measure of the generation of specific concepts of the input conceptual database models. The approach is implemented by the DBomnia tool – the first online web-based tool enabling automatic derivation of conceptual database models from heterogeneous source artifacts (business process models and textual specifications). DBomnia employs other pre-existing tools to derive conceptual models from sources of the same type and then integrates those models. The case study-based evaluation proves that the implemented approach enables effective automatic derivation of the conceptual database model from a set of heterogeneous source artifacts. Moreover, the automatic derivation of the conceptual database model from a set of heterogeneous source artifacts is more effective than each independent automatic derivation of the conceptual database model from sources of one single type only.

Keywords: AMADEOS, DBomnia, Schema integration, Schema matching, Schema merging, TexToData, UML class diagram, Uncertain schema.

1. Introduction

The database design process undergoes several typical steps [20], whereby the first and the most important step is conceptual design. The result of the conceptual database design is the *conceptual database model* (CDM), which is platform-independent and describes the target database on a high level of abstraction. The result of all subsequent steps, including the database specification for the specific target database management system, can be straightforwardly derived starting from the CDM. This is the main reason why researchers have been significantly interested in the topic of automated CDM design.

Motivation. Automated CDM design has been a research topic since the 1980s [15], and since then a plethora of papers have been published in the field. Although different types

* This article constitutes an extended version of the conference paper entitled *"Towards Automatic Conceptual Database Design based on Heterogeneous Source Artifacts"* presented at the workshop *Modern Approaches in Data Engineering and Information System Design*, 4th of September 2023 Barcelona, Spain.

of artifacts (models, textual specifications, recorded speech, etc.) have been introduced as a source in the automated CDM design, the existing tools enable CDM synthesis based on sources of one single type only, whereby the generated CDMs are not 100% complete nor 100% correct.

Objectives. Since there is still no tool enabling the fully automatic generation of the complete target CDM from sources of one single type, we started to investigate the possibilities of increasing the completeness and correctness of automatically generated target CDMs by deriving them from heterogeneous source artifacts (compared to the CDMs derived from sources of one single type). Accordingly, we define our research objectives as follows:

1. *define an approach and implement a tool enabling automatic CDM derivation from a set of heterogeneous source artifacts,*
2. *evaluate the approach by assessing the CDM derived from a set of heterogeneous source artifacts against the CDMs that are derived from the source artifacts of one single type.*

Contributions. The main contribution of our research is the approach for the integration of incomplete and incorrect CDMs that are automatically derived from specific source artifacts. In our approach, we employ existing, already developed tools for CDM derivation from specific source artifacts, and then we integrate those CDMs. Uncertainty of CDMs automatically derived from specific source artifacts is expressed and managed through the effectiveness measure of generation of specific concepts (classes, attributes, associations, generalizations) of the input CDMs.

The second main contribution is the implemented tool named DBomnia. DBomnia is the first online web-based tool that enables automatic CDM derivation from a set of heterogeneous source artifacts. Currently, DBomnia supports two types of source artifacts: *business process models* (BPMs) and *textual specifications*. DBomnia is a full CASE tool for forward engineering of relational databases.

This article constitutes an extended version of the workshop paper [4], which is extended by: (*i*) formalization of the proposed approach, (*ii*) detailed presentation of the proposed approach, (*iii*) more detailed presentation of the related work, and (*iv*) case study-based evaluation of the proposed approach and implemented tool.

Article organization. This article is structured as follows. After this introduction, the second section presents the related work. The necessary definitions are provided in the third section. The proposed approach is presented in the next three sections. Firstly, an overview of the approach is presented in the fourth section, then the schema matching step is presented in the fifth section, while the sixth section presents the schema merging step. The implemented tool is presented in the seventh section. The eighth section presents the evaluation. The final section concludes the article.

2. Related Work

This section presents the related work. Firstly, we provide an overview of the existing approaches and tools to (semi-)automatic CDM design, followed by an overview of the existing approaches to schema matching and integration.

2.1. (Semi-)automatic CDM Design

The existing approaches and tools for (semi-)automatic CDM design derive the target model from the source artifacts of the same type and can be classified as: *Text-based*, *Model-based*, *Form-based*, and *Speech-based*.

Text-based approaches constitute the oldest and most dominant category. These approaches and tools derive CDMs from textual specifications that are typically unstructured and represented in some *natural language* (NL). They can be further classified (as suggested in [56]) as: (1) *Linguistics-based*, (2) *Pattern-based*, (3) *Case-based*, (4) *Ontology-based*, and (5) *Multiple approaches*.

Most text-based approaches and tools are *linguistics-based*. They use *natural language processing* (NLP) techniques to convert NL text into the CDM. The development of these approaches started with Chen's eleven rules [15] for the translation of English text into the corresponding CDM, which have been further enhanced and extended in [42, 40, 28]. The most important *linguistics-based* tools are: ER-Converter [40], CM-Builder [27], and LIDA [42]. The main representatives of other categories are: *pattern-based* APSARA [44], *case-based* CABSYDD [17], *ontology-based* OMDDE [54], and HBT [56] belonging to the *multiple approaches*.

The existing text-based tools typically support one single source NL (mainly English) and do not provide multilingual support. Only TexToData [13] enables automatic CDM derivation from textual specifications in different source NLs, even with very complex morphology (e.g. Slavic languages). In our approach presented in this article, we employ TexToData as one of the generators of uncertain CDMs in the multi-sourced automatic CDM synthesis.

Model-based approaches and tools emerged as an alternative to those that are text-based, in order to avoid their shortcomings mainly related to the modest effectiveness for languages with complex morphology. The existing approaches and tools take source models that can be represented by a number of different notations, which can be classified (according to [10]) as: (1) *Process-oriented* (e.g. BPMN), (2) *Function-oriented* (e.g. Data

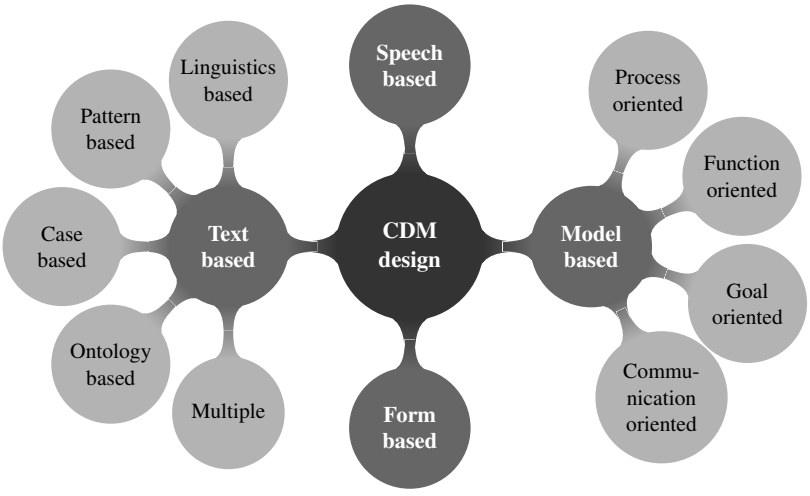


Fig. 1. Taxonomy of approaches to (semi-)automatic CDM design

Flow Diagram), (3) *Communication-oriented* (e.g. Sequence Diagram), and (4) *Goal-oriented* (e.g. TROPOS).

There is still no model-based approach nor tool enabling automatic derivation of the complete target CDM from a source model (regardless of the notation). Only a few papers present a set of formal rules for automatic CDM derivation (e.g. [55, 9]), while the majority give only guidelines and informal rules that do not enable automatic CDM derivation. Most of the proposed tools are actually transformation programs (e.g. [47, 31]) specified in some model-to-model transformation language (e.g. ATL [30]), while only a small number of papers present real CASE tools for automatic model-driven CDM synthesis (e.g. [39, 11]).

There is only one single online model-driven tool named AMADEOS [11], which enables the automatic derivation of an initial CDM from a set of BPMs. The most recent AMADEOS release [53] supports the whole BPM-driven forward database engineering process by using the standard UML notation in all stages. In our approach presented in this article, we employ AMADEOS as one of the generators of uncertain CDMs in the multi-sourced automatic CDM synthesis.

Form-based approaches take *collections of forms* as the source, whereby the most important tools are EDDS [16] and IIS*Case [33].

Speech-based approaches constitute the smallest category – in the existing literature, there is only one paper [12] presenting the SpeeD tool that is able to derive CDM from recorded English speech.

In comparison to the existing approaches that consider only sources of one single type, and in comparison to the existing tools that enable (semi-)automatic CDM derivation from sources of one single type, there is only our paper [4] that considers automatic CDM derivation from heterogeneous source artifacts and corresponding tool that enables automatic CDM derivation from two different types of source artifacts: textual specifications and BPMs.

2.2. Schema Matching and Integration

In the essence of our research objective is the problem of *uncertain schema* integration, i.e. integration of CDMs that are automatically generated by the tools that do not generate 100% correct nor 100% complete models.

Schema integration has been a research topic since the 1980s. It has been heavily investigated and is part of the research to this day. The term *schema integration*, introduced in [5], is defined as the activity of integrating database schemas. It is a generic term that is used in two contexts. In the database design context, it means to produce a global conceptual schema of the proposed database, and in the distributed database management context it means to produce a global schema that represents a virtual view of all databases in such an environment.

Schema integration implies the creation of a unified representation of the schemas from different sources. It consists of two core parts: *schema matching* and *schema merging*. Schema matching is the process of discovering mappings (correspondences) between different schemas, while schema merging is the process of the unification of different schemas based on the discovered mappings [36].

In the 2000s researchers started to investigate the topic of generic model management. In [8], the authors introduce the term *model management* and propose to generalize and integrate model management operations (such as matching and merging) to support generic model management. The term *model* is used to refer to any structured representation of the data (e.g. relational schema, XML, etc.). After that, numerous papers emerged focusing on generic model management operations, such as generic matching (e.g. [35]) and generic merging (e.g. [43]).

Although some papers deal with the uncertainty that is inherent in the schema matching step [36], the state of the art is the integration of the source schemas that can be described as *reliable*. Reliable source schemas are the product of reliable sources (e.g. when integrating schemas of the databases already in use). In this article, we deal with the integration of the models with reduced reliability, because tools for automated CDM derivation do not generate 100% correct nor 100% complete models.

Techniques. Over the years, many schema matching techniques have been introduced. Here we provide a basic taxonomy of matching techniques (according to [7]) with the focus on techniques adopted in our approach.

Basic categorization of schema matching techniques includes *instance-based* and *schema-based* techniques. Instance-based techniques use data instances to discover mappings among schema elements, while schema-based techniques rely only on schema information.

In our approach we deal with the conceptual (database) models, meaning there aren't any data instances that can be used during the schema matching process. Consequently, we are not able to use the instance-based techniques, which makes the schema-based techniques the cornerstone of our approach. Schema-based techniques can be categorized into *element-level* (they consider individual schema elements) and *structure-level* techniques (they consider groups of elements).

Another category of matching techniques is *constraint-based*. Constraint-based techniques use schema constraints such as data types, relationship cardinalities, participation constraints, etc.

Due to the unreliability of the input CDMs (in the sense of the CDM structure), which constitutes the main challenge in our approach, the usage of structure-based techniques is limited (but it is still possible to some extent). Therefore, we have to rely on *linguistic* matching techniques. Linguistic matching techniques use names and/or other textual descriptions of the schema elements to discover mappings between elements of different schemas. Linguistic matching (according to [57]) commonly includes string (e.g. element name) pre-processing (e.g. tokenization, elimination of special characters, stemming, lemmatization, elimination of stop words, etc.), identification of the *semantic similarity* (e.g. using NLP-based techniques) and/or identification of the *syntactic similarity*. Linguistic matching can include techniques of *auxiliary information usage* (e.g. thesaurus usage to find acronyms, synonyms, homonyms, heteronyms, etc.).

Our approach is not limited to single-language input artifacts and automatically generated CDMs, rather our approach limits the language in one execution to a single language (e.g. in one execution all artifacts can be in the English language, but it is possible that in another execution all artifacts are in the German language, etc.). Currently, a combination of multiple languages in one execution is not considered.

Due to the multilingual nature of the input CDMs (in different executions) and considering a large number of supported languages, the semantic analysis constitutes a great challenge and belongs to a separate research project that is out of the scope of this article. Here we are focused on the syntactic similarity measures.

Syntactic similarity measures for string comparison can be categorized (according to [24]) as *character-level* or *token-level*. Character-level measures compare strings on the level of a single character, while token-level measures compare string tokens. Typical representatives of character-level measures are *edit-distance measures* (Levenshtein [32], Damerau–Levenshtein [19], Needleman and Wunsch [38], Smith and Waterman [52], Smith–Waterman–Gotoh [25], Hamming [26], Jaro [29], Jaro–Winkler [58]) and *longest common string* [23]. Typical representatives of token-level measures are *set-based measures* (Jaccard [45], Dice [1]) and *bag-of-tokens measures* (Cosine [18], Euclidean [37], Manhattan [37]).

Some papers consider a combination of character-level and token-level measures. Such measures are called *soft measures* [24]. The basic principle of a soft measure is to apply the token-level measure, but when comparing tokens a character-level measure is used, i.e. two tokens are considered a match if their character-level measure is above some defined *threshold*. The typical usage of a soft measure is to eliminate possible typographical errors on a token level.

If different techniques are used to perform matching, then such combination of techniques is called *hybrid technique* (if different techniques are combined together to perform matching) or *composite technique* (if different techniques are performed independently and only combine the results). Our approach can be classified as a hybrid.

Usage of machine learning techniques. A number of schema matching approaches leverage of machine learning (ML) techniques. Some papers present approaches that apply ML techniques to the problem of combining different matchers (i.e. matching techniques). Examples of such approaches are LSD [21], YAM [22], and ALMa [46].

Another use case for ML techniques in the schema matching process is when such techniques are used as part of a similarity measure. The Automatch system [6] uses probabilistic methods and a knowledge base about schema attributes to calculate matching scores between the attributes of two schemas. Afterwards, an optimization process is carried out to find optimal schema matching (with respect to the sum of the individual attribute matching scores). Schema matching at the element level and structure level is proposed in [2], where ML techniques are used to generate matching results at the element level. Features for ML are constructed using different string similarity metrics and text processing techniques based on entity names. Element level and structure level mappings are combined to produce the final result. In [14], features are constructed using string similarity metrics and semantic information from entity names. These features are used as input to an ML model that predicts entity matchings. In [48], features (about attributes) for ML are constructed based on schema information (data type, not null specification, etc.). These features are used to create corresponding clusters, and attribute mappings are discovered from these clusters. SMAT [59] is based on NLP techniques and uses attribute names and descriptions to obtain schema mappings. Matching between two attributes is proportionate to the similarity of the corresponding sentence pair, where each sentence is constructed from attribute name and description.

Usage of ML techniques in the schema matching process is diverse. Corpus-based matching approach [34] uses a corpus of schemas and mappings in particular domain to augment information about input schemas. Authors argue that augmenting evidence about input schemas provide better matching results. The ADnEV approach [51] presents an additional step in the schema matching task. It uses deep learning to calibrate matching result of other (algorithmic) matchers. PoWareMatch [50] combines algorithmic matching and deep learning to improve human matching results.

Lately, researchers have tried to leverage the recent emergence of (large) language models. Paper [61] presents a novel linguistic schema matching approach that uses fine-tuned pre-trained language model for a more precise calculation of similarity between each pair of attributes in input schemas. The problem of similarity calculation is regarded as a binary text classification problem, where prediction of a correct match is based on a sentence that describes corresponding attributes (concatenation of the attributes' names and descriptions). The SMUTF approach [60] combines rule-based feature engineering, pre-trained language models, and generative large language models to predict the match score (the probability that two columns are matched). The ReMatch approach [49] uses retrieval-enhanced large language models in the process of selecting candidate matches. The approach is designed to aid human matchers throughout their work.

Schema matching approaches that use ML techniques also focus on matching complete and reliable schemas, with the dominant usage of attribute names and descriptions, and attribute values (where possible). However, in our future work we will focus on utilizing ML techniques to improve current results.

3. Definitions

In this section, we provide the necessary definitions.

3.1. CDM Representation

We use the *UML class diagram* [41] to represent the CDMs, thus the representation of the CDMs generated by different generators, as well as the target integrated CDM, is unified.

Each CDM (UML class diagram) contains two packages, ICM_PT and ICM_CD. The ICM_PT package contains definitions of the primitive types (e.g. *Integer*, *Text*, *Double*, etc.), while the ICM_CD package contains the CDM itself.

Definition 1: Let P, E, R, and G be sets of primitive types, classes, associations, and generalizations, respectively. The conceptual database model, $CDM(P, E, R, G)$, is a UML class diagram with the following properties:

1. Each entity type is represented by the corresponding class $e \in E$ of the same name.
2. Let $attrs(e)$ be a set of attributes (*ownedAttribute:Property*) of the class $e \in E$. Each attribute $a \in attrs(e)$ corresponds to an attribute of the respective entity type, whereby its type, denoted $type(a)$, is a primitive type ($type(a) \in P$).
3. A class may contain a single operation, named *PK*, which represents the primary key of the corresponding entity type [53]. The primary key attributes should be listed as the operation parameters, i.e. operation parameters should correspond to the primary key attributes by the name and by the type.

4. Each relationship type is represented by the corresponding binary association $r \in R$ of the same name, whose *memberEnd:Property* attribute is an ordered pair of association ends ($source(r)$ and $target(r)$). The type of each association end is an entity type ($type(source(r)) \in E \wedge type(target(r)) \in E$). The association end multiplicities correspond to the relationship cardinalities and participation constraints. The lower value of the association end multiplicity ($lower(source(r))$ and $lower(target(r))$) corresponds to the participation constraint of the entity type at the opposite end of the association. The upper value of the association end multiplicity ($upper(source(r))$ and $upper(target(r))$) corresponds to the mapping cardinality of the entity type at the opposite end of the association.
5. Each generalization/specialization relationship is represented by the corresponding generalization $g \in G$. It relates *specific* entity type ($specific(g) \in E$) to a more *general* entity type ($general(g) \in E, specific(g) \neq general(g)$).

3.2. Quantitative CDM Assessment

The input in the schema integration process is a set of automatically generated CDMs, which are not 100% correct nor 100% complete. In order to integrate such unreliable CDMs, it is necessary to know their correctness and completeness. The only way to precisely determine such measures for some concrete automatically generated CDM is to manually evaluate it. Considering that we are dealing with automated schema integration, manual evaluation is not an option, but rather approximate or estimated values of those measures should be known in advance. Since all our generators are already evaluated, we propose *a posteriori* approach – use the calculated measures from the evaluation of the CDM generators as the estimated measures of the automatically generated CDMs in the schema integration process.

During the evaluation of generators, we used *recall*, *precision*, and *F-score* as measures for the evaluation of the automatically generated CDM.

Definition 2: If N_c represents the number of correctly generated concepts in the generated CDM and N_m represents the number of missing concepts in the generated CDM, then **recall** (R) constitutes a measure of completeness of the generated CDM, and it is defined as:

$$R = \frac{N_c}{N_c + N_m}\,. \tag{1}$$

Definition 3: If N_c represents the number of correctly generated concepts in the generated CDM and N_w represents the number of incorrectly generated concepts in the generated CDM, then **precision** (P) constitutes a measure of correctness of the generated CDM, and it is defined as:

$$P = \frac{N_c}{N_c + N_w}\,. \tag{2}$$

Definition 4: The **effectiveness**, named *F-score* or *F-measure* (F), is defined as the harmonic mean of precision and recall, i.e.

$$F = \frac{2PR}{P + R}\,. \tag{3}$$

3.3. Input and Output

Previously described measures are calculated separately for each concept of the CDM, i.e. classes, attributes, associations, and generalizations. The F-score, as the effectiveness measure, will be used in the integration approach. Therefore, input in the schema integration process is a set of automatically generated CDMs, as well as estimated measures for each input CDM.

Definition 5: Let F_E be the estimated effectiveness for classes, and F_A be the estimated effectiveness for attributes in the classes, and F_R be the estimated effectiveness for associations, and F_G be the estimated effectiveness for generalizations, then the estimated effectiveness F of the CDM generation process is represented by the $< F_E, F_A, F_R, F_G >$ tuple, i.e. $F =< F_E, F_A, F_R, F_G >$.

Definition 6: The input I in the schema integration process, is a set of n ordered pairs (CDM_i, F_i), where CDM_i is the *i*-th CDM, while F_i represents its estimated F-measures, i.e.

$$I = \{(CDM_i, F_i) : CDM_i = CDM(P_i, E_i, R_i, G_i) \wedge F_i =< F_{E_i}, F_{A_i}, F_{R_i}, F_{G_i} > \wedge i = 1, \ldots, n\}. \quad (4)$$

Definition 7: The output of the schema integration process is the integrated CDM $CDM_O = CDM(P_O, E_O, R_O, G_O)$.

3.4. Syntactic Similarity Assessment

The names of the concepts in input (automatically generated by our generators) CDMs typically consist of one or a few words, usually without stop words and possibly with special characters such as underscore ('_'). Therefore, tokenization of names and a set-based measure stand out as the candidates for use. Also, to eliminate possible typographical errors, a set-based measure should be reinforced with the character-level measure – thus making it a soft measure.

There are (too) many possible combinations of the token-level and character-level measures and, possibly, some combinations would prove to be a little better than others for this concrete purpose. However, by defining empirically determined *threshold* for the character-level measure, we can achieve acceptable results despite the chosen measures in the concrete implementation of the tool. Also, when a new type of source artifacts is introduced and expected measures are determined for the CDM derived from those source artifacts, again, by (if necessary) modified threshold, chosen (matching) measures should also run smoothly in that case.

We tried some combinations of measures, and, at the moment, decided to go with the combination of Jaccard similarity as the token-level measure, and Levenshtein distance as the character-level measure. Therefore, further in the article, a syntactic similarity between two strings A and B, denoted $synt_s(A, B)$, is calculated as follows.

First, we tokenize the strings on special characters and uppercase letters (e.g. $tokens(BookEdition) = \{book, edition\}$), and we compare two sets of tokens $tokens(A)$ and $tokens(B)$ to determine the intersection. To determine the intersection

of two sets of tokens, we use Levenshtein distance – if the value of Levenshtein distance between tokens $t_a \in tokens(A)$ and $t_b \in tokens(B)$ is less than defined *threshold* (LD), then tokens t_a and t_b are added to intersection (as a single element) of token sets. Such intersection of token sets is denoted $tokens(A) \cap_{LD} tokens(B)$.

Definition 8: Considering that Jaccard similarity [45] of two sets P and Q (in general) is calculated as a ratio between the number of elements in the intersection of the sets ($|P \cap Q|$) and the number of elements in the union of the sets ($|P \cup Q|$), i.e.

$$Jaccard(P, Q) = \frac{|P \cap Q|}{|P \cup Q|} = \frac{|P \cap Q|}{|P| + |Q| - |P \cap Q|}, \tag{5}$$

we calculate the syntactic similarity of two strings A and B as follows:

$$synt_s(A, B) = \frac{|tokens(A) \cap_{LD} tokens(B)|}{|tokens(A)| + |tokens(B)| - |tokens(A) \cap_{LD} tokens(B)|}. \tag{6}$$

Note that when determining the number of elements in the union of token sets, we only use Levenshtein distance when computing the intersection of token sets.

4. Approach Overview

Our task is to integrate multiple (unreliable) CDMs into a single unified CDM. The schema integration process consists of two core parts: *schema matching* and *schema merging*. Schema matching is the process of discovering mappings (correspondences) between different schemas, while schema merging is the process of unification of different schemas based on the discovered mappings [36]. Therefore, the schema matching process precedes the schema merging process. Figure 2 shows an overview of the schema integration process. The input in the schema integration process are (unreliable) CDMs automatically generated by tools for automatic derivation of CDMs from specific source artifacts (e.g. CDM derived from a source set of BPMs by AMADEOS, CDM derived from a textual specification by TexToData).

Based on the type of CDM concepts, we divided the schema matching process into four steps: (i) *classes matching step*, (ii) *attributes matching step*, (iii) *associations matching step*, and (iv) *generalizations matching step*. The top elements of the CDM are classes, so the first step in the schema matching process is the classes matching step. Other matching steps (attributes matching step, associations matching step, and generalizations matching step) are directly dependent on the result of the classes matching step. Therefore, the classes matching step must precede other steps in the schema matching process. However, the attributes matching step, associations matching step, and generalizations matching step are not mutually dependent, therefore these steps can be executed in an arbitrary order.

The schema merging process is, also, divided into four steps: (i) *classes merging step*, (ii) *attributes merging step*, (iii) *associations merging step*, and (iv) *generalizations merging step*. The classes merging step is dependent only on the classes matching step, while other merging steps are dependent on the classes merging step and corresponding matching step. Therefore, the classes merging step must precede other merging steps. Other merging steps are not mutually dependent, therefore these steps can be executed in an arbitrary order.

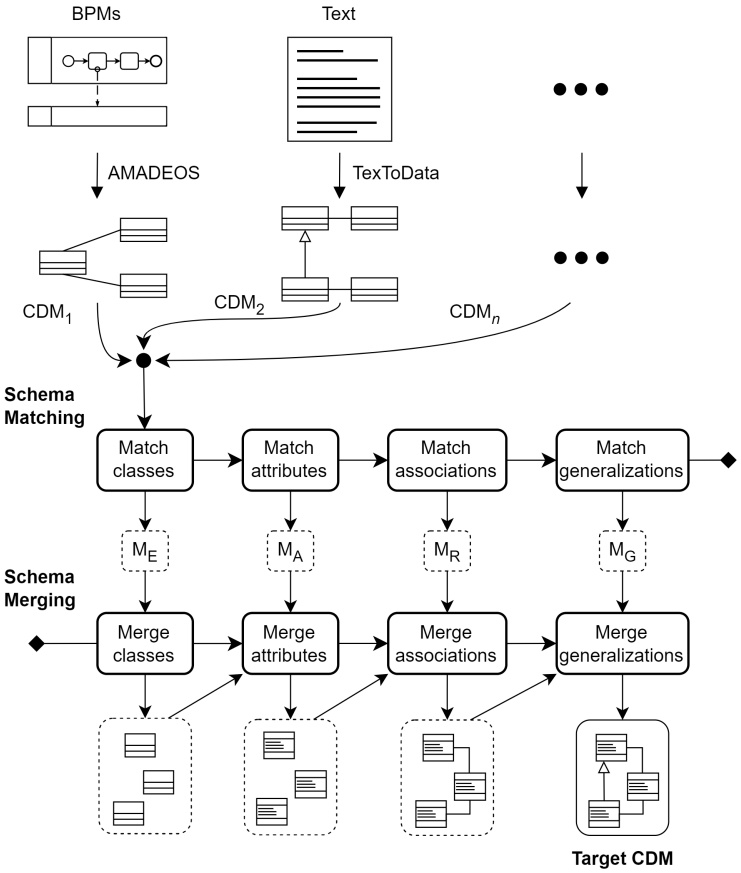


Fig. 2. Overview of the schema integration process

The schema matching and schema merging processes are described in detail in the following two sections.

Note that the default input in the algorithms is I (equation (4)), i.e. a set of n ordered pairs (CDM_i, F_i), so we purposely omitted the explicit definition of such input (in the *require* section) and explicitly stated only necessary inputs that are produced as intermediate results of other algorithms.

5. Schema Matching

The schema matching process in the task of automated schema integration is very important because the result of the schema merging process (i.e. the final result of the schema integration task) directly depends on the result of the schema matching process. Inadequate schema matching may lead to a lower value of the effectiveness of the integrated CDM than the value of the effectiveness of each input CDM (CDMs that are being integrated).

As already stated, the cornerstone of our approach are schema-based matching techniques, which consider only schema information. In our case, schema information includes names, properties (e.g. the properties of classes are attributes and connections to other classes such as associations and generalizations), and constraints (e.g. cardinalities and participation constraints for associations). The usage of structure-based techniques depends directly on the reliability of generated CDM concepts which is, in our approach, directly expressed with the effectiveness measures.

Due to the unreliability of the input CDMs, the usage of structure-based techniques is limited. Therefore, we need to use the names of the concepts and rely on linguistic matching. In cases where we cannot more rely on the structure and constraints, we have to mostly rely on the names. On the other hand, if the reliability of the structure is good, then we can combine linguistic matching and structure-based matching and/or constraint-based matching.

In our approach, we expect that all input CDMs in one execution are in the same language, but in different executions, the input CDMs can be in different languages. This brings us to the fact that our approach is not a single-language approach (focusing on only one language). Due to the multilingual nature of the input CDMs (in different executions), we are focused on the syntactic similarity measure for linguistic matching. The semantic analysis constitutes a great challenge and it is not considered in this article.

The aforementioned combination of different matching techniques classifies our approach in the hybrid matching category. Concerning *matching cardinality*, we produce 1:1:...:1 mapping between schema elements, which means that, for example, a class from CDM_1 can be matched with at most one class from CDM_2 and at most one class from CDM_3 and, so on, at most one class from CDM_n.

5.1. Classes Matching

In the classes matching step, it is necessary to discover class mappings based on the sets of classes from input CDMs ($E_1, \ldots, E_n$). The result of the classes matching step is a *list*[1] of sets of matched classes from different CDMs, $M_E = [M_E^{(1)}, \ldots, M_E^{(m)}]$, where:

- m is the number of sets of matched classes (each element $M_E^{(i)} \in M_E$ is the set of classes),
- each set $M_E^{(i)}$ must be a non-empty set,
- each class from each input CDM ($\forall e \in E_i, i = 1, \ldots, n$) belongs to exactly one set $M_E^{(j)}$, and
- not a single set $M_E^{(i)}$ contains multiple classes from the same CDM.

The minimal number of elements in any set $M_E^{(i)}$ is 1, which means that the corresponding class $e \in M_E^{(i)}$ is not matched with any other class from any other CDM. The maximal number of elements in any set $M_E^{(i)}$ is equal to the number of input CDMs (n).

[1] We use the term *list* (according to [3]) to denote an ordered collection of elements (the elements of the list will usually be sets, but in general may be other entities) separated by commas and surrounded by square brackets (e.g. $X = [X^{(1)}, X^{(2)}, X^{(3)}]$ is the list of length 3).

The classes matching step is executed in two phases. In the first phase (Algorithm 1), we compare every pair of classes (e_a, e_b) from different CDMs ($e_a \in E_i$, $e_b \in E_j$, $i \neq j$), which results in the set of triplets $(e_a, e_b, sim_E(e_a, e_b))$. The result of the comparison of two classes e_a and e_b, is the *similarity coefficient*, denoted $sim_E(e_a, e_b) \in [0, 1]$. A higher value of the similarity coefficient means that the corresponding classes are more likely to be the same class, while the lower value means that the corresponding classes are less likely to be the same class. If the similarity coefficient is greater than or equal to the defined threshold (th_{S_E}), then the triplet is accepted and added to the output set Y_E.

The value of the similarity coefficient for classes from different CDMs, $e_a \in E_i$, $e_b \in E_j$, $i \neq j$, is determined from classes' names and attributes, i.e.

$$\begin{aligned} sim_E(e_a, e_b) = {} & synt_s(name(e_a), name(e_b)) \cdot (1 - attr_w) \\ & + attr_s(attrs(e_a), attrs(e_b)) \cdot attr_w\,, \end{aligned} \tag{7}$$

where $attr_w = min(F_{A_i}, F_{A_j}) \cdot W_A$ represents the attributes' similarity weight[2] (which depends on the effectiveness measure for attributes of the corresponding CDMs and empirically determined coefficient $W_A \in [0, 1]$), and $attr_s(attrs(e_a), attrs(e_b)) \in [0, 1]$ is the similarity measure for the sets of attributes of the corresponding classes. The similarity measure for two sets of attributes, $A = attrs(e_a)$ and $B = attrs(e_b)$, similar to the calculation of two sets of string tokens, is calculated using Jaccard similarity, i.e.

$$attr_s(A, B) = \frac{|A \cap_{sim_A} B|}{|A| + |B| - |A \cap_{sim_A} B|}\,. \tag{8}$$

where the intersection of two sets of attributes is determined using the similarity measure for comparison of two attributes. The calculation of the similarity of two attributes is described in the next subsection. The combination of linguistic matching (names) and structure-based matching (attributes) makes classes matching a hybrid technique.

Algorithm 1 The first phase of the classes matching step

Ensure: Y_E
1: $Y_E \leftarrow \emptyset$
2: **for all** $E_i : i = 1, \ldots, n-1$ **do**
3: **for all** $e_a \in E_i$ **do**
4: **for all** $E_j : j = i+1, \ldots, n$ **do**
5: **for all** $e_b \in E_j$ **do**
6: $sim_{ab} \leftarrow sim_E(e_a, e_b)$
7: **if** $sim_{ab} \geq th_{S_E}$ **then**
8: $Y_E \leftarrow Y_E \cup \{(e_a, e_b, sim_{ab})\}$
9: **end if**
10: **end for**
11: **end for**
12: **end for**
13: **end for**

[2] Values $attr_w$ and $(1 - attr_w)$ allow us to normalize the $sim_E(e_a, e_b)$ value, as well as to favor more reliable similarity measure ($synt_s(name(e_a), name(e_b))$ or $attr_s(attrs(e_a), attrs(e_b))$).

Based on the results from the first phase (i.e. the Y_E set), in the second phase (Algorithm 2), we try to produce sets of best matching classes from different CDMs. In the beginning, the result set M_E is empty. We also create a helper set T_E containing classes that have not yet been matched. In the beginning, this set contains all classes from all input CDMs. Later, when we match some class, then we remove it from this set. Thus, this set enables easy check (whenever necessary) whether some class is already matched or not.

Until the input set does not become an empty set, steps are repeated as follows. First, we find the triplet $(e_a, e_b, sim_E(e_a, e_b))$ in the input set with the highest value of similarity, i.e. the triplet with the most compatible pair of classes. Such triplet is then removed from the input set (in the next iteration of the *while* loop, the next triplet should be considered). The next step is to check if class e_a or class e_b are already matched with

Algorithm 2 The second phase of the classes matching step

Require: Y_E
Ensure: $M_E = [M_E^{(1)}, \ldots, M_E^{(m)}]$
1: $M_E \leftarrow \emptyset$; $T_E \leftarrow \cup_{i=1}^{n} E_i$
2: **while** $Y_E \neq \emptyset$ **do**
3: $(e_a, e_b, sim_E(e_a, e_b)) \leftarrow max_{sim}(Y_E)$
4: $Y_E \leftarrow Y_E \setminus \{(e_a, e_b, sim_E(e_a, e_b))\}$
5: **if** $e_a \in T_E \wedge e_b \in T_E$ **then**
6: $me \leftarrow \{e_a, e_b\}$; $M_E \leftarrow M_E \cup \{me\}$; $T_E \leftarrow T_E \setminus me$
7: **else if** $e_a \in T_E$ **then**
8: $me \leftarrow me_t : me_t \in M_E \wedge e_b \in me_t$; $E_t \leftarrow E_i : e_a \in E_i$
9: **if** $me \cap E_t = \emptyset$ **then**
10: $me \leftarrow me \cup \{e_a\}$; $T_E \leftarrow T_E \setminus \{e_a\}$
11: **end if**
12: **else if** $e_b \in T_E$ **then**
13: $me \leftarrow me_t : me_t \in M_E \wedge e_a \in me_t$; $E_t \leftarrow E_i : e_b \in E_i$
14: **if** $me \cap E_t = \emptyset$ **then**
15: $me \leftarrow me \cup \{e_b\}$; $T_E \leftarrow T_E \setminus \{e_b\}$
16: **end if**
17: **else**
18: $me_a \leftarrow me_t : me_t \in M_E \wedge e_a \in me_t$; $me_b \leftarrow me_t : me_t \in M_E \wedge e_b \in me_t$
19: $found \leftarrow false$
20: **for all** $E_i : i = 1, \ldots, n$ **do**
21: **if** $E_i \cap me_a \neq \emptyset \wedge E_i \cap me_b \neq \emptyset$ **then**
22: $found \leftarrow true$
23: **end if**
24: **end for**
25: **if** $\neg found$ **then**
26: $me_a \leftarrow me_a \cup me_b$; $M_E \leftarrow M_E \setminus \{me_b\}$
27: **end if**
28: **end if**
29: **end while**
30: **for all** $e \in T_E$ **do**
31: $me \leftarrow \{e\}$; $M_E \leftarrow M_E \cup \{me\}$
32: **end for**

some other classes. If classes e_a and e_b are not already matched with some other classes ($e_a \in T_E \wedge e_b \in T_E$), then classes e_a and e_b are the elements of the new set that needs to be added to the output set. If only class e_a is already matched with some other classes, then it is necessary to add class e_b to that set of classes that are matched with class e_a (if that does not break the rule that set must not contain multiple classes from same input CDM). Analogously, if class e_b is already matched with some other classes, then it is necessary to add class e_a to that set of classes that are matched with class e_b (if that does not break the rule that set must not contain multiple classes from same input CDM). Otherwise, if both classes are already matched, but not together (they are in different sets), then those sets can be joined (if that does not break the rule that set must not contain multiple classes from the same input CDM). Finally, all unmatched classes from all CDMs are added to separate (single element) new sets. In the following steps, the order of the elements in the result is important so the final result (M_E) can be considered to be the list of sets.

Example. To illustrate the classes matching step, let us consider Fig. 3 which shows an illustrative example of three input CDMs ($n = 3$): $CDM_1 = CDM(P_1, E_1, R_1, G_1)$, $CDM_2 = CDM(P_2, E_2, R_2, G_2)$, and $CDM_3 = CDM(P_3, E_3, R_3, G_3)$, where:

$P_1 = \{Text\}, E_1 = \{A1, B1\}, R_1 = \{Z1\}, G_1 = \emptyset$,
$P_2 = \{Text\}, E_2 = \{A2, B2, C2\}, R_2 = \{Z2, Y2, X2\}, G_2 = \emptyset$,
$P_3 = \{Text\}, E_3 = \{A3, B3, D3\}, R_3 = \{Z3\}, G_3 = \emptyset$.

In the first phase of the classes matching step, we compare classes from different CDMs, i.e. $A1$ and $A2$, $A1$ and $B2$, $A1$ and $C2$, $A1$ and $A3$, etc. Let us assume that the result of the comparison, i.e. the result of the first phase is the following set of triplets (for simplicity, the set is shown in a supposed descending order by the value of the similarity coefficient):

$$Y_E = \{(A1, A2, sim_E(A1, A2)), (A1, A3, sim_E(A1, A3)), (A2, A3, sim_E(A2, A3)), (B1, B2, sim_E(B1, B2)), (B1, B3, sim_E(B1, B3)), (B2, B3, sim_E(B2, B3)) (A1, B2, sim_E(A1, B2))\}.$$

Note that the Y_E set contains only combinations where similarity is greater than or equal to the defined threshold.

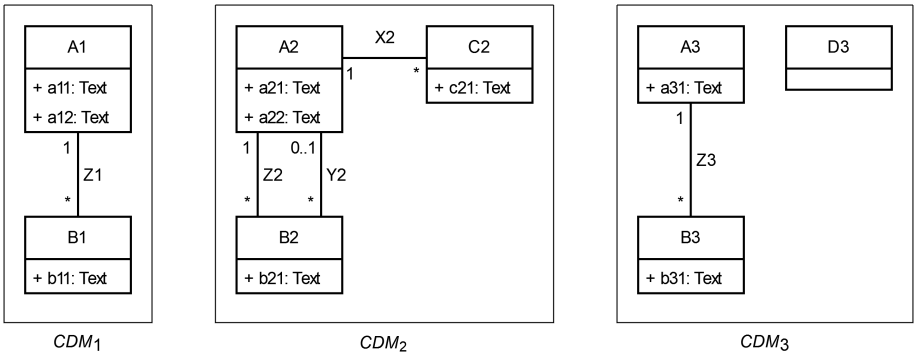


Fig. 3. Illustrative generic example of three input CDMs

The second phase of the classes matching step (i.e. the population of the M_E set, based on the Y_E set) is illustrated in the following lines.

$$\begin{aligned} M_E &\Rightarrow \{\emptyset\} \Rightarrow \{\{A1, A2\}\} \Rightarrow \{\{A1, A2, A3\}\} \Rightarrow \{\{A1, A2, A3\}, \{B1, B2\}\} \\ &\Rightarrow \{\{A1, A2, A3\}, \{B1, B2, B3\}\} \Rightarrow \{\{A1, A2, A3\}, \{B1, B2, B3\}, \{C2\}\} \\ &\Rightarrow \{\{A1, A2, A3\}, \{B1, B2, B3\}, \{C2\}, \{D3\}\} \\ &\Rightarrow [\{A1, A2, A3\}, \{B1, B2, B3\}, \{C2\}, \{D3\}]\,. \end{aligned}$$

At the beginning, classes $A1$ and $A2$ form a new set of matched classes, and then the $A3$ class is added to the same set (containing the $A1$ class). The $(A2, A3)$ pair is ignored, because both classes are already matched (together). The $\{B1, B2, B3\}$ set is formed in a similar manner. The $(A1, B2)$ pair is ignored because joining the sets containing these classes would result in a set that contains multiple classes from a single input CDM. At the end, sets $\{C2\}$ and $\{D3\}$ are formed from the unmatched classes.

5.2. Attributes Matching

In the attributes matching step, it is necessary to match attributes of already matched classes. The input in this step is the output of the classes matching step, i.e. the list of sets of matched classes from different CDMs ($M_E = [M_E^{(1)}, \ldots, M_E^{(m)}]$).

The algorithm for attributes matching is similar to the classes matching step – first, it is necessary to compare attributes of matched classes, and then create sets of matched attributes. The output of the attributes matching step is the list of sets of sets of matched attributes $M_A = [M_A^{(1)}, \ldots, M_A^{(m)}]$, where each set $M_A^{(i)}$ is the set of sets of matched attributes of classes from the $M_E^{(i)}$ set.

The input in the first part of the algorithm (Algorithm 3) is the M_E list of sets of matched classes from different CDMs, while the output is the Y_A list of sets of triplets $(at_a, at_b, sim_A(at_a, at_b))$. Each set $Y_A^{(i)} \in Y$ corresponds to the set of matched classes $M_E^{(i)} \in M_E$. Also, each set $Y_A^{(i)}$ is empty in the beginning. Since it is necessary to match attributes only for matched classes, then the following steps are repeated for each input set of matched classes $M_E^{(i)}$. Each attribute ($at_a \in attrs(e_j)$) of each class ($e_j \in M_E^{(i)}$) is compared with every attribute ($at_b \in attrs(e_k)$) of every other class from the same set of matched classes ($e_k \in M_E^{(i)}$). The result of the comparison of two attributes is the similarity coefficient $sim_A(at_a, at_b)$. Attributes at_a and at_b, and the similarity coefficient $sim_A(at_a, at_b)$ form a triplet $(at_a, at_b, sim_A(at_a, at_b))$ which is added to the output set $Y_A^{(i)}$ if the value of the similarity coefficient is greater than or equal to the defined threshold (th_{S_A}).

The value of the similarity coefficient for two attributes (of classes from different CDMs) is determined based on attributes' names, i.e.

$$sim_A(at_a, at_b) = synt_s(name(at_a), name(at_b))\,. \tag{9}$$

Here we completely rely on the attribute names (linguistic matching) as they are, currently, the most useful information about attributes. If proved to be necessary, it will be easy to include attribute types in the equation.

Algorithm 3 The first phase of the attributes matching step

Require: $M_E = [M_E^{(1)}, \ldots, M_E^{(m)}]$
Ensure: $Y_A = [Y_A^{(1)}, \ldots, Y_A^{(m)}]$
for all $M_E^{(i)} \in M_E$ **do**
  $Y_A^{(i)} \leftarrow \emptyset$; $n_c \leftarrow |M_E^{(i)}|$
  for all $e_j \in M_E^{(i)} : j = 1, \ldots, n_c - 1$ **do**
    for all $at_a \in attrs(e_j)$ **do**
      for all $e_k \in M_E^{(i)} : k = j + 1, \ldots, n_c$ **do**
        for all $at_b \in attrs(e_k)$ **do**
          $sim_{ab} \leftarrow sim_A(at_a, at_b)$
          if $sim_{ab} \geq th_{S_A}$ **then**
            $Y_A^{(i)} \leftarrow Y_A^{(i)} \cup \{(at_a, at_b, sim_{ab})\}$
          end if
        end for
      end for
    end for
  end for
end for

In the second phase (Algorithm 4) of the attributes matching step, it is necessary to create sets of matched attributes. The input is the list of sets of matched classes and the list of sets of similarity triplets. The output of the attributes matching step is the list of sets of sets of matched attributes, where each set $M_A^{(i)} \in M_A$ is the set of sets of matched attributes that correspond to the set of matched classes $M_E^{(i)} \in M_E$.

In the beginning, each output set $M_A^{(i)}$ is initialized to an empty set, and the T_A set contains attributes from all classes from the $M_E^{(i)}$ set. When an attribute is matched, we remove it from the T_A set (which makes it easy to check whether some attribute is already matched or not). For each input set $Y_A^{(i)} \in Y_A$ of the similarity triplets it is necessary to create sets of matched attributes as follows. In each iteration of the inner *while* loop, we find the $(at_a, at_b, sim_A(at_a, at_b))$ triplet with the highest value of similarity. That triplet is removed from the input set (in the next iteration, another triplet should be considered). Then, it is necessary to check if any of the attributes, at_a or at_b, has already been matched with other attributes. If neither of the current attributes is already matched with other attributes ($at_a \in T_A \wedge at_b \in T_A$), then the given attributes at_a and at_b form a new set of matched attributes (ma) that is added to the set of sets of matched attributes. If one of the attributes is already matched with other attributes, i.e. if there exists set $ma_t \in M_A^{(i)}$ such that $at_a \in ma_t$ or $at_b \in ma_t$, then it is necessary to add other attribute to the same set (if this does not break the rule that the set must not contain multiple attributes from the same class). Otherwise, if both attributes are already matched, but not together (they are in different sets), then those sets can be joined (if this does not break the rule that the set must not contain multiple attributes from the same class). Finally, all unmatched attributes are added to separate (single element) new sets – unmatched attributes of classes from the $M_E^{(i)}$ set are added to separate sets of the output set $M_A^{(i)}$.

Algorithm 4 The second phase of the attributes matching step

Require: $M_E = [M_E^{(1)}, \ldots, M_E^{(m)}] \quad Y_A = [Y_A^{(1)}, \ldots, Y_A^{(m)}]$
Ensure: $M_A = [M_A^{(1)}, \ldots, M_A^{(m)}]$

for all $Y_A^{(i)} \in Y_A$ **do**
  $M_A^{(i)} \leftarrow \emptyset$; $T_A \leftarrow \cup_{\forall e \in M_E^{(i)}} attrs(e)$
  while $Y_A^{(i)} \neq \emptyset$ **do**
    $(at_a, at_b, sim_A(at_a, at_b)) \leftarrow max_{sim}(Y_A^{(i)})$
    $Y_A^{(i)} \leftarrow Y_A^{(i)} \setminus \{(at_a, at_b, sim_A(at_a, at_b))\}$
    if $at_a \in T_A \wedge at_b \in T_A$ **then**
      $ma \leftarrow \{at_a, at_b\}$; $M_A^{(i)} \leftarrow M_A^{(i)} \cup \{ma\}$; $T_A \leftarrow T_A \setminus ma$
    else if $at_a \in T_A$ **then**
      $ma \leftarrow ma_t : ma_t \in M_A^{(i)} \wedge at_b \in ma_t$; $e \leftarrow e_t : e_t \in M_E^{(i)} \wedge at_a \in attrs(e_t)$
      if $ma \cap attrs(e) = \emptyset$ **then**
        $ma \leftarrow ma \cup \{at_a\}$; $T_A \leftarrow T_A \setminus \{at_a\}$
      end if
    else if $at_b \in T_A$ **then**
      $ma \leftarrow ma_t : ma_t \in M_A^{(i)} \wedge at_a \in ma_t$
      $e \leftarrow e_t : e_t \in M_E^{(i)} \wedge at_b \in attrs(e_t)$
      if $ma \cap attrs(e) = \emptyset$ **then**
        $ma \leftarrow ma \cup \{at_b\}$; $T_A \leftarrow T_A \setminus \{at_b\}$
      end if
    else
      $ma_a \leftarrow ma_t : ma_t \in M_A^{(i)} \wedge at_a \in ma_t$
      $ma_b \leftarrow ma_t : ma_t \in M_A^{(i)} \wedge at_b \in ma_t$
      $found \leftarrow false$
      for all $e \in M_E^{(i)}$ **do**
        if $attrs(e) \cap ma_a \neq \emptyset \wedge attrs(e) \cap ma_b \neq \emptyset$ **then**
          $found \leftarrow true$
        end if
      end for
      if $\neg found$ **then**
        $ma_a \leftarrow ma_a \cup ma_b$; $M_A^{(i)} \leftarrow M_A^{(i)} \setminus \{ma_b\}$
      end if
    end if
  end while
  for all $at \in T_A$ **do**
    $ma \leftarrow \{at\}$; $M_A^{(i)} \leftarrow M_A^{(i)} \cup \{ma\}$
  end for
end for

Example. Now we will illustrate the attributes matching step using the example from the previous subsection. Here we repeat the output, i.e. the M_E list of sets of matched classes:

$$M_E = [M_E^{(1)}, M_E^{(2)}, M_E^{(3)}, M_E^{(4)}],$$

where

$$M_E^{(1)} = \{A1, A2, A3\},\ M_E^{(2)} = \{B1, B2, B3\},\ M_E^{(3)} = \{C2\},\ M_E^{(4)} = \{D3\}.$$

In the first phase, we need to create the output list Y_A of sets of similarity triplets for attributes. The number of elements in the output list corresponds to the number of elements in the list M_E ($m = 4$), and element/set $Y_A^{(i)}$ corresponds to the element/set $M_E^{(i)}$. Note that set $Y_A^{(i)}$ contains similarity triplets for attributes of only those classes that are in the $M_E^{(i)}$ set. Finally, let us assume the following output of the first phase (elements in the sets are shown in a supposed descending order by the value of the similarity coefficient):

$$\begin{aligned} Y_A =& \{Y_A^{(1)}, Y_A^{(2)}, Y_A^{(3)}, Y_A^{(4)}\}, \\ Y_A^{(1)} =& \{(a11, a21, sim_A(a11, a21)), (a21, a31, sim_A(a21, a31)), \\ & (a12, a22, sim_A(a12, a22))\}, \\ Y_A^{(2)} =& \{(b11, b21, sim_A(b11, b21)), (b11, b31, sim_A(b11, b31)), \\ & (b21, b31, sim_A(b21, b31))\}, \\ Y_A^{(3)} =& Y_A^{(4)} = \{\emptyset\}. \end{aligned}$$

In the second phase of the algorithm, we create the output list of sets of sets of matched attributes $M_A = [M_A^{(1)}, M_A^{(2)}, M_A^{(3)}, M_A^{(4)}]$, where each $M_A^{(i)}$ is created from the set of similarity triplets $Y_A^{(i)}$. The following lines describe the population of each set $M_A^{(i)}$.

$$\begin{aligned} & M_A^{(1)} \Rightarrow \{\emptyset\} \Rightarrow \{\{a11, a21\}\} \Rightarrow \{\{a11, a21, a31\}\} \Rightarrow \{\{a11, a21, a31\}, \{a12, a22\}\} \\ & M_A^{(2)} \Rightarrow \{\emptyset\} \Rightarrow \{\{b11, b21\}\} \Rightarrow \{\{b11, b21, b31\}\} \\ & M_A^{(3)} \Rightarrow \{\emptyset\} \Rightarrow \{\{c21\}\} \\ & M_A^{(4)} \Rightarrow \{\emptyset\}. \end{aligned}$$

The $M_A^{(3)}$ set is empty in the beginning. Since $Y_A^{(3)}$ is empty, only a set containing (unmatched) attribute $c21$ is added to $M_A^{(3)}$. The $M_A^{(4)}$ set remains empty because $Y^{(4)}$ is empty and the only class in $M_E^{(4)}$ ($D3 \in M_E^{(4)}$) has no attributes.

5.3. Associations Matching

The next step in the matching process is the associations matching step. In this step, it is necessary to match associations between different input CDMs with respect to already matched classes. For example, let us consider the result in the classes matching step example, i.e. the list of sets of matched classes

$$M_E = [\{A1, A2, A3\}, \{B1, B2, B3\}, \{C2\}, \{D3\}].$$

In the associations matching step, it is necessary to create the sets of matched associations for all combinations of sets of matched classes ($M_E^{(i)}$ and $M_E^{(j)}$ for $i = 1, \ldots, m$ and $i \leq j \leq m$). If the list of sets of matched classes has m elements, then the list of matched associations has to have $m(m+1)/2$ elements. In this example, since $m=4$ and $m(m+1)/2=10$, it is necessary to create the list with ten elements, i.e.

$$M_R = [M_R^{(1)(1)}, M_R^{(1)(2)}, M_R^{(1)(3)}, M_R^{(1)(4)}, \\ M_R^{(2)(2)}, M_R^{(2)(3)}, M_R^{(2)(4)}, M_R^{(3)(3)}, M_R^{(3)(4)}, M_R^{(4)(4)}] .$$

The elements of the sets $M_R^{(i)(j)}$, where $i \neq j$, should be the sets of matched associations between the classes from sets $M_E^{(i)}$ and $M_E^{(j)}$. For example, the $M_R^{(1)(2)}$ element should be the set of sets of matched associations between the classes from sets $M_E^{(1)}$ and $M_E^{(2)}$. Since the $M_E^{(1)}$ set contains classes $A1$, $A2$, and $A3$, and the $M_E^{(2)}$ set contains classes $B1$, $B2$, and $B3$, then we must compare the associations between the pair of classes $(A1, B1)$ and the associations between the pair of classes $(A2, B2)$ and the associations between the pair of classes $(A3, B3)$.

The elements of the sets $M_R^{(i)(i)}$ should be sets of matched reflexive associations of the classes from the $M_E^{(i)}$ set. For example, the $M_R^{(1)(1)}$ element should be the set of sets of matched reflexive associations between the classes from set $M_E^{(1)}$. Since the $M_E^{(1)}$ set contains classes $A1$, $A2$, and $A3$, then we must compare reflexive associations of the $A1$ class and reflexive associations of the $A2$ class and reflexive associations of the $A3$ class.

In order to produce the list of sets of sets of matched associations, we split the associations matching step into two phases (similar to previous matching steps). In the first phase (Algorithm 5), we only compare the associations and create a list of sets of triplets $(r_a, r_b, sim_R(r_a, r_b))$, where $sim_R(r_a, r_b)$ is the result of the comparison (similarity coefficient) of the associations r_a and r_b. As mentioned earlier, if the list of sets of matched classes has m elements, the Y_R list of sets of triplets $(r_a, r_b, sim_R(r_a, r_b))$ has $m(m+1)/2$ elements. In the beginning, each set $Y_R^{(i)(j)}$ is initialized to an empty set, and then it is populated as follows. First, it is necessary to find the pair of classes (e_{k_a}, e_{k_b}) from the *k*-th input CDM such that $e_{k_a} \in M_E^{(i)}$ and $e_{k_b} \in M_E^{(j)}$. Then, it is necessary to find all associations (R'_k) between that pair of classes. The next step is to find the pair of classes (e_{p_a}, e_{p_b}) from the *p*-th input CDM such that $e_{p_a} \in M_E^{(i)}$ and $e_{p_b} \in M_E^{(j)}$. Here is also necessary to find all associations (R'_p) between that pair of classes. Now, we can compare all associations from R'_k with all associations from R'_p. If the value of the similarity coefficient is greater than or equal to the defined threshold (th_{S_R}), then the $(r_a, r_b, sim_R(r_a, r_b))$ triplet is added to the $Y_R^{(i)(j)}$ set. These steps are repeated for every possible combination of k and p, i.e. for each input set of associations. All these steps represent the steps for the population of the output set $Y_R^{(i)(j)}$.

As described above, we compare only associations between already matched classes. Let there be two associations from different CDMs, $r_a \in R_k$ and $r_b \in R_p$. Let r_a be an association between the classes $e_{k_a} \in E_k \wedge e_{k_a} \in M_E^{(i)}$ and $e_{k_b} \in E_k \wedge e_{k_b} \in M_E^{(j)}$, and let r_b be an association between the classes $e_{p_a} \in E_p \wedge e_{p_a} \in M_E^{(i)}$ and $e_{p_b} \in E_p \wedge e_{p_b} \in M_E^{(j)}$. This means that one end of the first association is a class matched with a class that is one end of the second association (both classes are in the $M_E^{(i)}$ set). Also, the other end of the first association is a class matched with a class that is the other end of the second association (both classes are in the $M_E^{(j)}$ set). The value of the similarity coefficient for such two associations (and we only compare associations between the pairs of matched

classes) is determined based on associations' names and constraints (cardinalities and participation constraints), i.e.

$$\begin{aligned} sim_R(r_a, r_b) = synt_s(name(r_a), name(r_b)) \cdot (1 - W_{card} - W_{pc}) \\ + ceq(r_a, r_b) \cdot W_{card} + peq(r_a, r_b) \cdot W_{pc}\,, \end{aligned} \tag{10}$$

where $ceq(r_a, r_b) \in \{0, 1\}$ checks whether the cardinalities of associations are equal, i.e.

$$ceq(r_a, r_b) = \begin{cases} 1, & (type(source(r_a)) \in M_E^{(i)} \wedge type(source(r_b)) \in M_E^{(i)} \\ & \vee type(source(r_a)) \in M_E^{(j)} \wedge type(source(r_b)) \in M_E^{(j)}) \\ & \wedge upper(source(r_a)) = upper(source(r_b)) \\ & \wedge upper(target(r_a)) = upper(target(r_b)) \\ & \vee \\ & (type(source(r_a)) \in M_E^{(i)} \wedge type(source(r_b)) \in M_E^{(j)} \\ & \vee type(source(r_a)) \in M_E^{(j)} \wedge type(source(r_b)) \in M_E^{(i)}) \\ & \wedge upper(source(r_a)) = upper(target(r_b)) \\ & \wedge upper(target(r_a)) = upper(source(r_b)) \\ 0, & \text{otherwise} \end{cases}, \tag{11}$$

Algorithm 5 The first phase of the associations matching step

Require: $M_E = [M_E^{(1)}, \ldots, M_E^{(m)}]$
Ensure: $Y_R = [Y_R^{(1)(1)}, \ldots, Y_R^{(1)(m)}, Y_R^{(2)(2)}, \ldots, Y_R^{(2)(m)}, \ldots Y_R^{(m)(m)}]$

for $i = 1, \ldots, m$ **do**
  for $j = i, \ldots, m$ **do**
    $Y_R^{(i)(j)} \leftarrow \emptyset$
    for $k = 1, \ldots, n-1$ **do**
      $e_{k_a} \leftarrow e : e \in E_k \wedge e \in M_E^{(i)}$; $e_{k_b} \leftarrow e : e \in E_k \wedge e \in M_E^{(j)}$
      $R'_k \leftarrow \{r : r \in R_k \wedge (source(r) = e_{k_a} \wedge target(r) = e_{k_b} \vee$
        $source(r) = e_{k_b} \wedge target(r) = e_{k_a})\}$
      for all $r_a \in R'_k$ **do**
        for $p = k+1, \ldots, n$ **do**
          $e_{p_a} \leftarrow e : e \in E_p \wedge e \in M_E^{(i)}$; $e_{p_b} \leftarrow e : e \in E_p \wedge e \in M_E^{(j)}$
          $R'_p \leftarrow \{r : r \in R_p \wedge (source(r) = e_{p_a} \wedge target(r) = e_{p_b} \vee$
            $source(r) = e_{p_b} \wedge target(r) = e_{p_a})\}$
          for all $r_b \in R'_p$ **do**
            $sim_{ab} \leftarrow sim_R(r_a, r_b)$
            if $sim_{ab} \geq th_{S_R}$ **then**
              $Y_R^{(i)(j)} \leftarrow Y_R^{(i)(j)} \cup \{(r_a, r_b, sim_{ab})\}$
            end if
          end for
        end for
      end for
    end for
  end for
end for

$peq(r_a, r_b) \in \{0, 1\}$ checks whether the participation constraints are equal, i.e.

$$peq(r_a, r_b) = \begin{cases} 1, & (type(source(r_a)) \in M_E^{(i)} \wedge type(source(r_b)) \in M_E^{(i)} \\ & \vee type(source(r_a)) \in M_E^{(j)} \wedge type(source(r_b)) \in M_E^{(j)}) \\ & \wedge lower(source(r_a)) = lower(source(r_b)) \\ & \wedge lower(target(r_a)) = lower(target(r_b)) \\ & \vee \\ & (type(source(r_a)) \in M_E^{(i)} \wedge type(source(r_b)) \in M_E^{(j)} \\ & \vee type(source(r_a)) \in M_E^{(j)} \wedge type(source(r_b)) \in M_E^{(i)}) \\ & \wedge lower(source(r_a)) = lower(target(r_b)) \\ & \wedge lower(target(r_a)) = lower(source(r_b)) \\ 0, & \text{otherwise} \end{cases}, \quad (12)$$

and W_{card} and W_{pc} are the empirically determined coefficients ($0 \leq W_{card} + W_{pc} \leq 1$).

The combination of linguistic matching (names) and constraint-based matching (cardinalities and participation constraints) makes the associations matching, also, a hybrid technique.

In the second phase (Algorithm 6), it is necessary to produce the list of sets of sets of matched associations based on the results from the first phase. The output list M_R has the same number of elements as the input list Y_R, i.e. $m(m+1)/2$ (this corresponds to every combination of sets of the list M_E). The T_R set is a utility set of unmatched associations between the corresponding pairs of classes from the sets $M_E^{(i)}$ and $M_E^{(j)}$ (when an association is matched it is then removed from this set). Each element $M_R^{(i)(j)}$ of the output list M_R is created from the corresponding set $Y_R^{(i)(j)}$ of triplets $(r_a, r_b, sim_R(r_a, r_b))$ as follows. First, it is necessary to find the triplet $(r_a, r_b, sim_R(r_a, r_b))$ in the $Y_R^{(i)(j)}$ set with the highest value of similarity. The found triplet is removed from the $Y_R^{(i)(j)}$ set (in the next iteration of the *while* loop, the next triplet must be considered). Now, it is necessary to check if any of the associations, r_a or r_b, has already been matched with other associations. If neither of the current associations is already matched with other associations ($r_a \in T_R \wedge r_b \in T_R$), then the given associations r_a and r_b form a new set of matched associations that is added to the $M_R^{(i)(j)}$ set. If one of the associations is already matched with other associations, i.e. if there exists a set $mr_t \in M_R^{(i)(j)}$ such that $r_a \in mr_t$ or $r_b \in mr_t$, then it is necessary to add other association to the same set (if this does not break the rule that the set must not contain multiple associations from the same input CDM). Otherwise, if both associations are already matched, but not together (they are in different sets), then those sets can be joined (if this does not break the rule that the set must not contain multiple associations from the same input CDM). Finally, all unmatched associations are added to separate (single element) new sets – unmatched associations from each input CDM between the corresponding classes in sets $M_E^{(i)}$ and $M_E^{(j)}$ are added to separate new sets of the output set $M_R^{(i)(j)}$.

Algorithm 6 The second phase of the associations matching step

Require: $M_E = [M_E^{(1)}, \ldots, M_E^{(m)}] \quad Y_R = [Y_R^{(1)(1)}, \ldots, Y_R^{(1)(m)}, Y_R^{(2)(2)}, \ldots, Y_R^{(m)(m)}]$
Ensure: $M_R = [M_R^{(1)(1)}, \ldots, M_R^{(1)(m)}, M_R^{(2)(2)}, \ldots, M_R^{(2)(m)}, \ldots, M_R^{(m)(m)}]$
for all $Y_R^{(i)(j)} \in Y_R$ **do**
  $M_R^{(i)(j)} \leftarrow \emptyset$; $T_R \leftarrow \emptyset$
  for $k = 1, \ldots, n$ **do**
    $e_{k_a} \leftarrow e : e \in E_k \wedge e \in M_E^{(i)}$; $e_{k_b} \leftarrow e : e \in E_k \wedge e \in M_E^{(j)}$
    $T_R \leftarrow T_R \cup \{r : r \in R_k \wedge (source(r) = e_{k_a} \wedge target(r) = e_{k_b} \vee$
      $source(r) = e_{k_b} \wedge target(r) = e_{k_a})\}$
  end for
  while $Y_R^{(i)(j)} \neq \emptyset$ **do**
    $(r_a, r_b, sim_R(r_a, r_b)) \leftarrow max_{sim}(Y_R^{(i)(j)})$
    $Y_R^{(i)(j)} \leftarrow Y_R^{(i)(j)} \setminus \{(r_a, r_b, sim_R(r_a, r_b))\}$
    if $r_a \in T_R \wedge r_b \in T_R$ **then**
      $M_R^{(i)(j)} \leftarrow M_R^{(i)(j)} \cup \{\{r_a, r_b\}\}$; $T_R \leftarrow T_R \setminus \{r_a, r_b\}$
    else if $r_a \in T_R$ **then**
      $mr \leftarrow mr_t : mr_t \in M_R^{(i)(j)} \wedge r_b \in mr_t$; $R_t \leftarrow R_k : r_a \in R_k$
      if $mr \cap R_t = \emptyset$ **then**
        $mr \leftarrow mr \cup \{r_a\}$; $T_R \leftarrow T_R \setminus \{r_a\}$
      end if
    else if $r_b \in T_R$ **then**
      $mr \leftarrow mr_t : mr_t \in M_R^{(i)(j)} \wedge r_a \in mr_t$; $R_t \leftarrow R_k : r_b \in R_k$
      if $mr \cap R_t = \emptyset$ **then**
        $mr \leftarrow mr \cup \{r_b\}$; $T_R \leftarrow T_R \setminus \{r_b\}$
      end if
    else
      $mr_a \leftarrow mr_t : mr_t \in M_R^{(i)(j)} \wedge r_a \in mr_t$
      $mr_b \leftarrow mr_t : mr_t \in M_R^{(i)(j)} \wedge r_b \in mr_t$
      $found \leftarrow false$
      for all $R_k : k = 1, \ldots, n$ **do**
        if $R_k \cap mr_a \neq \emptyset \wedge R_k \cap mr_b \neq \emptyset$ **then**
          $found \leftarrow true$
        end if
      end for
      if $\neg found$ **then**
        $mr_a \leftarrow mr_a \cup mr_b$; $M_R^{(i)(j)} \leftarrow M_R^{(i)(j)} \setminus \{mr_b\}$
      end if
    end if
  end while
  for all $r \in T_R$ **do**
    $M_R^{(i)(j)} \leftarrow M_R^{(i)(j)} \cup \{\{r\}\}$
  end for
end for

Example. Continuing the example from previous subsections, here we need to produce the list of sets of sets of matched associations. In order to do that, first we need to compare the associations and create a list of ten ($m = |M_E| = 4 \Rightarrow m(m+1)/2 = 10$) sets of similarity triplets, i.e.

$$Y_R = [Y_R^{(1)(1)}, Y_R^{(1)(2)}, Y_R^{(1)(3)}, Y_R^{(1)(4)}, \\ Y_R^{(2)(2)}, Y_R^{(2)(3)}, Y_R^{(2)(4)}, Y_R^{(3)(3)}, Y_R^{(3)(4)}, Y_R^{(4)(4)}] .$$

To populate the $Y_R^{(1)(2)}$ set we need to compare the associations between the pairs of classes $(A1, B1)$, $(A2, B2)$, and $(A3, B3)$, i.e. associations $Z1$, $Z2$, $Y2$, and $Z3$. Other sets of similarity triplets will remain empty because there are no reflexive associations and sets $M_E^{(3)}$ and $M_E^{(4)}$ contain only a single class.

Let us assume the following output of the first phase (elements in the sets are shown in a supposed descending order by the value of the similarity coefficient):

$$\begin{aligned} Y_R^{(1)(1)} =& Y_R^{(1)(3)} = Y_R^{(1)(4)} = \{\emptyset\}, \\ Y_R^{(1)(2)} =& \{(Z1, Z2, sim_R(Z1, Z2)), (Z1, Z3, sim_R(Z1, Z3)), \\ & (Z2, Z3, sim_R(Z2, Z3)), (Z1, Y2, sim_R(Z1, Y2))\}, \\ Y_R^{(2)(2)} =& Y_R^{(2)(3)} = Y_R^{(2)(4)} = Y_R^{(3)(3)} = Y_R^{(3)(4)} = Y_R^{(4)(4)} = \{\emptyset\} . \end{aligned}$$

In the second phase of the algorithm, we create the output list of sets of sets of matched associations M_R, where each $M_R^{(i)(j)}$ is created from the $Y_R^{(i)(j)}$ set of similarity triplets. The following lines describe the population of each set $M_R^{(i)(j)} \in M_R$.

$$\begin{aligned} & M_R^{(1)(2)} \Rightarrow \{\emptyset\} \Rightarrow \{\{Z1, Z2\}\} \Rightarrow \{\{Z1, Z2, Z3\}\} \Rightarrow \{\{Z1, Z2, Z3\}, \{Y2\}\}, \\ & M_R^{(1)(3)} \Rightarrow \{\emptyset\} \Rightarrow \{\{X2\}\}, \\ & M_R^{(1)(1)}, M_R^{(1)(4)}, M_R^{(2)(2)}, M_R^{(2)(3)}, M_R^{(2)(4)}, M_R^{(3)(3)}, M_R^{(3)(4)}, M_R^{(4)(4)} \Rightarrow \{\emptyset\} . \end{aligned}$$

The $M_R^{(1)(3)}$ set is empty in the beginning. Since $Y_R^{(1)(3)}$ is an empty set, only a set containing (unmatched) association $X2$ is added to the $M_R^{(1)(3)}$ set. The sets $M_R^{(1)(1)}$, $M_R^{(1)(4)}$, $M_R^{(2)(2)}$, $M_R^{(2)(3)}$, $M_R^{(2)(4)}$, $M_R^{(3)(3)}$, $M_R^{(3)(4)}$, and $M_R^{(4)(4)}$ remain empty because there are no respective associations in the input CDMs.

5.4. Generalizations Matching

The generalizations matching step is similar to the associations matching step, with the exception that there can be only one generalization between two classes. The generalizations matching step is, also, split into two phases.

In the first phase (Algorithm 7), we create a list of sets of triplets $(g_a, g_b, sim_G(g_a, g_b))$, where $sim_G(g_a, g_b)$ is the result of the comparison (similarity coefficient) of the generalizations g_a and g_b. If the list of sets of matched classes has m elements, then the Y_G list of sets of triplets $(g_a, g_b, sim_G(g_a, g_b))$ has $m(m-1)/2$ elements because it is necessary to create the sets of matched generalizations for all combinations of sets of matched classes ($M_E^{(i)}$ and $M_E^{(j)}$ for $i = 1, \ldots, m-1$ and $i < j \leq m$). The number of elements of the output list for generalizations is less than the number of elements of the output list for associations because we had to take into account reflexive associations.

Algorithm 7 The first phase of the generalizations matching step

Require: $M_E = [M_E^{(1)}, \ldots, M_E^{(m)}]$
Ensure: $Y_G = [Y_G^{(1)(2)}, \ldots, Y_G^{(1)(m)}, Y_G^{(2)(3)}, \ldots, Y_G^{(2)(m)}, \ldots, Y_G^{(m-1)(m)}]$
for $i = 1, \ldots, m-1$ **do**
  for $j : i+1, \ldots, m$ **do**
    $Y_G^{(i)(j)} \leftarrow \emptyset$
    for $k = 1, \ldots, n-1$ **do**
      $e_{k_a} \leftarrow e : e \in E_k \wedge e \in M_E^{(i)}$; $e_{k_b} \leftarrow e : e \in E_k \wedge e \in M_E^{(j)}$
      $g_a \leftarrow g : g \in G_k \wedge (specific(g) = e_{k_a} \wedge general(g) = e_{k_b} \vee$
        $specific(g) = e_{k_b} \wedge general(g) = e_{k_a})$
      for $p = k+1, \ldots, n$ **do**
        $e_{p_a} \leftarrow e : e \in E_p \wedge e \in M_E^{(i)}$; $e_{p_b} \leftarrow e : e \in E_p \wedge e \in M_E^{(j)}$
        $g_b \leftarrow g : g \in G_p \wedge (specific(g) = e_{p_a} \wedge general(g) = e_{p_b} \vee$
          $specific(g) = e_{p_b} \wedge general(g) = e_{p_a})$
        $sim_{ab} \leftarrow sim_G(g_a, g_b)$
        if $sim_{ab} \geq th_{S_G}$ **then**
          $Y_G^{(i)(j)} \leftarrow Y_G^{(i)(j)} \cup \{(g_a, g_b, sim_{ab})\}$
        end if
      end for
    end for
  end for
end for

In the beginning, each set $Y_G^{(i)(j)}$ is initialized to an empty set, and then it is populated as follows. First, we need to find a pair of classes (e_{k_a}, e_{k_b}) from CDM_k such that $e_{k_a} \in M_E^{(i)}$ and $e_{k_b} \in M_E^{(j)}$, and then we need to find the generalization (g_a) between that pair of classes. Then, we need to find a pair of classes (e_{p_a}, e_{p_b}) from CDM_p such that $e_{p_a} \in M_E^{(i)}$ and $e_{p_b} \in M_E^{(j)}$, and then we, also, need to find the generalization (g_b) between that pair of classes. Now, we can compare generalization g_a with the generalization g_b, and add the $(g_a, g_b, sim_G(g_a, g_b))$ triplet to the $Y_G^{(i)(j)}$ set if the value of the similarity coefficient is greater than or equal to the defined threshold (th_{S_G}). These steps are repeated for every combination of k and p, i.e. for each input set of generalizations. All these steps represent the steps for the population of the output set $Y_G^{(i)(j)}$.

As described above, we compare only generalizations between already matched classes. Let there be two generalizations from different CDMs, $g_a \in G_k$ and $g_b \in G_p$. Let g_a be a generalization between the classes $e_{k_a} \in E_k \wedge e_{k_a} \in M_E^{(i)}$ and $e_{k_b} \in E_k \wedge e_{k_b} \in M_E^{(j)}$, and let g_b be a generalization between the classes $e_{p_a} \in E_p \wedge e_{p_a} \in M_E^{(i)}$ and $e_{p_b} \in E_p \wedge e_{p_b} \in M_E^{(j)}$. The value of the similarity coefficient for such two generalizations is determined only based on the direction, i.e.

$$sim_G(g_a, g_b) = \begin{cases} 1, & specific(g_a) \in M_E^{(i)} \wedge specific(g_b) \in M_E^{(i)} \\ & \vee \\ & specific(g_a) \in M_E^{(j)} \wedge specific(g_b) \in M_E^{(j)} \\ 0, & \text{otherwise} \end{cases}. \quad (13)$$

In the second phase (Algorithm 8), it is necessary to produce a list of sets of matched generalizations based on the results from the first phase. The output list M_G has the same number of elements as the input list Y_G, i.e. $m(m-1)/2$.

Algorithm 8 The second phase of the generalizations matching step

Require: $M_E = [M_E^{(1)}, \ldots, M_E^{(m)}] \quad Y_G = [Y_G^{(1)(2)}, \ldots, Y_G^{(1)(m)}, Y_G^{(2)(3)}, \ldots, Y_G^{(m-1)(m)}]$
Ensure: $M_G = [M_G^{(1)(2)}, \ldots, M_G^{(1)(m)}, M_G^{(2)(3)}, \ldots, M_G^{(2)(m)}, \ldots, M_G^{(m-1)(m)}]$
for all $Y_G^{(i)(j)} \in Y_G$ **do**
  $M_G^{(i)(j)} \leftarrow \emptyset$; $T_G \leftarrow \emptyset$
  for $k = 1, \ldots, n$ **do**
    $e_{k_a} \leftarrow e : e \in E_k \wedge e \in M_E^{(i)}$; $e_{k_b} \leftarrow e : e \in E_k \wedge e \in M_E^{(j)}$
    $T_G \leftarrow T_G \cup \{g : g \in G_k \wedge (specific(g) = e_{k_a} \wedge general(g) = e_{k_b} \vee$
      $specific(g) = e_{k_b} \wedge general(g) = e_{k_a})\}$
  end for
  while $Y_G^{(i)(j)} \neq \emptyset$ **do**
    $(g_a, g_b, sim_G(g_a, g_b)) \leftarrow max_{sim}(Y_G^{(i)(j)})$
    $Y_G^{(i)(j)} \leftarrow Y_G^{(i)(j)} \setminus \{(g_a, g_b, sim_G(g_a, g_b))\}$
    if $g_a \in T_G \wedge g_b \in T_G$ **then**
      $M_G^{(i)(j)} \leftarrow M_G^{(i)(j)} \cup \{\{g_a, g_b\}\}$; $T_G \leftarrow T_G \setminus \{g_a, g_b\}$
    else if $g_a \in T_G$ **then**
      $mg \leftarrow mg_t : mg_t \in M_G^{(i)(j)} \wedge g_b \in mg_t$; $G_t \leftarrow G_k : g_a \in G_k$
      if $mg \cap G_t = \emptyset$ **then**
        $mg \leftarrow mg \cup \{g_a\}$; $T_G \leftarrow T_G \setminus \{g_a\}$
      end if
    else if $g_b \in T_G$ **then**
      $mg \leftarrow mg_t : mg_t \in M_G^{(i)(j)} \wedge g_a \in mg_t$; $G_t \leftarrow G_k : g_b \in G_k$
      if $mg \cap G_t = \emptyset$ **then**
        $mg \leftarrow mg \cup \{g_b\}$; $T_G \leftarrow T_G \setminus \{g_b\}$
      end if
    else
      $mg_a \leftarrow mg_t : mg_t \in M_G^{(i)(j)} \wedge g_a \in mg_t$
      $mg_b \leftarrow mg_t : mg_t \in M_G^{(i)(j)} \wedge g_b \in mg_t$
      $found \leftarrow false$
      for all $G_k : k = 1, \ldots, n$ **do**
        if $G_k \cap mg_a \neq \emptyset \wedge G_k \cap mg_b \neq \emptyset$ **then**
          $found \leftarrow true$
        end if
      end for
      if $\neg found$ **then**
        $mg_a \leftarrow mg_a \cup mg_b$; $M_G^{(i)(j)} \leftarrow M_G^{(i)(j)} \setminus \{mg_b\}$
      end if
    end if
  end while
  for all $g \in T_G$ **do**
    $M_G^{(i)(j)} \leftarrow M_G^{(i)(j)} \cup \{\{g\}\}$
  end for
end for

Each element $M_G^{(i)(j)}$ of the output list M_G is created from the corresponding set $Y_G^{(i)(j)}$ of triplets $(g_a, g_b, sim_G(g_a, g_b))$ as follows. First, it is necessary to find the $(g_a, g_b, sim_G(g_a, g_b))$ triplet in the $Y_G^{(i)(j)}$ set with the highest value of similarity. The found triplet is removed from the $Y_G^{(i)(j)}$ set (in the next iteration of the *while* loop, the next triplet must be considered). Then, it is necessary to check if any of the generalizations, g_a or g_b, has already been matched with other generalizations. If neither of the current generalizations is already matched with other generalizations ($g_a \in T_G \wedge g_b \in T_G$), then given generalizations g_a and g_b form a new set of matched generalizations that is added to the $M_G^{(i)(j)}$ set. If one of the generalizations is already matched with other generalizations, i.e. if there exists set $mg_t \in M_G^{(i)(j)}$ such that $g_a \in mg_t$ or $g_b \in mg_t$, then it is necessary to add other generalization to the same set (if this does not break the rule that the set must not contain multiple generalizations from the same input CDM). Otherwise, if both generalizations are already matched, but not together (they are in different sets), then those sets can be joined (if this does not break the rule that the set must not contain multiple generalizations from the same input CDM). Finally, all unmatched generalizations are added to separate (single element) new sets – unmatched generalizations from each input CDM between the corresponding classes in sets $M_E^{(i)}$ and $M_E^{(j)}$ are added to separate new sets of the output set $M_G^{(i)(j)}$.

Due to the article length and the similarity of the generalizations matching step to the associations matching step (there can be only one generalization between two classes, while there can be more than one association between two classes) we do not provide an example for the generalizations matching step.

6. Schema Merging

The merging process is the process of creating a new integrated CDM ($CDM_O = CDM(P_O, E_O, R_O, G_O)$) based on the mappings discovered in the matching process.

6.1. Primitive types

The creation of primitive types is the pre-step in the merging process, and it is a rather simple one. The P_O set of primitive types of the integrated CDM is a union of primitive types of the input CDMs, i.e.

$$P_O = \cup_{i=1}^{n} P_i \,. \tag{14}$$

For example, the set of primitive types of the integrated CDM in the sample used in the previous section is

$$P_O = \cup_{i=1}^{3} P_i = P_1 \cup P_2 \cup P_3 = \{Text\} \,.$$

6.2. Classes

The merging process starts with the classes (Algorithm 9) as the root elements of the CDM. The input in the classes merging step is the output of the classes matching step, i.e.

the list of sets of matched classes $M_E = [M_E^{(1)}, \ldots, M_E^{(m)}]$. Each set $M_E^{(i)}$ of matched classes is a candidate to be mapped into the class in the integrated CDM. The $M_E^{(i)}$ set is mapped into the $e_o \in E_O$ class if at least one of the following criteria is met:

- the number of elements in the $M_E^{(i)}$ set is equal to the number of input CDMs ($|M_E^{(i)}| = n$), meaning this set contains a class from each input CDM, or
- the $M_E^{(i)}$ set contains a class from an input CDM with the effectiveness for classes greater than or equal to the defined threshold ($e \in M_E^{(i)} \wedge e \in E_j \wedge F_{E_j} \geq th_{F_E}$), meaning the set contains a class from an input CDM with the acceptable effectiveness for classes, or
- the $M_E^{(i)}$ set contains a relevant class. A class is said to be relevant if its *relevance coefficient* is greater than or equal to the defined threshold (th_{R_E}).

The relevance coefficient of the $e \in E_j$ class is determined based on its attributes, associations, and generalizations, i.e.

$$\begin{aligned} rel_E(e) = {} & |attrs(e)| \cdot RW_A \\ & + (|\{r : r \in R_j \wedge (source(r) = e \vee target(r) = e)\}| \\ & + |\{g : g \in G_j \wedge (specific(g) = e \vee general(g) = e)\}|) \cdot RW_R \,, \end{aligned} \tag{15}$$

where RW_A and RW_R represent empirically determined weight coefficients for attributes and relationships (associations and generalizations), respectively. The value of the relevance coefficient is not normalized.

Algorithm 9 The classes merging step

Require: $M_E = [M_E^{(1)}, \ldots, M_E^{(m)}]$
Ensure: $I_E = [I_E^{(1)}, \ldots, I_E^{(m)}]$
Ensure: E_O
1: $E_O \leftarrow \emptyset$
2: **for all** $M_E^{(i)} \in M_E$ **do**
3: $cr_met \leftarrow false$
4: **if** $|M_E^{(i)}| = n$ **then**
5: $cr_met \leftarrow true$
6: **else**
7: **for all** $e \in M_E^{(i)}$ **do**
8: **if** $e \in E_j \wedge F_{E_j} \geq th_{F_E} \vee rel_E(e) \geq th_{R_E}$ **then**
9: $cr_met \leftarrow true$
10: **end if**
11: **end for**
12: **end if**
13: **if** cr_met **then**
14: $e_o \leftarrow new_class()$; $e' \leftarrow max_{F_E}(M_E^{(i)})$
15: $name(e_o) \leftarrow name(e')$; $E_O \leftarrow E_O \cup \{e_o\}$; $I_E^{(i)} \leftarrow e_o$
16: **else**
17: $I_E^{(i)} \leftarrow \emptyset$
18: **end if**
19: **end for**

The name of the class (if created) is inherited from a class from CDM with the highest effectiveness for classes. Besides the set of classes of integrated CDM (E_O), the output of the algorithm is a list of indicators of created classes, where $I_E^{(i)}$ is the class created from the $M_E^{(i)}$ set or $I_E^{(i)} = \emptyset$ if the $M_E^{(i)}$ set is not mapped into a class.

Example. The classes merging step will be illustrated using the result from the example of the classes matching step, i.e., the following list of sets of matched classes

$$M_E = [\{A1, A2, A3\}, \{B1, B2, B3\}, \{C2\}, \{D3\}] .$$

The output of the classes merging step, is the list of indicators of created classes

$$I_E = [I_E^{(1)}, I_E^{(2)}, I_E^{(3)}, I_E^{(4)}] .$$

First, we consider the set $M_E^{(1)} = \{A1, A2, A3\}$. The number of elements in this set is equal to the number of input CDMs ($|\{A1, A2, A3\}| = 3$), and we map this set into a new class (AI) in the integrated CDM, and the indicator $I_E^{(1)}$ is set to this class ($I_E^{(1)} = AI$). If we assume that CDM_1 has the highest effectiveness for classes, then a new class in the integrated CDM inherits the name from the class $A1$ ($A1 \in E_1$), i.e. $name(AI) = name(A1)$.

Similarly, the next set $M_E^{(2)}$ is also mapped into a new class (BI) in the integrated CDM and the $I_E^{(2)}$ indicator is set to this class ($I_E^{(2)} = BI$). However, the number of elements in the third set is less than the number of input CDMs ($|\{C2\}| = 1 < 3$). If we assume that the effectiveness for classes of the second CDM is greater than or equal to the threshold (th_{F_E}), or if the $C2$ class is relevant (it does have an attribute and an association), then this set is also mapped into a new class (CI) in the integrated CDM, and the $I_E^{(3)}$ indicator is set to this class ($I_E^{(3)} = CI$). Of course, the CI class inherits the name from the $C2$ class, i.e. $name(CI) = name(C2)$.

The number of elements in the fourth set is also less than the number of input CDMs ($|\{D3\}| = 1 < 3$). If we assume that the effectiveness for classes of the third CDM is less than the threshold (th_{F_E}), and if the $D3$ class is not relevant (it does not have any attributes nor associations), then this set is not mapped into a new class and the $I_E^{(4)}$ indicator is unset ($I_E^{(4)} = \emptyset$).

Considering all of the above, we have $I_E = [AI, BI, CI, \emptyset]$ and $E_O = \{AI, BI, CI\}$. Figure 4 shows the partial result of the integration task after the classes merging step.

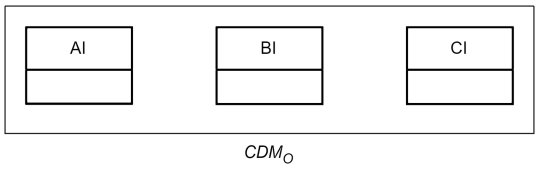


Fig. 4. Partial result of the integration task after the classes merging step

6.3. Attributes

The input in the attributes merging step (Algorithm 10) is the result of the attributes matching step (i.e. the list of sets of sets of matched attributes $M_A = [M_A^{(1)}, \ldots, M_A^{(m)}]$) and the result of the classes merging step (i.e. the list of class indicators $I_E = [I_E^{(1)}, \ldots, I_E^{(m)}]$). The input is also the list of sets of matched classes $M_E = [M_E^{(1)}, \ldots, M_E^{(m)}]$.

In this step, it is necessary to create attributes for the classes that are already created in the integrated CDM, i.e. if $I_E^{(i)}$ is a class ($I_E^{(i)} \neq \emptyset$) then it is necessary to create attributes for this class based on the set of sets of matched attributes $M_A^{(i)}$.

Each set of matched attributes $ma_j \in M_A^{(i)}$ is a candidate to be mapped into the attribute of the class $I_E^{(i)} \neq \emptyset$. The ma_j set is mapped into the attribute $at_o \in attrs(I_E^{(i)})$ if at least one of the following criteria is met:

- the number of elements in the ma_j set is equal to the number of input CDMs ($|ma_j| = n$), meaning this set contains an attribute from a class from each input CDM (this is possible only if $|M_E^{(i)}| = n$), or
- the ma_j set contains an attribute from a class from CDM with the effectiveness for attributes greater than or equal to the defined threshold ($at \in ma_j \wedge at \in attrs(e) \wedge e \in E_k \wedge F_{A_k} \geq th_{F_A}$), meaning the set contains an attribute of the class from input CDM with the acceptable effectiveness for attributes.

Algorithm 10 The attributes merging step

Require: $M_E = [M_E^{(1)}, \ldots, M_E^{(m)}]$
Require: $I_E = [I_E^{(1)}, \ldots, I_E^{(m)}]$
Require: $M_A = [M_A^{(1)}, \ldots, M_A^{(m)}]$
for all $I_E^{(i)} \in I_E$ **do**
  if $I_E^{(i)} \neq \emptyset$ **then**
    for all $ma_j \in M_A^{(i)}$ **do**
      $cr_met \leftarrow false$
      if $|ma_j| = n$ **then**
        $cr_met \leftarrow true$
      else
        for all $at \in ma_j$ **do**
          if $at \in attrs(e) \wedge e \in E_k \wedge F_{A_k} \geq th_{F_A}$ **then**
            $cr_met \leftarrow true$
          end if
        end for
      end if
      if cr_met **then**
        $at_o \leftarrow new_attribute()$; $at' \leftarrow max_{F_A}(ma_j)$
        $name(at_o) \leftarrow name(at')$; $type(at_o) \leftarrow type(at')$
        $attrs(I_E^{(i)}) \leftarrow attrs(I_E^{(i)}) \cup \{at_o\}$
      end if
    end for
  end if
end for

The name and the type of the attribute (if created) are inherited from the attribute of the class from the CDM with the highest effectiveness for attributes.

Example. In this example we need to create attributes for classes in the integrated CDM. The attributes of the class $I_E^{(i)} \neq \emptyset$ are created from the set of sets of matched attributes $M_A^{(i)}$. Each set in the $M_A^{(i)}$ set is a candidate to be mapped into an attribute (if the criteria is met).

First, we consider the class $I_E^{(1)} = AI$ and the set of sets of matched attributes $M_A^{(1)} = \{\{a11, a21, a31\}, \{a12, a22\}\}$. The first set in $M_A^{(1)}$ contains three elements, which is equal to the number of the input CDMs ($|\{a11, a21, a31\}| = 3$), thus this set is mapped into the $a1$ attribute of the AI class. If we assume that CDM_1 has the highest effectiveness for attributes, then the $a1$ attribute of the AI class inherits the name and type from the $a11$ attribute which is an attribute of the $A1$ class from CDM_1, i.e. $name(a1) = name(a11)$ and $type(a1) = type(a11)$.

The second set in $M_A^{(1)}$ contains two elements ($|\{a12, a22\}| = 2$). We already assumed that CDM_1 has the highest effectiveness for attributes. If we further assume that the effectiveness for attributes of CDM_1 is greater than or equal to the threshold (th_{F_A}), then this set is also mapped into an attribute ($a2$) of the AI class. The $a2$ attribute of the AI class inherits the name and type from the $a12$ attribute which is an attributes of the $A1$ class from CDM_1, i.e. $name(a2) = name(a12)$ and $type(a2) = type(a12)$.

Now, we come to the class $I_E^{(2)} = BI$ and the set of sets of matched attributes $M_A^{(2)} = \{\{b11, b21, b31\}\}$. The only set of matched attributes in the $M_A^{(2)}$ set is mapped into an attribute ($b1$) of the BI class because $|\{b11, b21, b31\}| = 3$.

Finally, we come to the class $I_E^{(3)} = CI$ and the set of sets of matched attributes $M_A^{(3)} = \{\{c21\}\}$. The $M_A^{(3)}$ set contains only one set of matched attributes ($\{c21\}$). This set has only one attribute ($|\{c21\}| = 1$). If we assume that the effectiveness for attributes of CDM_2 is also greater than or equal to the threshold (th_{F_A}), then the $\{c21\}$ set is also mapped into an attribute ($c1$) of the CI class. Also, the $c1$ attribute inherits the name and type from the $c21$ attribute, i.e. $name(c1) = name(c21)$ and $type(c1) = type(c21)$.

Considering all of the above, we have $attrs(AI) = \{a1, a2\}$, $attrs(BI) = \{b1\}$, and $attrs(CI) = \{c1\}$. The partial result of the integration task (after the attributes merging step) is shown in Fig. 5.

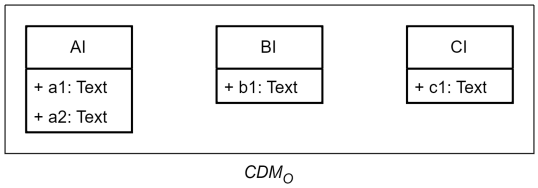


Fig. 5. Partial result of the integration task after the attributes merging step

6.4. Associations

The input in the associations merging step (Algorithm 11) is the result of the associations matching step (i.e. the list of sets of sets of matched associations M_R) and the result of the classes merging step (i.e. the list of class indicators I_E). The input is also the list of sets of matched classes M_E.

In this step, it is necessary to create associations between the classes that are already created in the integrated CDM, i.e. if $I_E^{(i)}$ and $I_E^{(j)}$ are classes ($I_E^{(i)} \neq \emptyset \wedge I_E^{(j)} \neq \emptyset$) then it is necessary to create associations between those classes based on the set of sets of matched associations $M_R^{(i)(j)}$.

Algorithm 11 The associations merging step

Require: $M_E = [M_E^{(1)}, \ldots, M_E^{(m)}]$
Require: $I_E = [I_E^{(1)}, \ldots, I_E^{(m)}]$
Require: $M_R = [M_R^{(1)(1)}, \ldots, M_R^{(1)(m)}, M_R^{(2)(2)}, \ldots, M_R^{(2)(m)}, \ldots, M_R^{(m)(m)}]$
Ensure: R_O
for all $M_R^{(i)(j)} \in M_R$ **do**
  if $I_E^{(i)} \neq \emptyset \wedge I_E^{(j)} \neq \emptyset$ **then**
    for all $mr_k \in M_R^{(i)(j)}$ **do**
      $cr_met \leftarrow false$
      if $|mr_k| = n$ **then**
        $cr_met \leftarrow true$
      else
        for all $r \in mr_k$ **do**
          if $r \in R_p \wedge F_{R_p} \geq th_{F_R}$ **then**
            $cr_met \leftarrow true$
          end if
        end for
      end if
      if cr_met **then**
        $r_o \leftarrow new_association()$; $r' \leftarrow max_{F_R}(mr_k)$
        $name(r_o) \leftarrow name(r')$
        $lower(source(r_o)) \leftarrow lower(source(r'))$
        $upper(source(r_o)) \leftarrow upper(source(r'))$
        $lower(target(r_o)) \leftarrow lower(target(r'))$
        $upper(target(r_o)) \leftarrow upper(target(r'))$
        if $type(source(r')) \in M_E^{(i)} \wedge type(target(r')) \in M_E^{(j)}$ **then**
          $type(source(r_o)) \leftarrow I_E^{(i)}$; $type(target(r_o)) \leftarrow I_E^{(j)}$
        else if $type(source(r')) \in M_E^{(j)} \wedge type(target(r')) \in M_E^{(i)}$ **then**
          $type(source(r_o)) \leftarrow I_E^{(j)}$; $type(target(r_o)) \leftarrow I_E^{(i)}$
        end if
        $R_O \leftarrow R_O \cup \{r_o\}$
      end if
    end for
  end if
end for

Each set of matched associations $mr_k \in M_R^{(i)(j)}$ is a candidate to be mapped into the association between the classes $I_E^{(i)} \neq \emptyset$ and $I_E^{(j)} \neq \emptyset$. The set $mr_k \in M_R^{(i)(j)}$ is mapped into the association $r_o \in R_O$ if at least one of the following criteria is met:

- the number of elements in the set $mr_k \in M_R^{(i)(j)}$ is equal to the number of input CDMs ($|mr_k| = n$), meaning this set contains an association from each input CDM (this is possible only if $|M_E^{(i)}| = |M_E^{(j)}| = n$), or
- the set $mr_k \in M_R^{(i)(j)}$ contains an association from the CDM with the effectiveness for associations greater than or equal to the defined threshold ($r \in mr_k \wedge r \in R_p \wedge F_{R_p} \geq th_{F_R}$), meaning the set contains an association from input CDM with the acceptable effectiveness for associations.

The name and the association end multiplicities (if created) are inherited from the association from CDM with the highest effectiveness for associations.

Example. Using the result of the associations matching step, in the associations merging step we need to create associations in the integrated CDM.

Let us recall the result of the classes merging step

$$I_E = [AI, BI, CI, \emptyset]\,.$$

Since $I_E^{(4)} = \emptyset$, then sets $M_R^{(1)(4)}$, $M_R^{(2)(4)}$, $M_R^{(3)(4)}$, and $M_R^{(4)(4)}$ are automatically eliminated. The sets $M_R^{(1)(1)}$, $M_R^{(2)(2)}$, $M_R^{(2)(3)}$, and $M_R^{(3)(3)}$ are considered, but those sets are empty (meaning there are no candidate sets to be mapped into associations).

Non-empty sets are $M_R^{(1)(2)}$ and $M_R^{(1)(3)}$. The $M_R^{(1)(2)}$ set contains two sets of matched associations between the corresponding pairs of classes from $M_E^{(1)}$ and $M_E^{(2)}$. The number of the elements in the first set is equal to the number of input CDMs ($|\{Z1, Z2, Z3\}| = 3$), thus this set is mapped into an association (ZI) between the classes AI and BI in the integrated CDM. Assuming that CDM_1 has the highest effectiveness for associations, the ZI association inherits the name and end multiplicities from the $Z1$ association. Further, if we assume that the $A1$ class is the *source* end of $Z1$ and the $B1$ class is the *target* end of $Z1$, then we have

$$\begin{gathered} name(ZI) = name(Z1), \quad source(ZI) = AI, \quad target(ZI) = BI, \\ lower(source(ZI)) = 1, \quad upper(source(ZI)) = 1, \\ lower(target(ZI)) = 0, \quad upper(target(ZI)) = \infty\,. \end{gathered}$$

The second set of the $M_R^{(1)(2)}$ set contains only one element ($|\{Y2\}| = 1$). If we assume that CDM_2 has the effectiveness for associations greater than or equal to the threshold (th_{F_R}), then this set is also mapped into an association (YI) between the classes AI and BI in the integrated CDM. The YI association inherits the name and end multiplicities from the $Y2$ association. If we assume that the $B2$ class is the *source* end of $Y2$ and the $A2$ class is the *target* end of $Y2$ (opposite from the case with $Z1$), then we have

$$\begin{gathered} name(YI) = name(Y2), \quad source(YI) = BI, \quad target(YI) = AI, \\ lower(source(YI)) = 0, \quad upper(source(YI)) = \infty, \\ lower(target(YI)) = 0, \quad upper(target(YI)) = 1\,. \end{gathered}$$

Finally, we come to the set $M_R^{(1)(3)} = \{\{X2\}\}$, which contains only one set of matched associations ($\{X2\}$). This set has only one association ($|\{X2\}| = 1$). We already assumed that the effectiveness for associations for CDM_2 is greater than or equal to the threshold (th_{F_R}), meaning that the $\{X2\}$ set is also mapped into an association (XI) in the integrated CDM. If we assume that the $A2$ class is the *source* end of $X2$ and the $C2$ class is the *target* end of $X2$, then we have

$$name(XI) = name(X2), \quad source(XI) = AI, \quad target(XI) = CI,$$
$$lower(source(XI)) = 1, \quad upper(source(XI)) = 1,$$
$$lower(target(XI)) = 0, \quad upper(target(XI)) = \infty\,.$$

Considering all of the above, we have $R_O = \{ZI, YI, XI\}$. The result of the integration task (after the associations merging step) is shown in Fig. 6.

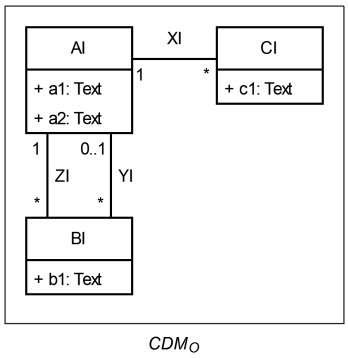


Fig. 6. Result of the integration task after the associations merging step

6.5. Generalizations

In this step it is necessary to create generalizations between the classes that are already created in the integrated CDM. The input in the generalizations merging step (Algorithm 12) is the result of the generalizations matching step (i.e. the list of sets of sets of matched generalizations M_G), the result of the classes merging step (i.e. the list of class indicators I_E), as well as the list of sets of matched classes M_E.

Each set of matched generalizations $mg_k \in M_G^{(i)(j)}$ is a candidate to be mapped into the generalization between the classes $I_E^{(i)} \neq \emptyset$ and $I_E^{(j)} \neq \emptyset$. The mg_k set is mapped into the generalization $g_o \in G_O$ if at least one of the following criteria is met:

- the number of elements in the mg_k set is equal to the number of input CDMs ($|mg_k| = n$), meaning this set contains a generalization from each input CDM (this is possible only if $|M_E^{(i)}| = |M_E^{(j)}| = n$), or
- the mg_k set contains a generalization from CDM with the effectiveness for generalizations greater than or equal to the defined threshold ($g \in mg_k \wedge g \in G_p \wedge F_{G_p} \geq th_{F_G}$), meaning the set contains a generalization from input CDM with the acceptable effectiveness for generalizations.

Algorithm 12 The generalizations merging step

Require: $M_E = [M_E^{(1)}, \ldots, M_E^{(m)}]$
Require: $I_E = [I_E^{(1)}, \ldots, I_E^{(m)}]$
Ensure: $M_G = [M_G^{(1)(2)}, \ldots, M_G^{(1)(m)}, M_G^{(2)(3)}, \ldots, M_G^{(2)(m)}, \ldots, M_G^{(m-1)(m)}]$
Ensure: G_O
1: **for all** $M_G^{(i)(j)} \in M_G$ **do**
2: **if** $I_E^{(i)} \neq \emptyset \wedge I_E^{(j)} \neq \emptyset$ **then**
3: **for all** $mg_k \in M_G^{(i)(j)}$ **do**
4: $cr_met \leftarrow false$
5: **if** $|mg_k| = n$ **then**
6: $cr_met \leftarrow true$
7: **else**
8: **for all** $g \in mg_k$ **do**
9: **if** $g \in G_p \wedge F_{G_p} \geq th_{F_G}$ **then**
10: $cr_met \leftarrow true$
11: **end if**
12: **end for**
13: **end if**
14: **if** cr_met **then**
15: $g_o \leftarrow new_generalization()$; $g' \leftarrow max_{F_G}(mg_k)$
16: **if** $specific(g') \in M_E^{(i)} \wedge general(g') \in M_E^{(j)}$ **then**
17: $specific(g_o) \leftarrow I_E^{(i)}$; $general(g_o) \leftarrow I_E^{(j)}$
18: **else if** $specific(g') \in M_E^{(j)} \wedge general(g') \in M_E^{(i)}$ **then**
19: $specific(g_o) \leftarrow I_E^{(j)}$; $general(g_o) \leftarrow I_E^{(i)}$
20: **end if**
21: $G_O \leftarrow G_O \cup \{g_o\}$
22: **end if**
23: **end for**
24: **end if**
25: **end for**

Due to the article length and the similarity of the generalizations merging step to the associations merging step (there can be only one generalization between two classes, while there can be more than one association between two classes) we do not provide an example for the generalization merging step.

6.6. Tidying up

In the final step of the schema integration process, we create surrogate attributes, i.e. attributes named *id*, for classes that do not already have an attribute named *id* and do not have generalizations. If a class already has an attribute named *id* and it has generalizations, then such attribute is removed from the class. Also, an operation named *PK* which represents the *primary key* specification is added to classes that do not have generalizations. Each *PK* operation has a single parameter named *id* (which corresponds to a surrogate attribute of the class).

7. Implemented Tool

This section presents the implemented tool named DBomnia[3]. DBomnia is the first online web-based tool providing the functionality of CDM derivation from heterogeneous source artifacts. Currently, DBomnia enables automatic CDM derivation from two different types of source artifacts: textual specifications and collections of BPMs. Figure 7 shows the architecture of the DBomnia tool, while Fig. 8 shows a screenshot of the tool in action.

The process of the CDM synthesis consists of four steps: (1) generation of the CDM from the source collection of BPMs, (2) generation of the CDM from the input textual specification, (3) integration of the generated CDMs, and (4) generation of the diagram layout for the integrated CDM.

In the first step, the source collection of BPMs is sent to AMADEOS. AMADEOS generates the corresponding CDM and responds with a JSON object containing the generated CDM, execution status, etc. In the second step, the input textual specification is sent to TexToData, which generates the corresponding CDM. TexToData also responds with a JSON object containing generated CDM, error messages (if any), etc.

In the third step, DBomnia integrates the CDMs generated in the first two steps, by applying the approach described in the previous sections. Then, in the fourth step, it sends the integrated CDM to the Layouter service, which generates the corresponding diagram layout. Layouter is the pre-existing service that provides the functionality of generating a diagram layout for the input UML class diagram (or in our case, CDM represented by the UML class diagram). Layouter responds with a file containing the generated layout.

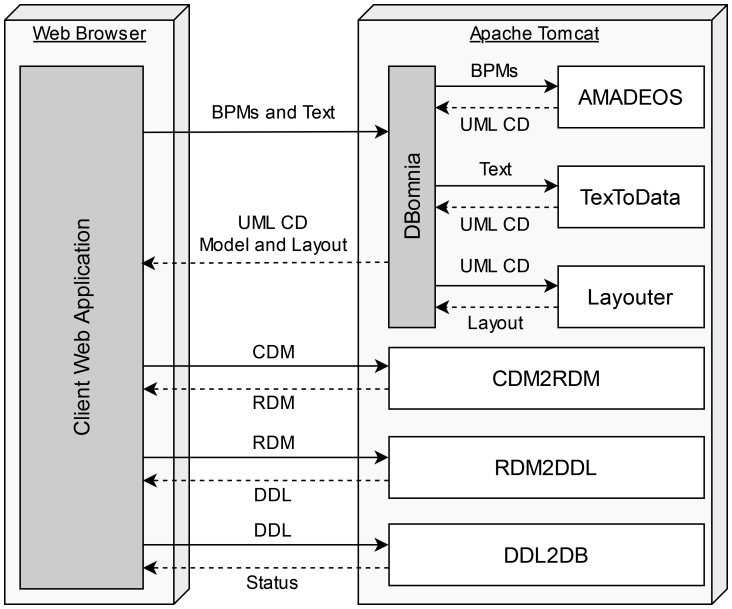


Fig. 7. DBomnia architecture

[3] http://m-lab.etf.unibl.org:8080/dbomnia

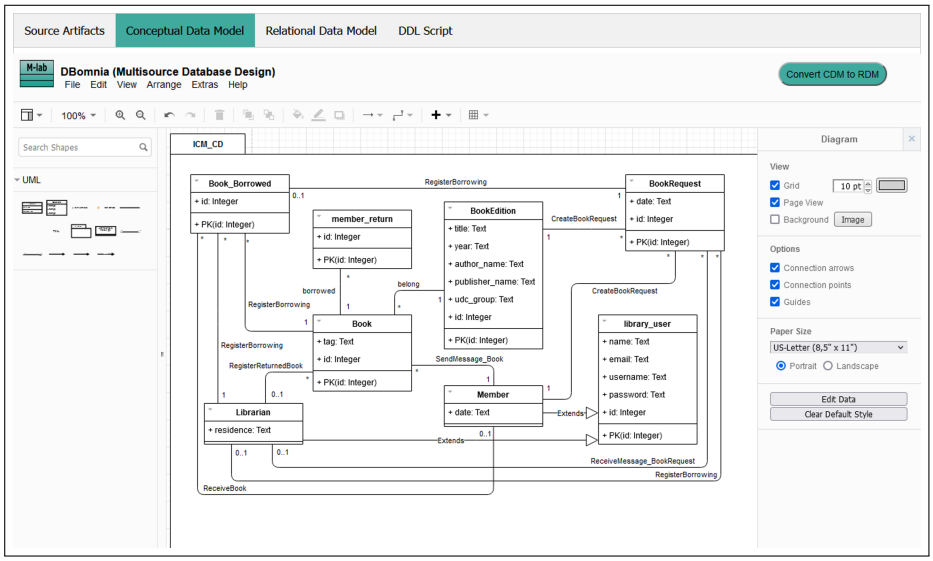


Fig. 8. Screenshot of DBomnia in action

Finally, DBomnia prepares and returns the response – a JSON object containing the integrated CDM, diagram layout, and status information.

The user interface of the client web application is separated into four tabs: *Source Artifacts*, *Conceptual Data Model*, *Relational Data Model*, and *DDL Script*. Tab *Source Artifacts* contains fields that allow users to input a textual specification and upload a collection of source BPMs. Upon the user's request (click on the *Generate CDM* button), all source artifacts are sent to DBomnia. When the client web application receives the JSON response, it visualizes[4] the class diagram in the browser (in the *Conceptual Data Model* tab). The visualized diagram is editable, so users can additionally improve it.

DBomnia also supports all the subsequent steps of forward database engineering, from CDM to the target physical database, by employing CDM2RDM, RDM2DDL, and DDL2DB services, as described in [53].

8. Evaluation

This section presents the evaluation of the approach and the tool on three case studies. One case study-based evaluation is described in detail, while only the most significant results are provided for the other two.

8.1. Environment Setup

This subsection provides concrete values of the estimated measures for input CDMs, as well as empirically determined thresholds and weights used by the tool.

Table 1 shows estimated measures for input CDMs generated by AMADEOS and estimated measures for input CDMs generated by TexToData. The measures for input

[4] Implementation is based on the mxGraph library (https://jgraph.github.io/mxgraph/)

Table 1. Values of estimated measures for input CDMs

Tool	F_E	F_A	F_R	F_G
AMADEOS	0.76	0.00	0.41	0.00
TexToData	0.70	0.72	0.30	1.00

CDMs generated by AMADEOS are experimentally determined [11], while measures for input CDMs generated by TexToData are determined from case study-based evaluation. AMADEOS is able to generate a highly complete CDM structure, but generates only *id* attribute which represents the surrogate key in each entity type. TexToData is able to generate CDMs of less complete and less correct structure but with a more complete set of attributes in each entity type. Also, AMADEOS does not generate generalizations.

Table 2 shows threshold and weight values. To eliminate only minor typographical errors, the Levenshtein distance threshold is set to 1. The weight of attributes in similarity calculation for classes in the current environment (with AMADEOS and TexToData) does not play any role, because AMADEOS does not generate attributes (except for the surrogate key attribute *id* ignored in the integration task). Currently, for the same reason, the value of the similarity threshold for attributes is irrelevant. To keep the attributes generated by TexToData, the effectiveness threshold for attributes is lower than the effectiveness measure for attributes for TexToData.

Although AMADEOS generates associations with lower precision than classes, the CDM generated by AMADEOS is expected to be more precise and more complete than the CDM generated by TexToData regarding associations. To increase the completeness of the associations in the integrated CDM, the environment is set to keep unmatched associations from both input CDMs. We consider two associations a match if their cardinalities (upper values of the respective member ends) are equal or if their name similarity is very high. We only compare the cardinalities because we expect that the CDM generated by AMADEOS will be more precise than the CDM generated by TexToData regarding the participation constraints.

On the other hand, in order to reduce the number of excessive classes, the environment is set up to keep only relevant unmatched classes, i.e. the effectiveness threshold for classes is greater than the effectiveness measure for classes for AMADEOS and TexToData. Currently, an unmatched class is considered to be relevant if it has at least one attribute or at least one association or if it participates in at least one generalization (the relevance threshold for classes is set to one, while the weight of attributes and weight of associations and generalizations is also set to one).

Since AMADEOS does not generate generalizations, we want to keep generalizations generated by TexToData (the effectiveness threshold for generalizations is less than the effectiveness measure for generalizations for TexToData).

Note that we provided concrete values for thresholds and weights. Since currently there can be only two input CDMs (CDM generated by AMADEOS and CDM generated by TexToData), other values can be set up to achieve the same result. For example, if we want to keep only relevant unmatched classes from both CDMs, then the value of the effectiveness threshold for classes can be any value in the segment $(0.76, 1]$, i.e. any value greater than the maximum of the effectiveness measure for classes for AMADEOS and TexToData.

Table 2. Values for thresholds and weights

Symbol	Description	Value
LD	Levenshtein distance threshold	1
W_A	Weight of attributes in similarity calculation for classes	0.00
W_{card}	Weight of cardinalities in similarity calculation for associations	0.75
W_{pc}	Weight of participation constraints in similarity calculation for associations	0.00
th_{S_E}	Similarity threshold for classes	0.25
th_{S_A}	Similarity threshold for attributes	1.00
th_{S_R}	Similarity threshold for associations	0.20
th_{S_G}	Similarity threshold for generalizations	1.00
th_{F_E}	Effectiveness threshold for classes	0.80
th_{F_A}	Effectiveness threshold for attributes	0.50
th_{F_R}	Effectiveness threshold for associations	0.10
th_{F_G}	Effectiveness threshold for generalizations	0.10
th_{R_E}	Relevance threshold for classes	1.00
RW_A	Weight of attributes in relevance calculation for classes	1.00
RW_R	Weight of associations and generalizations in relevance calculation for classes	1.00

8.2. Case Study: *Online Library*

We evaluated the approach through the assessment of the CDM derived from a set of heterogeneous source artifacts against the CDMs that are derived from sources of one single type. Firstly, we prepared a sample source set of BPMs and a textual description of the *Online Library*, as well as the reference (target) CDM (Fig. 9). Secondly, we evaluated generated CDMs (CDM generated by AMADEOS from source BPMs, CDM generated by TexToData from textual specification, and CDM generated by DBomnia) against the reference CDM. Finally, we compared the results obtained by DBomnia against those obtained by AMADEOS and TexToData.

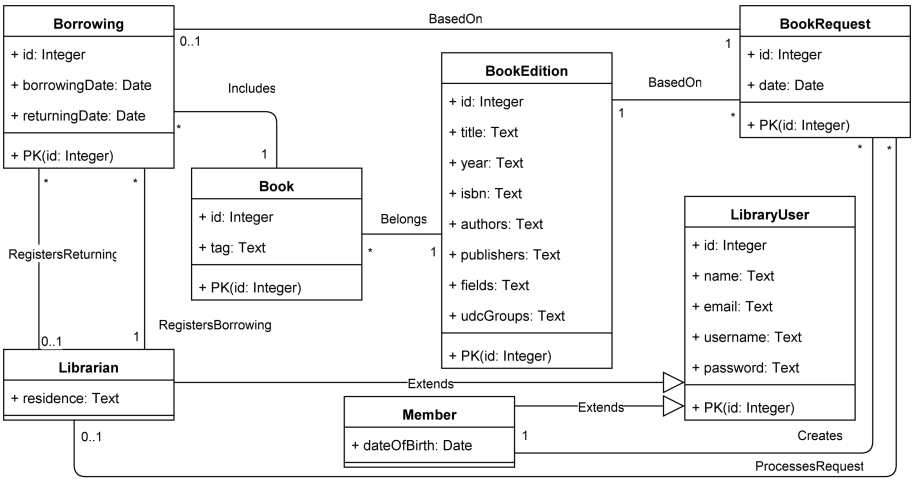


Fig. 9. Reference CDM of the *Online Library*

AMADEOS. The sample source set of BPMs (Fig. 10) represents two main processes in the *Online Library – Book Borrowing* and *Book Returning*. In the first process: a member creates a borrowing request for a copy of some book edition; a librarian registers the borrowing and issues a book. In the second process: the member returns the borrowed book, and the librarian registers the returned book.

Figure 11 shows the CDM derived from the source set of BPMs. When compared against the reference CDM (Fig. 9):

1. all classes, four *id* attributes, and six associations (*BookEdition-CreateBookRequest-BookRequest*, *BookRequest-RegisterBorrowing-Book_Borrowed*, *Member-CreateBookRequest-BookRequest*, *Librarian-RegisterBorrowing-BookRequest*, *Librarian-RegisteBorrowing-Book_Borrowed*, *Book-RegisterBorrowing-Book_Borrowed*) could be evaluated as correctly generated,
2. two *id* attributes and four associations (*Librarian-RegisterReturnedBook-Book*, *Librarian-ReceiveMessage_BookRequest-BookRequest*, *Member-SendMessage_Book-Book*, *Member-ReceiveBook-Book_Borrowed*) could be evaluated as surplus, and

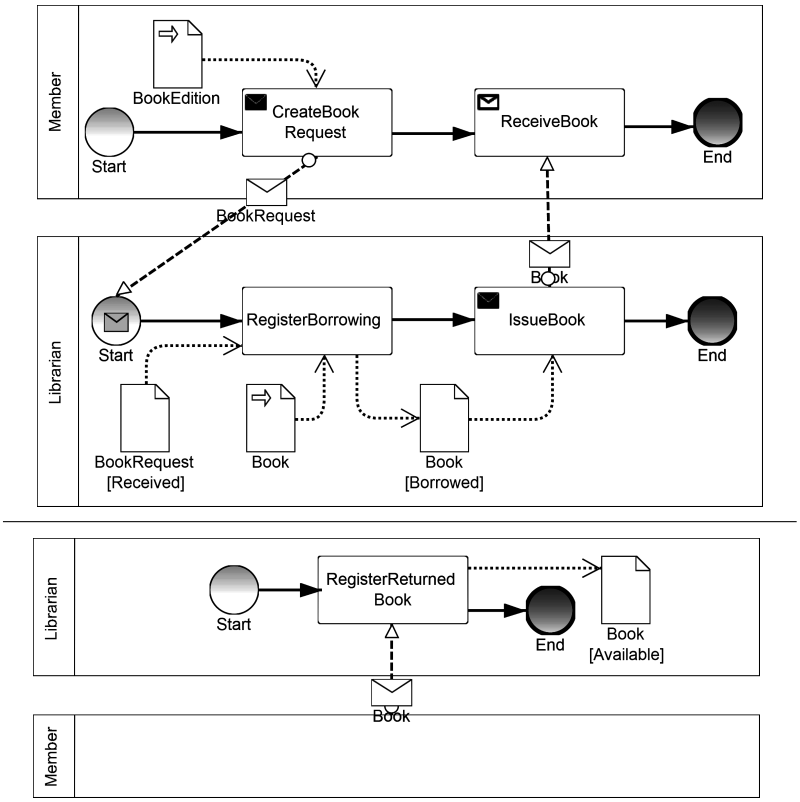


Fig. 10. Sample set of BPMs: *Book Borrowing* (top), *Book Returning* (bottom)

3. one class (*LibraryUser*), 18 attributes, both generalizations, and two associations (*BookEdition*↔*Book*, and *Librarian*↔*Book Borrowed* that represents the returning of the book) are missing in the generated CDM.

Table 3 shows the results of quantitative evaluation based on the previous discussion.

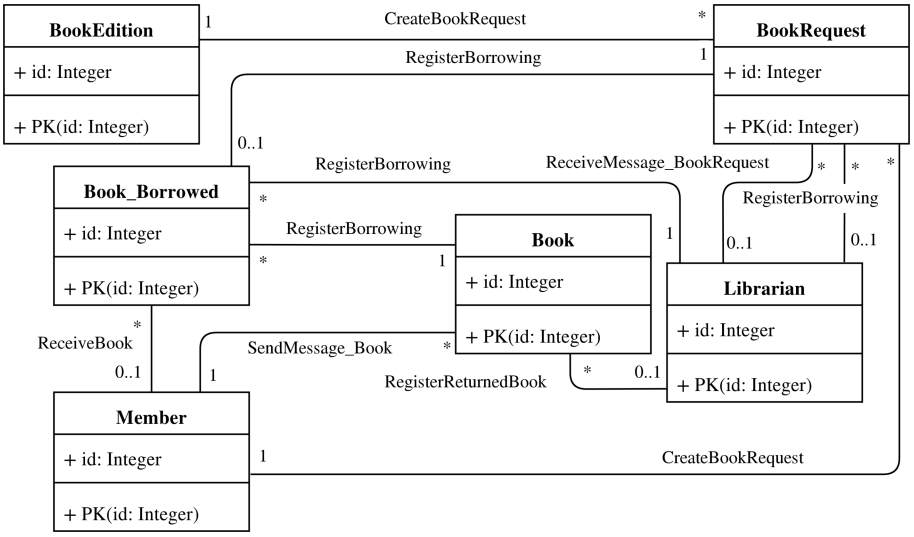


Fig. 11. CDM generated by AMADEOS based on the source set of BPMs

Table 3. Quantitative evaluation of the CDM generated by AMADEOS

Concept	N_c	N_m	N_w	R	P	F
Classes	6	1	0	0.857	1	0.923
Attributes	4	18	2	0.182	0.667	0.286
Associations	6	2	4	0.75	0.6	0.667
Generalizations	0	2	0	0	–	–

TexToData. Figure 12 shows the textual description of the *Online Library*, while Fig. 13 shows the CDM derived from the given textual description. The automatically generated CDM has ten classes, 13 attributes, two generalizations, and only four associations. When compared against the reference CDM (Fig. 9):

1. seven classes (*library_user*, *librarian*, *member*, *borrowing_request*, *book*, *book_edition*, and *borrowing*) and 11 attributes, both generalizations, and three associations (*book-belong-book_edition*, *book_edition-belongs-borrowing_request*, *member-creates-borrowing_request*) could be evaluated as correctly generated,
2. one association (*book-borrowed-member_return*) and three classes (*issue*, *register*, *member_return*) could be evaluated as surplus,
3. two attributes (*date* in the *member* class and *date* in the *borrowing_request* class) could be evaluated as incorrectly generated (incorrect type), and
4. five associations, and nine attributes are missing in the generated CDM.

Table 4 shows the results of quantitative evaluation based on the previous discussion.

Library users are librarians or members. Library user has name, email, username, and password. Librarian has residence. Member has date of birth. Book edition has title, year, ISBN, authors names, publishers names, fields, and UDC groups. Book has tag. Books belong to book edition. Member creates borrowing requests. Borrowing request has date. Borrowing requests belongs to book edition. Librarian registers borrowings and issues books. Member returns borrowed book. Librarian registers returned book.

Fig. 12. Sample textual description of the *Online Library*

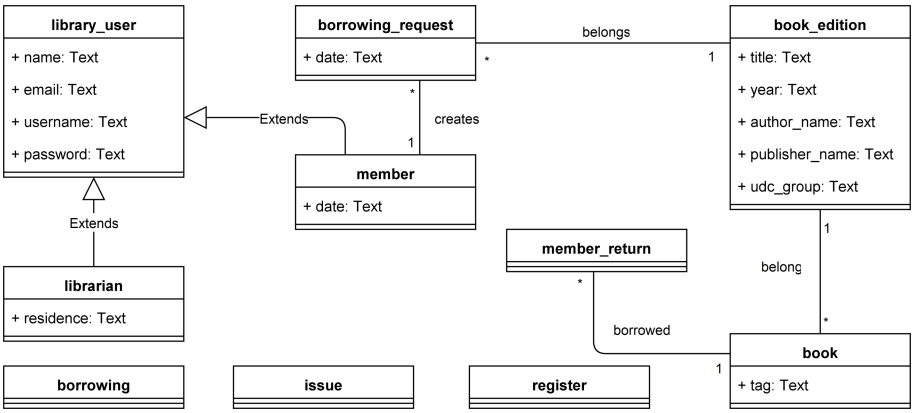


Fig. 13. CDM generated by TexToData based on the sample textual specification

Table 4. Quantitative evaluation of the CDM generated by TexToData

Concept	N_c	N_m	N_w	R	P	F
Classes	7	0	3	1	0.7	0.824
Attributes	11	9	2	0.55	0.846	0.667
Associations	3	5	1	0.375	0.75	0.5
Generalizations	2	0	0	1	1	1

DBomnia. When we use the sample set of BPMs (Fig. 10) and sample textual description (Fig. 12) with DBomnia, we obtain the CDM (Fig. 14) containing eight classes, 19 attributes, 12 associations, and two generalizations. When compared against the reference CDM (Fig. 9):

1. seven classes, 16 attributes, both generalizations, and seven associations could be evaluated as correct,
2. one class (*member_return*) and corresponding *id* attribute, and five associations (*Librarian-RegisterReturnedBook-Book*, *Member-SendMessage_Book-Book*, *Book-borrowed-member_return*, *Librarian-ReceiveMessage_BookRequest-BookRequest*, *Member-ReceiveBook-Book_Borrowed*) could be evaluated as surplus,
3. two attributes (*date* in the *Member* class and *date* in the *BookRequest* class) could be evaluated as incorrectly generated (incorrect type), and

4. only one association (*Librarian↔Book_Borrowed*) and only four attributes (*ISBN* and *fields* in the *BookEdition* class, and *borrowing_date* and *returning_date* in the *Book_Borrowed* class) are missing in the generated CDM.

Table 5 shows the results of quantitative evaluation based on the previous discussion.

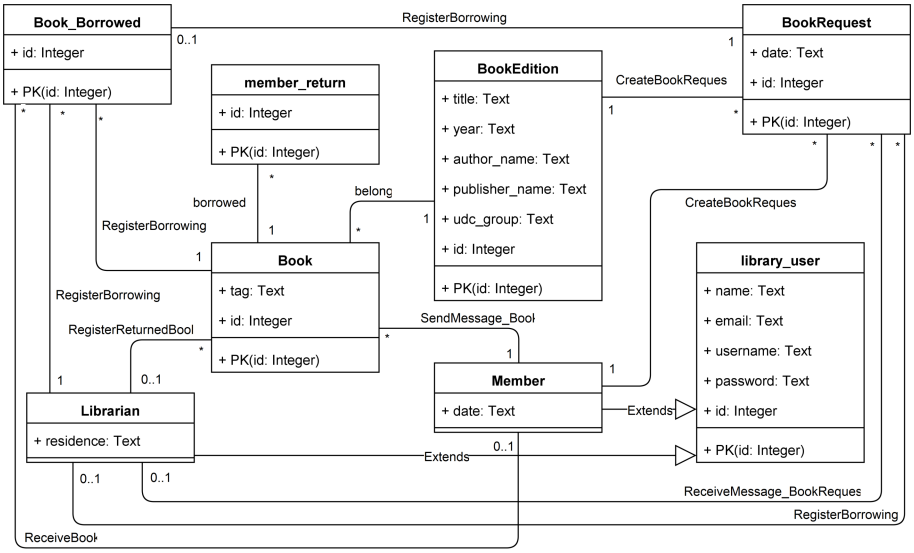


Fig. 14. CDM of the *Online Library* generated by DBomnia

Table 5. Quantitative evaluation of the CDM generated by DBomnia

Concept	N_c	N_m	N_w	R	P	F
Classes	7	0	1	1	0.875	0.933
Attributes	16	4	3	0.8	0.842	0.821
Associations	7	1	5	0.875	0.583	0.7
Generalizations	2	0	0	1	1	1

Results discussion. Table 6 shows the overview of the quantitative evaluation (i.e. the effectiveness measure) for AMADEOS, TexToData, and DBomnia. The presented results show that DBomnia still does not generate 100% complete nor 100% correct target model. However, the effectiveness measure for each CDM concept generated by DBomnia is greater than or equal to the higher value of the effectiveness measure for the corresponding concept generated by AMADEOS and TexToData. Although the automatically generated CDM has one surplus class, it is easy to spot and simply delete such (surplus) classes – unlike the correctly generated classes, such classes (probably) will not contain attributes or will not have associations with other classes. We believe that it is easier to delete a surplus class than to add a new class when it's missing in the generated CDM.

The results show that the automatic derivation of the CDM from a set of heterogeneous source artifacts is more effective than each independent automatic derivation of the CDM based on sources of one single type only. Hence, this case study-based evaluation of the approach and implemented tool confirm the research goal.

Table 6. Overview of the quantitative evaluation for the *Online Library*

Effectiveness measure	AMADEOS	TexToData	DBomnia
F_E	0.923	0.824	0.933
F_A	0.286	0.667	0.821
F_R	0.667	0.5	0.7
F_G	–	1	1

8.3. Case Studies: *Admission Exam* and *Student Jobs*

In order to conduct a more extensive evaluation, compared to the illustrative example from the previous subsection, we run two more case study-based evaluations as follows. Firstly, two authors of the paper had to choose an arbitrary domain and create a set of at least ten BPMs and a textual specification (at least 100 words) for the chosen domain. Secondly, they had to (i) exchange created artifacts, (ii) manually design a reference CDM based on received source artifacts, and (iii) evaluate CDMs generated by AMADEOS, TexToData, and DBomnia against the manually designed reference CDM. Finally, the third author had to inspect created artifacts and reference CDMs, as well as the entire evaluation process.

The chosen domains were *Admission Exam* and *Student Jobs*. Source artifacts[5] consist of 12 BPMs and 112-word textual specification for *Admission Exam*, and ten BPMs and 117-word textual specification for *Student Jobs*. Reference CDM[6] for *Admission Exam* contains 16 classes, 41 attributes, 25 associations, and three generalizations, while reference CDM for *Student Jobs* contains 15 classes, 37 attributes, 18 associations, and three generalizations.

Table 7 shows the results of quantitative evaluation[7] for both domains and all three tools. Each row of the table contains values for one concept type (classes – *E*, attributes – *A*, associations – *R*, or generalizations – *G*) of the generated CDM.

Results discussion. Table 8 shows the overview (only F-scores) of the quantitative evaluation, while Fig. 15 shows the comparison of the F-score values. The obtained results confirm the results obtained for the illustrative example, i.e. the effectiveness measure for each CDM concept generated by DBomnia is greater than or equal to the higher value of the effectiveness measure for the corresponding concept generated by AMADEOS and TexToData. The proposed approach and implemented tool (DBomnia) exploit the best features of other tools (as already mentioned, AMADEOS is able to generate a highly complete CDM structure, while TexToData is able to generate CDMs of less complete and less correct structure but with a more complete set of attributes in each entity type) to achieve more effective derivation of the CDMs. Therefore, the evaluation of the approach and implemented tool on three case-studies confirm the research goal. Since the results of the case study-based evaluation are positive, we are now directed toward extensive evaluation of the approach and implemented tool.

[5] Source artifacts for both domains are available at: https://gitlab.com/m-lab-research/ComSIS-2024/-/tree/main/01-SourceArtifacts

[6] Reference CDMs for both domains are available at: https://gitlab.com/m-lab-research/ComSIS-2024/-/tree/main/02-ReferenceCDMs

[7] CDMs generated by all three tools for both domains are available at: https://gitlab.com/m-lab-research/ComSIS-2024/-/tree/main/03-GeneratedCDMs

Table 7. Quantitative evaluation for *Admission Exam* and *Student Jobs*

Tool		*Admission Exam*						*Student Jobs*					
		N_c	N_m	N_w	R	P	F	N_c	N_m	N_w	R	P	F
AMADEOS	*E*	16	0	3	1	0.842	0.914	15	0	0	1	1	1
	A	13	28	6	0.317	0.684	0.433	12	25	3	0.324	0.8	0.462
	R	24	1	13	0.96	0.649	0.774	15	1	14	0.938	0.517	0.667
	G	0	3	0	0	–	–	0	3	0	0	–	–
TexToData	*E*	14	2	2	0.875	0.875	0.875	11	4	2	0.733	0.846	0.786
	A	25	16	0	0.61	1	0.758	16	21	1	0.432	0.941	0.593
	R	7	18	1	0.28	0.875	0.424	7	11	5	0.389	0.583	0.467
	G	3	0	0	1	1	1	3	0	0	1	1	1
DBomnia	*E*	16	0	3	1	0.842	0.914	15	0	0	1	1	1
	A	38	3	3	0.927	0.927	0.927	28	9	1	0.757	0.966	0.848
	R	25	0	14	1	0.641	0.781	16	0	15	1	0.516	0.681
	G	3	0	0	1	1	1	3	0	0	1	1	1

Table 8. Summary of the evaluation results

Measure	*Admission Exam*			*Student Jobs*		
	AMADEOS	TexToData	DBomnia	AMADEOS	TexToData	DBomnia
F_E	0.914	0.875	**0.914**	1	0.786	**1**
F_A	0.433	0.758	**0.927**	0.462	0.593	**0.848**
F_R	0.774	0.424	**0.781**	0.667	0.467	**0.681**
F_G	–	1	**1**	–	1	**1**

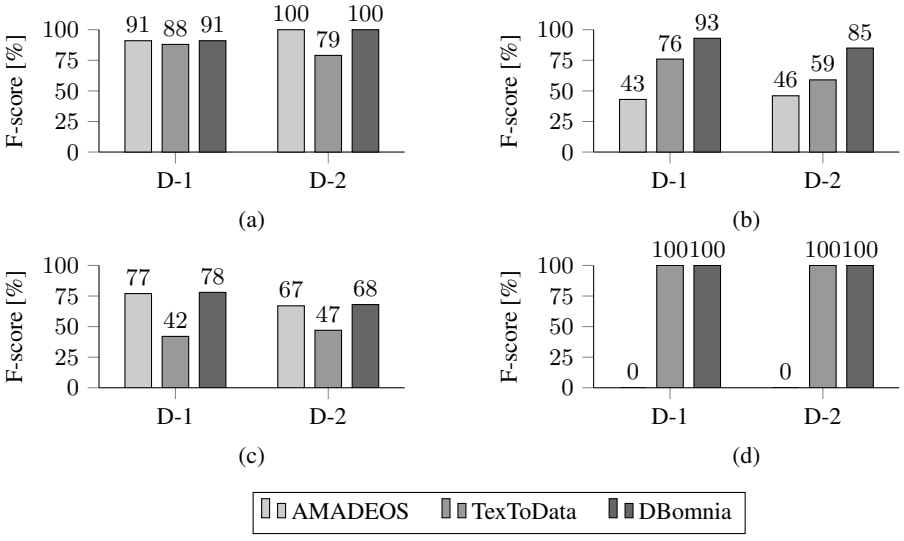


Fig. 15. Comparison of (a) F_E, (b) F_A, (c) F_R, and (d) F_G measures by tool per domain (D-1 – *Admission Exam*, D-2 – *Student Jobs*)

9. Conclusion

In this article we proposed an approach to the automatic CDM derivation from heterogeneous source artifacts. The approach is based on the integration of CDMs that are derived from source artifacts of one single type by already existing tools, whereby the main characteristic of these CDMs is uncertainty. Uncertainty of CDMs automatically derived from specific source artifacts is expressed and managed through the effectiveness measure of generation of specific concepts of the input CDMs.

The approach is implemented by the DBomnia tool – the first online web-based tool enabling automatic CDM derivation from a heterogeneous set of source artifacts, whereby the currently supported source artifacts are BPMs and textual specifications. DBomnia employs other tools (AMADEOS and TexToData) to generate CDMs from specific source artifacts (AMADEOS derives CDM from BPMs, while TexToData derives CDM from textual specifications) and then integrates the generated CDMs into a single unified CDM.

The approach and implemented tool were evaluated in three case studies. As expected, evaluation proves that the implemented approach enables effective automatic derivation of the conceptual database model from a set of heterogeneous source artifacts. Automatic derivation of the conceptual database model from a set of heterogeneous source artifacts is more effective than each independent automatic CDM derivation from sources of one single type only.

In line with the long-term research goals, our future work will focus on the approach and tool improvements and will include: work on semantic matching and combination of multiple languages in one execution of the automatic CDM derivation, further improvements of the specific CDM generators, inclusion of other types of source artifacts, and thorough validation and verification. Our intention is also to evaluate the approach with more complex real collections of BPMs and textual specifications.

Although the great advantage of the implemented multilingual tool is a large number of supported languages, the semantic analysis (e.g. analysis of the acronyms, synonyms, etc.) constitutes a great challenge and it is not considered in this paper. Taking that into consideration, as well as the effort of increasing the effectiveness of the overall CDM derivation process, we will focus on exploring the possibility of leveraging the ML techniques in our approach.

References

1. Adamson, G.W., Boreham, J.: The use of an association measure based on character structure to identify semantically related pairs of words and document titles. Information Storage and Retrieval 10(7), 253–260 (1974)
2. Anam, S., Kim, Y.S., Kang, B., Liu, Q.: Designing a knowledge-based schema matching system for schema mapping. Australasian Data Mining Conference (1 2015), https://figshare.utas.edu.au/articles/conference_contribution/Designing_a_knowledge-based_schema_matching_system_for_schema_mapping/23095379
3. Axler, S.: Linear Algebra Done Right. Springer Cham (2023)
4. Banjac, G., Brdjanin, D., Banjac, D.: Towards automatic conceptual database design based on heterogeneous source artifacts. In: Abelló, A. et al. (ed.) New Trends in Database and Information Systems. pp. 487–498. Springer Nature Switzerland, Cham (2023)
5. Batini, C., Lenzerini, M., Navathe, S.B.: A comparative analysis of methodologies for database schema integration. ACM Comput. Surv. 18(4), 323–364 (1986)

6. Berlin, J., Motro, A.: Database schema matching using machine learning with feature selection. In: Pidduck, A.B., Ozsu, M.T., Mylopoulos, J., Woo, C.C. (eds.) Advanced Information Systems Engineering. pp. 452–466. Springer Berlin Heidelberg, Berlin, Heidelberg (2002)
7. Bernstein, P., Madhavan, J., Rahm, E.: Generic schema matching, ten years later. Proc. VLDB Endow. 4(11), 695–701 (2011)
8. Bernstein, P.A., Halevy, A.Y., Pottinger, R.A.: A vision for management of complex models. SIGMOD Rec. 29(4), 55–63 (dec 2000), https://doi.org/10.1145/369275.369289
9. Brdjanin, D., Maric, S.: An Approach to Automated Conceptual Database Design Based on the UML Activity Diagram. Computer Science and Information Systems 9(1), 249–283 (2012)
10. Brdjanin, D., Maric, S.: Model-driven Techniques for Data Model Synthesis. Electronics 17(2), 130–136 (2013)
11. Brdjanin, D., Vukotic, A., Banjac, D., Banjac, G., Maric, S.: Automatic derivation of the initial conceptual database model from a set of business process models. Computer Science and Information Systems 19(1), 455–493 (2022)
12. Brdjanin, D., Banjac, G., Babic, N., Golubovic, N.: Towards the speech-driven database design. In: Proc. of TELFOR 2022. pp. 1–4. IEEE (2022)
13. Brdjanin, D., Grumic, M., Banjac, G., Miscevic, M., Dujlovic, I., Kelec, A., Obradovic, N., Banjac, D., Volas, D., Maric, S.: Towards an online multilingual tool for automated conceptual database design. In: Braubach, L., et al. (eds.) Intelligent Distributed Computing XV. pp. 144–153. Springer (2023)
14. Bulygin, L.: Combining lexical and semantic similarity measures with machine learning approach for ontology and schema matching problem. In: Proceedings of the XX International Conference "Data Analytics and Management in Data Intensive Domains"(DAMDID/RCDL'2018). pp. 245–249 (2018)
15. Chen, P.: English sentence structure and entity-relationship diagrams. Information Sciences 29(2-3), 127–149 (1983)
16. Choobineh, J., Mannino, M., Nunamaker, J., Konsynsky, B.: An expert database design system based on analysis of forms. IEEE Transaction on Software Engineering 14(2), 242–253 (1988)
17. Choobineh, J., Lo, A.W.: CABSYDD: Case-based system for database design. Journal of Management Information Systems 21(3), 281–314 (2004)
18. Cohen, W.W., Ravikumar, P., Fienberg, S.: A comparison of string metrics for matching names and records. In: Proc. of KDD 2003, Workshop on Data Cleaning, Record Linkage, and Object Consolidation (2003)
19. Damerau, F.J.: A technique for computer detection and correction of spelling errors. Commun. ACM 7(3), 171–176 (1964)
20. Date, C.: An Introduction to Database Systems, 8th edn. Addison-Wesley (2003)
21. Doan, A., Domingos, P., Halevy, A.Y.: Reconciling schemas of disparate data sources: a machine-learning approach. SIGMOD Rec. 30(2), 509–520 (may 2001), https://doi.org/10.1145/376284.375731
22. Duchateau, F., Coletta, R., Bellahsene, Z., Miller, R.J.: (not) yet another matcher. In: Proceedings of the 18th ACM Conference on Information and Knowledge Management. p. 1537–1540. CIKM '09, Association for Computing Machinery, New York, NY, USA (2009), https://doi.org/10.1145/1645953.1646165
23. Friedman, C., Sideli, R.: Tolerating spelling errors during patient validation. Computers and Biomedical Research 25(5), 486–509 (1992)
24. Gali, N., Mariescu-Istodor, R., Hostettler, D., Fränti, P.: Framework for syntactic string similarity measures. Expert Systems with Applications 129, 169–185 (2019)
25. Gotoh, O.: An improved algorithm for matching biological sequences. Journal of Molecular Biology 162(3), 705–708 (1982)
26. Hamming, R.W.: Error detecting and error correcting codes. The Bell System Technical Journal 29(2), 147–160 (1950)

27. Harmain, H., Gaizauskas, R.: CM-Builder: A Natural Language-Based CASE Tool for Object-Oriented Analysis. Automated Software Eng. 10(2), 157–181 (2003)
28. Hartmann, S., Link, S.: English sentence structures and EER modeling. In: Proc. of the 4th Asia-Pacific conf. on conceptual modelling – Vol. 67. pp. 27–35 (2007)
29. Jaro, M.A.: Advances in record-linkage methodology as applied to matching the 1985 census of tampa, florida. Journal of the American Statistical Association 84(406), 414–420 (1989)
30. Jouault, F., Allilaire, F., Bezivin, J., Kurtev, I.: ATL: A model transformation tool. Science of Computer Programming 72(1-2), 31–39 (2008)
31. Kriouile, A., Addamssiri, N., Gadi, T.: An MDA Method for Automatic Transformation of Models from CIM to PIM. American J. of Software Eng. and Applications 4(1), 1–14 (2015)
32. Levenshtein, I.V.: Binary codes capable of correcting deletions, insertions and reversals. Soviet Physics Doklady 10(8), 707–710 (1966)
33. Lukovic, I., Mogin, P., Pavicevic, J., Ristic, S.: An approach to developing complex database schemas using form types. Software: Practice & Experience 37(15), 1621–1656 (2007)
34. Madhavan, J., Bernstein, P., Doan, A., Halevy, A.: Corpus-based schema matching. In: 21st International Conference on Data Engineering (ICDE'05). pp. 57–68 (2005)
35. Madhavan, J., Bernstein, P., Rahm, E.: Generic schema matching with cupid. In: Proc. of VLDB 2001. pp. 49–58. Morgan Kaufmann (2001)
36. Magnani, M., Rizopoulos, N., Mc.Brien, P., Montesi, D.: Schema integration based on uncertain semantic mappings. In: Conceptual Modeling – ER 2005. pp. 31–46. Springer (2005)
37. Malakasiotis, P., Androutsopoulos, I.: Learning textual entailment using svms and string similarity measures. In: Proc. of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing. p. 42–47. Association for Computational Linguistics, USA (2007)
38. Needleman, S.B., Wunsch, C.D.: A general method applicable to the search for similarities in the amino acid sequence of two proteins. Journal of Molecular Biology 48(3), 443–453 (1970)
39. Nikiforova, O., Gusarovs, K., Gorbiks, O., Pavlova, N.: BrainTool: A tool for generation of the UML class diagrams. In: Proc. of ICSEA 2012. pp. 60–69. IARIA (2012)
40. Omar, N., Hanna, P., McKevitt, P.: Heuristics-based entity-relationship modelling through natural language processing. In: Proc. of AICS 2004. pp. 302–313 (2004)
41. OMG: Unified Modeling Language (OMG UML), v2.5. OMG (2015)
42. Overmyer, S.P., Benoit, L., Owen, R.: Conceptual modeling through linguistic analysis using LIDA. In: Proc. of ICSE 2001. pp. 401–410. IEEE (2001)
43. Pottinger, R.A., Bernstein, P.A.: Merging models based on given correspondences. In: Proceedings 2003 VLDB Conference, pp. 862–873. Morgan Kaufmann, San Francisco (2003), https://www.sciencedirect.com/science/article/pii/B9780127224428500811
44. Purao, S.: APSARA: A tool to automate system design via intelligent pattern retrieval and synthesis. SIGMIS Database 29(4), 45–57 (1998)
45. Rezaei, M., Fränti, P.: Matching similarity for keyword-based clustering. In: Fränti, P. et al. (ed.) Structural, Syntactic, and Statistical Pattern Recognition. pp. 193–202. Springer (2014)
46. Rodrigues, D., da Silva, A., Rodrigues, R., dos Santos, E.: Using active learning techniques for improving database schema matching methods. In: 2015 International Joint Conference on Neural Networks (IJCNN). pp. 1–8 (2015)
47. Rodriguez, A., Garcia-Rodriguez de Guzman, I., Fernandez-Medina, E., Piattini, M.: Semi-formal transformation of secure business processes into analysis class and use case models: An MDA approach. Information and Software Technology 52(9), 945–971 (2010)
48. Sahay, T., Mehta, A., Jadon, S.: Schema matching using machine learning. In: 2020 7th International Conference on Signal Processing and Integrated Networks (SPIN). pp. 359–366 (2020)
49. Sheetrit, E., Brief, M., Mishaeli, M., Elisha, O.: Rematch: Retrieval enhanced schema matching with llms (2024), https://arxiv.org/abs/2403.01567

50. Shraga, R., Gal, A.: Powarematch: A quality-aware deep learning approach to improve human schema matching. J. Data and Information Quality 14(3) (may 2022), https://doi.org/10.1145/3483423
51. Shraga, R., Gal, A., Roitman, H.: Adnev: cross-domain schema matching using deep similarity matrix adjustment and evaluation. Proc. VLDB Endow. 13(9), 1401–1415 (may 2020), https://doi.org/10.14778/3397230.3397237
52. Smith, T.F., Waterman, M.S.: Identification of common molecular subsequences. Journal of Molecular Biology 147(1), 195–197 (1981)
53. Spasic, Z., Vukotic, A., Brdjanin, D., Banjac, D., Banjac, G.: UML-based forward database engineering. In: Proc. of INFOTEH 2023. pp. 1–6. IEEE (2023)
54. Sugumaran, V., Storey, V.C.: Ontologies for conceptual modeling: their creation, use, and management. Data & Knowledge Engineering 42(3), 251–271 (2002)
55. Tan, H.B.K., Yang, Y., Blan, L.: Systematic Transformation of functional analysis model in Object Oriented design and Implementation. IEEE Trans. on Soft. Eng. 32(2), 111–135 (2006)
56. Thonggoom, O.: Semi-automatic conceptual data modelling using entity and relationship instance repositories. PhD Thesis, Drexel University (2011)
57. Unal, O., Afsarmanesh, H.: Using linguistic techniques for schema matching. In: Filipe, J. et al. (ed.) Proc. of ICSOFT 2006. pp. 115–120. INSTICC Press (2006)
58. Winkler, W.E.: String comparator metrics and enhanced decision rules in the fellegi-sunter model of record linkage. (1990), https://eric.ed.gov/?id=ED325505
59. Zhang, J., Shin, B., Choi, J.D., Ho, J.C.: Smat: An attention-based deep learning solution to the automation of schema matching. Symposium on Advances in Databases and Information Systems (2021)
60. Zhang, Y., Di, M., Luo, H., Xu, C., Tsai, R.T.H.: Smutf: Schema matching using generative tags and hybrid features (2024), https://arxiv.org/abs/2402.01685
61. Zhang, Y., Floratou, A., Cahoon, J., Krishnan, S., Müller, A.C., Banda, D., Psallidas, F., Patel, J.M.: Schema matching using pre-trained language models. In: 2023 IEEE 39th International Conference on Data Engineering (ICDE). pp. 1558–1571 (2023)

Goran Banjac is a Senior Teaching Assistant and PhD student at the Faculty of Electrical Engineering, University of Banja Luka (Bosnia and Herzegovina). He is a member of the M-lab Research Group. His research interests include model-driven software development, business process modeling, databases, and UML. He has published several research papers and articles.

Drazen Brdjanin is an Associate Professor at the Faculty of Electrical Engineering, University of Banja Luka (Bosnia and Herzegovina), where he heads the M-lab Research Group. His research interests focus on information systems and software engineering. He has participated in several national and international R&D projects and also authored a number of research papers and articles in the field of model-driven development.

Danijela Banjac is a Senior Teaching Assistant and PhD student at the Faculty of Electrical Engineering, University of Banja Luka (Bosnia and Herzegovina). She is a member of the M-lab Research Group. Her research interests include model-driven software development, business process modeling, object-oriented information systems, and UML. She has published several research papers and articles.

Received: March 01, 2024; Accepted: August 06, 2024.

Computer Science and Information Systems 21(4):1963–1978 https://doi.org/10.2298/CSIS240301063G

Maritime Trajectory Mining: An Automatic Zones of Interests Discovery and Annotation Framework*

Omar Ghannou, Etienne Thuillier, and Omar Boucelma

Aix-Marseille University, CNRS, LIS
Marseille, France
{omar.ghannou, etienne.thuillier, omar.boucelma}@univ-amu.fr

Abstract. As global traffic continues to grow, the identification of areas of particular significance, known as Zones of Interest (ZOI), becomes crucial for optimizing transportation systems and analyzing mobility patterns. In the maritime domain, effective ZOIs discovery is essential for enhancing route planning, improving safety measures, and managing resources efficiently. Within the context of trajectory mining, these ZOIs provide valuable insights into movement behaviors and operational efficiencies. In this paper, we present a framework for discovering and annotating ZOIs within maritime trajectories. The proposed approach involves processing raw positional data to initially identify candidate ZOIs, which are subsequently refined using contextual information. By leveraging real georeferenced vessels trajectories, collected from thousands of commercial ships, this framework proposes a structure of elements that will be implemented as part of the TNTM French project. While this research contributes to maritime field by providing a method for ZOIs discovery and annotation, it can be generalized to various application domains that may leverage of mobility data analytics.

Keywords: Maritime Trajectory Mining, Zone Of Interest, ZOI, Area Of Interest, AOI, Stops Extraction, Classification, Contextual Approach, OpenStreetMap, OSM, VGI

1. Introduction

Trajectory mining or trajectory data mining is an interdisciplinary research area that focuses on analyzing and extracting knowledge from trajectory data, which comprises spatio-temporal information collected from moving objects, such as GPS logs, mobile phone signals, and surveillance cameras. Trajectory mining has been widely applied in various domains, such as transportation, urban planning, social network analysis, or environmental monitoring.

For the specific maritime transportation domain, as global traffic continues to grow, accounting for over 80% of transportation transactions worldwide [6], the characterization of vessel trajectories is an activity of major importance.

In this paper, we focus on specific points of a trajectory, that is, Zones of Interest (ZOI). A ZOI denotes a geographical area that attracts particular attention for various reasons, such as significant events, research endeavors, or specific actions aligned with stakeholder priorities.

* This paper is an extended paper of the published DC paper "Automatic Discovery of Zones of Interests with Maritime Trajectory Mining" in ADBIS2023 conference

In the maritime domain, ZOIs are essential for optimizing routes, assessing risks, and identifying commercial opportunities. For instance, maritime carriers can leverage ZOI knowledge to evaluate risks related to weather conditions or illicit activities and capitalize on commercial prospects like servicing booming ports or supplying raw materials.

The increasing availability of GPS data and the proliferation of Location-Based Services have enabled the extraction of valuable insights from trajectory data. Effective ZOI discovery within maritime trajectories can lead to significant benefits, such as identifying optimal docking locations, which is vital given the high costs associated with boat docking. This can significantly impact overall trajectory expenses [12], allowing maritime operators to strategically allocate resources and minimize unnecessary expenses [5]. Moreover, ZOI discovery contributes to optimizing trajectories, enhancing efficiency, and reducing fuel consumption. Adherence to site-specific environmental and security regulations, such as constraints on pollutant emissions (e.g., CO_2, NO_2, SO_2) and the designation of protected areas, further underscores the importance of ZOI knowledge.

Given the significant role that ZOIs play in maritime trajectories, it is crucial to develop a methodology for their automatic discovery and labeling. This paves the way for the semantic annotation of trajectories, leading to a better understanding of maritime activities and improvements in maritime transport efficiency through a data-driven approach.

In this paper, we describe a framework for the discovery, description, and annotation of ZOIs within maritime trajectories. This framework is developed as part of the TNTM project [10] which aims to optimize the transportation system and address environmental concerns in line with the International Maritime Organization's strategy for reducing Greenhouse Gas emissions from ships [8]. The methodology behind the framework is data-driven, relying on raw trajectory data mining and contextual metadata for ZOI labeling.

The rest of this paper is organized as follows: In Section 2, we outline the research issues and define formally a ZOI. In Section 3, we review related works in the field. Section 4 offers a detailed description of the framework, emphasizing its structure and components. We discuss a simple use case to demonstrate the main benefits of our framework in Section 5. Finally, Section 6 concludes the paper by highlighting ongoing work and future research directions.

2. Problem statement

Formally, a ZOI can be defined and modelled as an extension of the Open Geospatial Consortium (OGC) features data model [11], representing spatial objects that are significant for various stakeholders. In the maritime domain, ZOIs include not only areas within the sea but also coastal objects such as ports, oil and fuel stocking tanks, and other related infrastructures. A ZOI comprises two essential dimensions: the physical or spatial aspect and the logical or contextual representation. The first dimension involves defining and delineating the geographical area of interest, capturing its spatial boundaries. The contextual aspect provides descriptive insight into the activities corresponding to the ZOI. Hence, there is a necessity for a solution that encompasses both the physical and logical representations of ZOIs, ensuring an effective approach to ZOI discovery and annotation.

Figure 1 illustrates the ZOI data model, an extension of the OGC features data model, defining a ZOI and highlighting its physical and contextual dimensions.

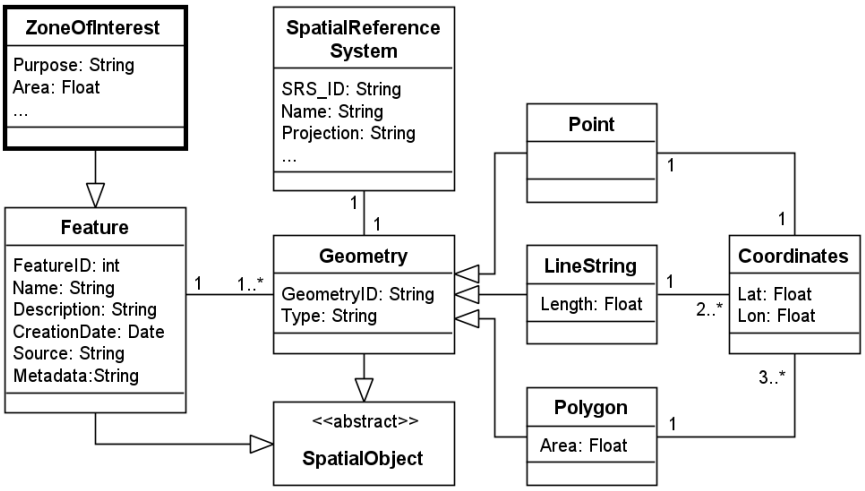


Fig. 1. ZOI data model extended from OGC features model

Framework Overview To meet the spatial and contextual dimensions of a ZOI criteria, we propose a threefold framework to process and transform raw maritime spatio-temporal trajectory data, collected from vessels and/or containers, into ZOI-wise trajectory data where every ZOI is annotated. In the first step we identify crucial stop points from raw trajectory data, which delineate primitive boundaries shapes of ZOIs. These shapes are then used to retrieve surface metadata from Volunteered Geographic Information (VGI) sources, such as OpenStreetMap[1] (OSM). OSM is particularly valuable due to its extensive and continuously updated geographic database. Finally, we use this contextual data to refine the ZOIs shapes and classify the activities they represent. The whole process is detailed in Section 4.

Challenges To develop this framework effectively, we face three primary challenges categories:

1. **ZOI Discovery**:
   Locating ZOIs within spatio-temporal trajectory data and accurately constructing their shapes is complex due to the dynamic nature of maritime activities. Factors such as swell waves, port management tasks, dock activities, changes in vessel paths, and weather conditions contribute to this complexity.
   Challenge examples: Locating ZOIs within trajectory data ; Delineating precise ZOI boundaries due to their dynamic nature
2. **ZOI Annotation**:
   Classifying the types of activities within ZOIs is another challenge. Indeed, classification requires a basis, and to our knowledge, no public ZOI types database exists. Thus, we need to construct a database of annotated ZOIs to unlock benefits like optimized docking costs, improved trajectories, reduced shipping emissions, and enhanced logistical ecosystems.

[1] https://www.openstreetmap.org

Challenge examples: Classification based on semi-structured data ; Utilizing unlabelled data for training ; Identifying possible ZOI types and classes

3. **Contextual Data Quality**:
   Collecting contextual metadata from VGI sources, such as OpenStreetMap, presents quality issues [9], including lack of usable data, imprecise forms, missing tags, heterogeneity of tags, and 'expansionist' relationships. Preparing a tag dataset for ZOI classification based on their tags poses additional challenges, such as selecting representative zones and processing tags manually or semi-automatically.
   Challenge examples: Inconsistencies in data representation and tagging ; Challenges in ensuring data accuracy and completeness ; Heterogeneity in data quality and tag distribution across different regions

3. Related work

Trajectory mining as discussed is widely applied in various domains, such as transportation, urban planning, social network analysis, environmental monitoring, etc. Within this broad context, the discovery and analysis of ZOIs have not gained yet significant attention in trajectory mining, despite their potential applications across various fields. This section delves into existing literature related to ZOI discovery and comparable studies that partially address this topic, discussing the gaps and challenges that our proposed framework aims to address.

To the best of our knowledge, the literature does not provide works specifically dedicated to the discovery and annotation of ZOIs. Most of the existing works focus on stops detection within trajectories, which can imply that a stop represents a significant point in the trajectory. However, stops alone do not necessarily indicate meaningful places or ZOIs, as they lack contextual semantics and do not specify the nature or type of these places. Despite the emphasis of these works on stops extraction within trajectories, we propose a comparative analysis to highlight the gaps targeted by this work. This framework not only identifies ZOIs but also precisely delineates their physical boundaries and annotates them with contextual information, specifically targeting the maritime domain where there is a lack of on-ground data to aid in context, classification, and annotation of ZOIs.

The SMoT model [1] enriches trajectories with semantic geographical information by extracting moves and stops using predefined geometries on the ground. This method efficiently identifies moves and stops, enriching trajectory data with semantic information. However, SMoT relies on predefined geometries, limiting its adaptability. An improved version, CB-SMoT [14], extracts additional stops not considered by SMoT as long as they do not necessary intersect with the predefined regions. While CB-SMoT enhances SMoT by identifying more stops, it remains dependent on predefined geometries.

Trajectory analysis using the Mobility Context Cube [13] integrates some contextual data, precisely points of interest opening hours, with extracted stops to classify real stops from fake ones using an SVM classifier. This approach effectively combines limited contextual data with trajectory analysis but does not address the precise spatial shape of ZOIs or extensive contextual dimension. This work primarily focuses on urban mobility contexts.

The framework proposed by Bisone [3] for extracting the operational behaviors of emergency vehicles using GPS data to optimize fleet management and route planning. It focuses on optimizing operational behaviors and utilizes GPS data to identify vehicles activities based on operational contexts. However, this work relies on physical devices installed inside the emergency vehicles - such as handbrake detection for instance - to extract stops, more precise shapes and contextual identification of ZOIs. This highly impacts the framework adaptability and applicability to other cases or scenarios as it is specifically limited to emergency vehicles' data.

DJ-Cluster [15] extends the DBSCAN algorithm to extract significant locations or ZOIs based on trajectory data by introducing density-joinable clusters. This method is effective for identifying significant locations using density-based clustering but focuses solely on spatial aspects, neglecting temporal, precise ZOI shape and contextual dimensions.

ST-DBSCAN [2] considers both spatial and non-spatial aspects of data to extract stops within trajectories. It integrates non-spatial dimensions with spatial data, providing a more comprehensive approach to stop extraction. Despite this, it lacks focus on precise ZOI shape delineation and lacks contextual data integration for annotation. Similarly, T-DBSCAN [4] combines temporal and spatial dimensions to mine stops from trajectories. While effective in combining temporal and spatial data for stop extraction, it does not address precise ZOI shape delineation or contextual information integration.

This Paper presents an extended and refined version of our previous work on trajectory mining and ZOIs discovery [7], introducing a matured and enhanced framework that addresses several gaps in the related works. Table 1 offers a summarized view of the related works drawbacks, and presents how our framework benefits from each of the related works. Our framework addresses the gaps identified in the existing literature by providing comprehensive ZOI discovery, precise spatial shape determination, contextual aspect identification, and cross-domain adaptability, without relying on information from predefined ZOIs. This integrated approach ensures an effective solution for ZOI detection and classification within spatio-temporal trajectory data. While the description of our framework is tailored for maritime data in this paper, the methodology is broadly applicable to various transportation domains, including vehicular and pedestrian trajectories.

4. Framework Proposal

In this section, we present an extended version of our previous framework [7], designed to identify ZOIs within trajectory data. While this work mainly focuses on the maritime domain, the framework is adaptable across different domains. Our approach includes several steps to detect, outline, and classify ZOIs efficiently. By combining raw trajectory data with contextual on-ground information and industry expertise, we aim to improve the accuracy and precision of ZOI detection and characterization. Our framework unfolds in three phases:

1. **Extraction of starts and stops from trajectory data.** We use a custom density-based algorithm, that we name STC-DBSCAN (see 4.1), to extract stop points within raw vessels and/or containers trajectories. Then, we conduct a second clustering layer to identify the most important stops areas, which form the initial shape of the ZOIs.

Table 1. Benefits and limitations of related works

Work	Stop Extraction	Precise ZOI Spatial Shape	Contextual Aspect	Cross-Domain	Non-Predefined ZOI Info
SMoT [1]	✓	✓		✓	
CB-SMoT [14]	✓			✓	
Mobility Context Cube [13]	✓			✓	
Emergency Vehicles Activity Identification [3]	✓	✓	✓		✓
DJ-Cluster [15]	✓			✓	✓
ST-DBSCAN [2]	✓			✓	✓
T-DBSCAN [4]	✓			✓	✓
This Framework	✓	✓	✓	✓	✓

2. **Context retrieving and ZOI shape refinement.** We leverage the constructed shapes to retrieve on-ground contextual data from open geographical base, particularly, OpenStreetMap (OSM). More specifically, we extract textual metadata (OSM tags) associated with all the objects located within or intersect with the initial ZOIs shapes. We then refine and construct a more realistic shape using the OSM objects' polygons.
3. **ZOI description and annotation.** We use the textual content of OSM tags to create a ZOI classification model. We then run all the extracted ZOI into this model to describe and annotate them.

The overall framework is depicted in Figure 2. For clarity, a simplified view of the framework is provided. Each step, indicated by a colored rectangle, is detailed in its corresponding subsection. The threefold splitting ensures a comprehensive approach to ZOI detection, refinement, and annotation, leveraging trajectory data, on-ground contextual data, and domain-specific knowledge.

4.1. Step 1: start/stop detection

The initial step of the framework involves detecting start and stop points from raw trajectory data, followed by a clustering process to extract the initial shapes of ZOIs. This step employs two algorithms, each designed to achieve one of the aforementioned objectives. The process of Step 1 is depicted in Figure 3.

Start & Stop detection To extract stops positions from raw trajectory data, we leverage the STC-DBSCAN algorithm, which we previously introduced in [7]. As described in Algorithm 1, STC-DBSCAN extends the DBSCAN (Density-Based Spatial Clustering of Applications with Noise) method to identify potential stop points based on density. It subsequently refines these points by evaluating their density levels and temporal characteristics to distinguish between genuine stops and slowdown areas.

DBSCAN is a well-known clustering algorithm used to identify clusters of varying shapes and sizes in spatial data. It regroups points that are closely packed together while

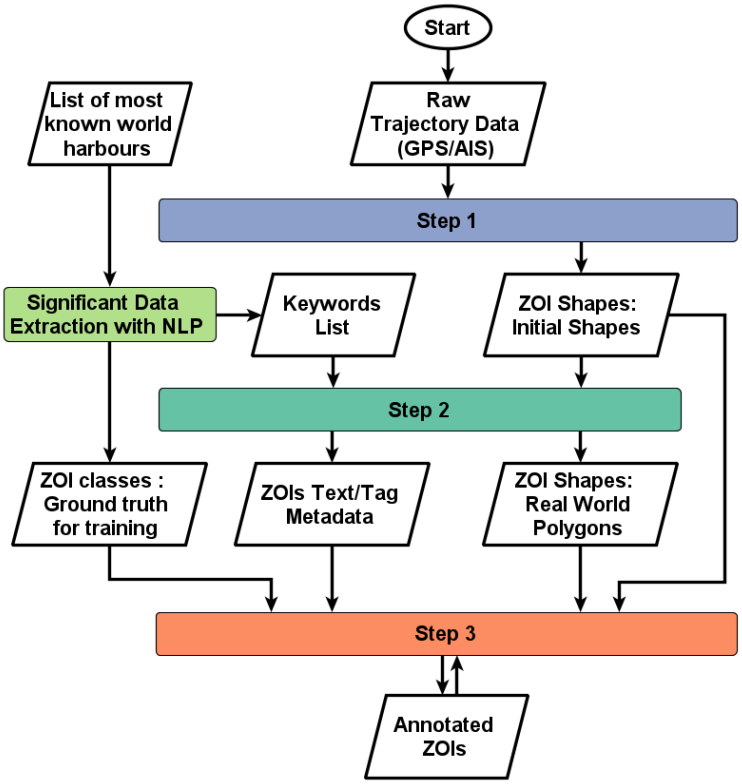


Fig. 2. ZOI Discovery and Annotation Framework

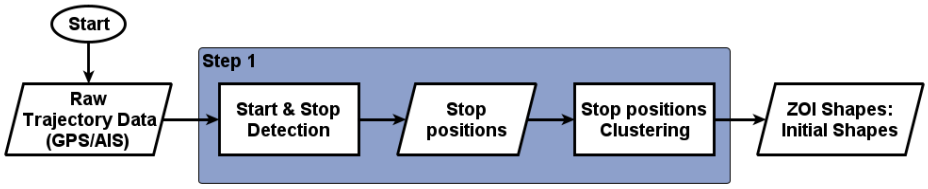


Fig. 3. Framework Step 1 : Start & stop detection

marking points that lie alone in low-density regions as outliers. By examining the density of data points in a given neighborhood, DBSCAN can effectively detect clusters and distinguish noise, making it suitable for applications where the data contains clusters of arbitrary shapes and noise.

Although the STC-DBSCAN algorithm has not yet undergone formal testing, we have initiated the development of a benchmark. Our objective is to compare our algorithm's

Algorithm 1: Brief representation of the **S**patio**T**emporal **C**lustering - DBSCAN Algorithm (STC-DBSCAN)

Data: T : trajectory points, $t1$: spatial threshold, $t2$: temporal threshold
Result: ST : Trips, S : Stops
if *consecutive points are within the threshold distance* **then**
  Delineate stop zone beginning;
else
  Delineate the end of stop zone if already initiated, then start a trip
end
for (*each stop zone z* **)** {
  if *z represents high density (BboxDiagDistance* $<$ *t1) && (cumulatedTime in z* $>$ *t2)* **then**
    Validate the stop zone
  else
    Represents a slowdown area in the current trip
  end
}

performance with that of several well-known algorithms in the context of start-stop extraction from raw trajectory data.

Stop positions clustering After extracting stops from all trajectories, we proceed to a clustering phase to identify areas of high stop density. Using the extracted stops as input, we apply the DBSCAN algorithm to cluster regions with concentrated stop occurrences, as the DBSCAN is suitable to construct clusters of arbitrary shapes. This step is crucial because individual stops within a trajectory do not necessarily indicate the existence of a ZOI. The reasons for individual stops can vary widely and may not always signify areas of importance. However, by identifying clusters with high stop density, we can more reliably locate areas likely to represent ZOIs. This allows us to establish the initial shape of the ZOI.

4.2. Step 2: Context retrieving and ZOI shape refining

Through the first step of the framework, we have identified areas of significance within maritime trajectories. These areas may be offshore (vessels paths or stops for instance), or on the coast (docking, container loading or unloading for instance). These zones hold significance, yet lack clear identification or identity. While their importance is evident, they require human intervention to ascertain their precise nature each time they are encountered. In order to enhance the discovery process and automate it, as well as to annotate our ZOIs, we retrieve meaningful contextual data that can help in ZOI identification.

Furthermore, the stop position clustering algorithm from step 1 provides coarse ZOI shapes. These shapes are determined by the raw trajectories dataset and hold two main drawbacks : 1) the shapes are dependent of several maritime factors such as changes in vessels paths or weather conditions, and 2) the shapes are not geographically refined as they do not detect nor encompass human buildings such as ports or docks.

The second step of our framework aims to achieve such goals through VGI context retrieving. The detail of this step is presented in Figure 4.

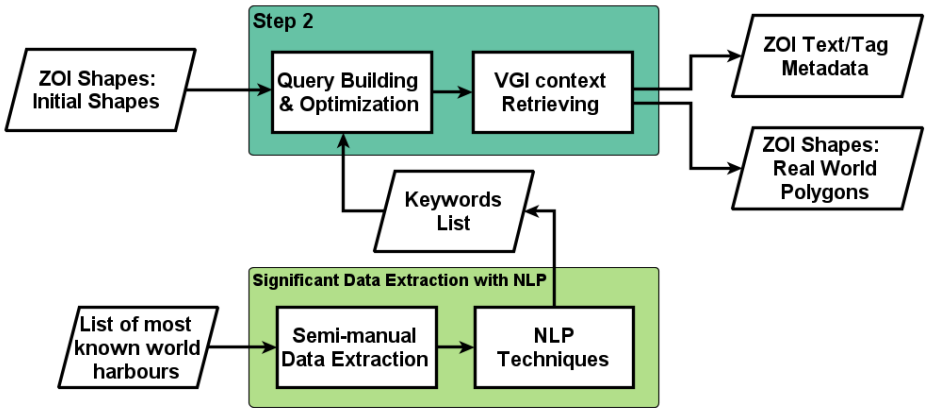


Fig. 4. Framework Step 2 : ZOI shape refining

The second step is divided into two main processes. The first process, named *"step 2"* in Figure 4 consists in implementing on-ground information retrieval over the initial ZOI shapes extracted in step 1. This process runs queries on VGI resources to extract contextual tags metadata, and precise polygons shapes. The second process, termed *"Significant data extraction with NLP"* involves constructing a list of keywords to facilitate VGI querying and, subsequently, in sub-section 4.3, to establish a set of ZOI types or classes. Each process is described in detail below.

VGI Context Retrieving To extract contextual metadata from the initial ZOIs, we run customized queries over the OSM database using the Overpass API tool[2]. The API retrieves relevant on-ground textual and geographical information (called tags), providing valuable insights about the ZOIs. Additionally, we use this metadata to construct refined ZOIs shapes using the actual OSM objects presents within these areas.

This process faces significant challenges due to data quality issues within the OSM database as discussed in section 2. The structure of OSM data, consisting of *nodes, ways*, and *relations*, does not consistently provide accurate logical representations of objects and their relationships. For instance, while attempting to retrieve objects present within a ZOI, the *relations* can expand extensively and retrieve irrelevant data and objects, posing a significant risk if integrated into a larger system or environment for further processing. We present an illustrative example in Figure 5. We consider a scenario where a small ZOI inadvertently retrieves data encompassing the entirety of the Arabian Sea. This occurrence illustrates the challenges of using such semi-structured data, especially the *relation* tags. Such anomalies highlight the need for careful consideration and mitigation strategies to address data quality issues effectively.

Significant Data Extraction with NLP To take up this challenge, we propose an adapted workflow that enhances the overall framework and specifically targets this phase and its issues. The main idea of this part is to build a list of keywords that specifically target maritime objects or infrastructures, and to use this list as a filtration basis aimed at restricting the scope of OSM queries research and their *relations* expansion.

[2] https://overpass-turbo.eu

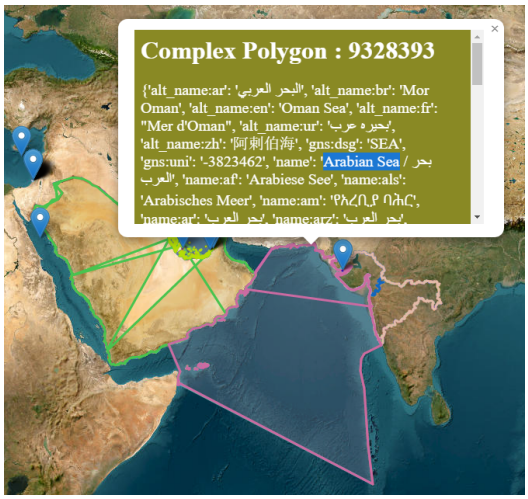


Fig. 5. Example of data quality issues

By incorporating specific keywords, we establish criteria that ensure only relevant objects are retrieved using the OSM Overpass engine. This approach mitigates the risk of extensive expansion caused by *relations* in the OSM data. By enforcing a requirement for objects to contain at least one of the specified keywords, we streamline the query process to prioritize the retrieval of useful data and mitigate quality issues. For instance, missing tags in some objects, variations in tag presence across objects, and data heterogeneity arising from multiple languages and regional verification practices.

To address these challenges, we propose a solution focused on the semi-automatic generation of a keywords database. In an initial version of our framework, we suggest selecting a representative sample of the largest and most active ports from the OSM dataset. By conducting data profiling on these ports, we aim to extract statistics on tag usage to identify consistent keywords indicative of port-related features. Additionally, while industrial keywords are essential, their indiscriminate usage may lead to inaccurate results due to the widespread presence of industrial zones worldwide. Hence, careful selection and refinement of keywords, along with tailored queries designed for OSM, are crucial for effectively leveraging the dataset for ZOI discovery and annotation.

4.3. Step 3: ZOI annotation

The final part of our framework involves three main processes: the reconstruction, classification and annotation of the ZOIs. The detail of this step is presented in Figure 6.

ZOI shape reconstruction Through the filtration process from step 2, we identify relevant on-ground objects inside a ZOI and extract their shapes. Then we merge the initial ZOI shape with the shapes of these objects and create a refined and final shape that accurately represents the ZOI.

With this final process we construct final ZOIs shapes that hold maritime and human meanings. Infrastructures such as port, docks, terminals, canals, are added to the initials ZOIs shapes, allowing a more precise contextual annotation of each ZOI.

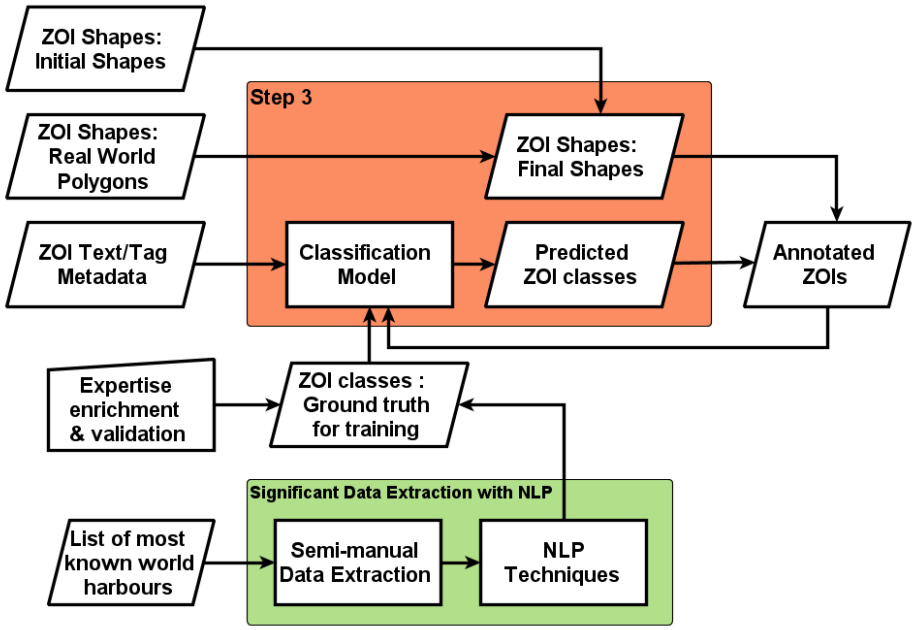


Fig. 6. Framework Step 3 : ZOI annotation

ZOI Classification The second process of this step consists in classifying each ZOI. This classification process is done using a list of ZOI classes which is built from two main sources. First, using the tags database from step 2, we conduct a clustering phase to identify potential ZOI classes. These classes are then validated and refined through expertise and business knowledge. In the TNTM french project, these classes will be validated by the main maritime transportation authority leading the project.

Once a list of maritime ZOI classes, or types, are defined, we plan to use large language models (LLMs) like BERT, embedding techniques such as Word2Vec or TF-IDF, and various text classification methods to classify each ZOI. These methods will leverage the tags and data extracted from step 2 to predict the class of each ZOI, ensuring a comprehensive and accurate annotation process.

ZOI Annotation At the conclusion of our framework, we merge the final shape of our ZOI with its predicted class. This integration results in an annotated ZOI, offering both spatial and contextual representation. Additionally, this annotation serves as valuable feedback for enhancing the classification model iteratively and progressively.

5. Experiments

The experiments for this study are conducted using a set of maritime trajectory data from the TNTM project. These data hold vessels and/or containers positions. Currently, our framework effectively detects and delineates ZOIs, and extract OSM tags and shapes. This section presents a simple use case with experimental results, focusing on demonstrating the capabilities and outcomes of our framework through some ZOIs identification examples.

5.1. Initial ZOI Shape Construction

The first part of our framework involves detecting and extracting stops from trajectory data and constructing initial ZOI shapes. Figure 7 illustrates this process with the STC-DBSCAN algorithm. With a focus on the *Jawaharlal Nehru* port, India, we identify significant stop points and delineate initial ZOI shapes from several raw trajectories. Sub-figure 7a, represents different stops identified by the STC-DBSCAN algorithm. Rounded points represent different vessels and/or containers positions, while star-shaped points represent the detected stops. Sub-figure 7b represents the construction of the initial ZOI shape using the DBSCAN algorithm over the detected stops. As we discussed in previous sections, this shape is raw and do not encompass human activities and infrastructure. The next steps consists in refining this ZOI shape to include the totality of the *Jawaharlal Nehru* port infrastructures.



(a) Stops detected with STC-DBSCAN algorithm



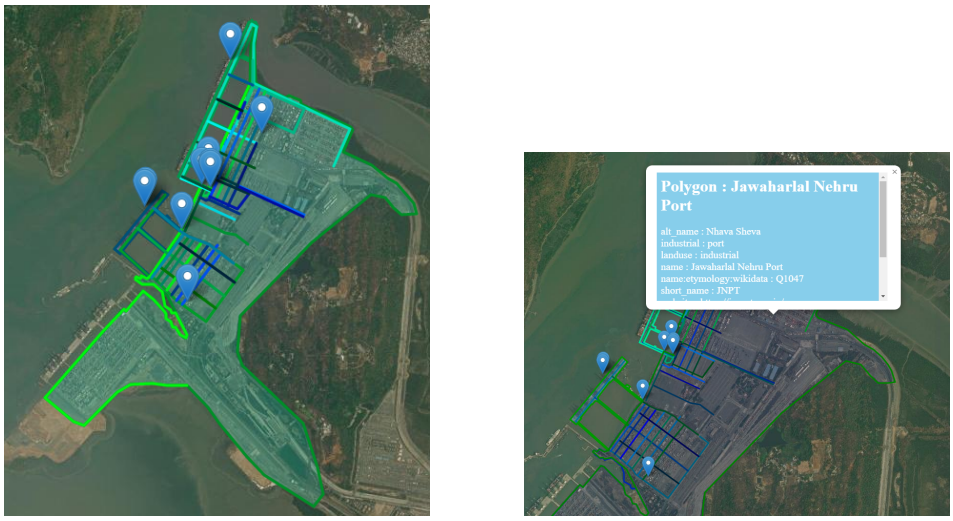
(b) Initial ZOI shapes extracted with DBSCAN algorithm

Fig. 7. Stops detection and initial ZOI construction: example of the *Jawaharlal Nehru* port

5.2. ZOI Shape Refinement and Context Retrieval

The second part of our framework refines the initial ZOI shapes using contextual data retrieved from OSM. We extract textual metadata associated with the objects located or intersect with the initial ZOI shapes and reconstruct more precise and realistic ZOI shapes. Figure 8 illustrates two perspectives of the refined shape of the *Jawaharlal Nehru* port ZOI, along with its retrieved description. On the left, sub-figure 8a displays the shapes on a satellite map. We see the retrieved objects shapes related to the initial ZOI, such as buildings, roads, as well as the docks where the ship are anchored. The pin-shaped objects represent various ZOI-related OSM node objects, as also correspond to the defined stops clusters within the ZOI. These objects contribute to constituting the final ZOI shape. In sub-figure 8b, we see the descriptive texts of the ZOI. Here, for example, we display the

textual information from the polygon presenting main shape of the *Jawaharlal Nehru* port, providing further insights about the ZOI characteristics.



(a) Transportation Map View **(b)** OSM Tags descriptions

Fig. 8. ZOI shape refining using VGI Context Retrieving: example of the *Jawaharlal Nehru* port

5.3. ZOI Annotation

The final part of our framework involves the classification and annotation of the ZOIs. By utilizing the textual content of OSM tags, we feed a trained ZOI classification model to classify and annotate the ZOIs. Sub-figure 8b depicts the retrieved contextual information that aids in the classification and annotation of these ZOIs. The classification model is currently under development, and we discuss this topic in the following subsection.

5.4. Discussion

The objective of this paper is to propose and demonstrate a framework for the discovery and annotation of ZOIs. Although the benchmarking of the stop extraction algorithm and the text classification model for ZOI classification are currently under development, these are not the primary objectives of this paper. The main challenge in conducting an extensive practical comparison study of the proposed framework is the lack of existing frameworks in the literature that cover the dimensions of ZOI discovery and annotation as comprehensively as ours. While individual components, such as the stop extraction algorithm and the text classification model, can be benchmarked and compared to existing literature, this paper focuses on demonstrating the overall framework for ZOI discovery and annotation.

The elements presented in this section highlight examples of the outcomes of our framework in identifying and refining ZOIs within maritime trajectories. The precise shapes and contextual descriptions of ZOIs are crucial for realizing the benefits mentioned previously. Future work will involve benchmarking the stop extraction algorithm and the text classification model to further validate and enhance the proposed framework.

6. Conclusion

Analysis/mining ship trajectories plays a crucial role for various purposes such as preventing vessel collision or optimizing vessel energy consumption, to cite a few.

This paper describes a machine-learning based approach and framework for the discovery, refinement and annotation of Zones of Interest (*aka* ZOI) in a maritime trajectory. Our approach integrates trajectory analysis with contextual data and business insights, providing accurate and precise ZOI characterization. The framework consists mainly of a three-step process: (1) extraction of stops to devise the initial ZOI shape, (2) shape refinement by means of keyword-based contextual data retrieval, and (3) ZOIs' classification and annotation.

We believe this framework may bring a useful piece in the maritime trajectory mining landscape, but also, in other mobility application domains. As a future work, we envision benchmarking of the framework components and its deployment with real world data vessel trajectories, supplied by the TNTM consortium.

References

1. Alvares, L.O., Bogorny, V., Kuijpers, B., de Macedo, J.A.F., Moelans, B., Vaisman, A.: A model for enriching trajectories with semantic geographical information. In: Proceedings of the 15th Annual ACM International Symposium on Advances in Geographic Information Systems. GIS '07, Association for Computing Machinery, New York, NY, USA (2007), https://doi.org/10.1145/1341012.1341041
2. Birant, D., Kut, A.: St-dbscan: An algorithm for clustering spatial–temporal data. Data Knowledge Engineering 60(1), 208–221 (2007), https://www.sciencedirect.com/science/article/pii/S0169023X06000218, intelligent Data Mining
3. Bisone, Frédérick, Étienne, Laurent, Devogele, Thomas: Modélisation et extraction de la sémantique des trajectoires à partir de données multicapteurs. Rev. Int. Geomat. 28(4), 461–483 (2018), https://doi.org/10.3166/rig.2018.00065
4. Chen, W., Ji, M., Wang, J.: T-dbscan: A spatiotemporal density clustering for gps trajectory segmentation. International Journal of Online and Biomedical Engineering (iJOE) 10(6), pp. 19–24 (Oct 2014), https://online-journals.org/index.php/i-joe/article/view/3881
5. Clark, X., Dollar, D., Micco, A.: Port efficiency, maritime transport costs, and bilateral trade. Journal of Development Economics 75(2), 417–450 (2004), https://www.sciencedirect.com/science/article/pii/S0304387804000689, 15th Inter American Seminar on Economics
6. Fratila (Adam), A., Gavril (Moldovan), I.A., Nita, S.C., Hrebenciuc, A.: The importance of maritime transport for economic growth in the european union: A panel data analysis. Sustainability 13(14) (2021), https://www.mdpi.com/2071-1050/13/14/7961

7. Ghannou, O.: Automatic discovery of zones of interests with maritime trajectory mining. In: Abelló, A., Vassiliadis, P., Romero, O., Wrembel, R., Bugiotti, F., Gamper, J., Vargas Solar, G., Zumpano, E. (eds.) New Trends in Database and Information Systems. pp. 684–692. Springer Nature Switzerland, Cham (2023)
8. International Maritime Organization: 2023 IMO Strategy on Reduction of GHG Emissions from Ships. https://www.imo.org/en/OurWork/Environment/Pages/2023-IMO-Strategy-on-Reduction-of-GHG-Emissions-from-Ships.aspx (2023), accessed on July 19, 2024
9. Kaur, J., Singh, J., Sehra, S.S., Rai, H.S.: Systematic literature review of data quality within openstreetmap. In: 2017 International Conference on Next Generation Computing and Information Systems (ICNGCIS). pp. 177–182 (2017)
10. Pôle Mer Méditerranée: Transformation Numérique du Transport Maritime. https://polemermediterranee.com/domaines-dactions-strategiques/transformation-numerique-du-transport-maritime/ (2022), accessed on July 20, 2024
11. W3C Group, OGC Document Number: OGC 15-107: Open Geospatial Consortium (OGC) features data model. https://www.w3.org/TR/sdw-bp/#spatial-things-features-and-geometry (2023), accessed on July 22, 2024
12. Wilmsmeier, G., Hoffmann, J., Sanchez, R.J.: The impact of port characteristics on international maritime transport costs. Research in Transportation Economics 16, 117–140 (2006), https://www.sciencedirect.com/science/article/pii/S0739885906160060, port Economics
13. Wu, T., Shen, H., Qin, J., Xiang, L.: Extracting stops from spatio-temporal trajectories within dynamic contextual features. Sustainability 13(2) (2021), https://www.mdpi.com/2071-1050/13/2/690
14. Xiu-Li, Z., Wei-Xiang, X.: A clustering-based approach for discovering interesting places in a single trajectory. In: 2009 Second International Conference on Intelligent Computation Technology and Automation. vol. 3, pp. 429–432 (2009)
15. Zhou, C., Frankowski, D., Ludford, P., Shekhar, S., Terveen, L.: Discovering personally meaningful places: An interactive clustering approach. ACM Trans. Inf. Syst. 25(3), 12–es (jul 2007), https://doi.org/10.1145/1247715.1247718

Omar Boucelma is a full Professor in Computer Science at Aix-Marseille University. He has a background in statistics (data analysis) and data management and has been conducting R&D projects in core databases and geospatial systems and applications. As part of his international research and educational activities, he was a visiting research fellow at CU Boulder, NUS Singapore and UNICAMP (Brazil). He has been involved in several French and (DARPA) American research projects, involving academia and/or industry. His recent works relate to geospatial entity matching, data streams and mobility data analytics.

Etienne Thuillier received a DUT (Technology degree) in computer sciences from the IUT of Lannion, France, in 2010 and the title of engineer in computer sciences (M.S. degree) at Belfort-Montbéliard University of Technology (UTBM), France, in 2013. He then obtained the Ph.D. degree in computer sciences at the FEMTO-ST/DISC OMNI team, at the UTBM. He is currently assistant professor at the Aix-Marseille University (AMU) at the LIS lab. In 2014 he was team a leader as part of the ComplexCity project between the UTBM and the Institute of Smart City, at Shanghai University Sino-European University of Technology (UTSEUS), Shanghai, China. His research interest includes

Intelligent Transportation Services (ITS) technologies, and more generally human and urban mobility.

Omar Ghannou is a PhD student at Aix-Marseille University with a background in machine learning. He has experience in domain adaptation, federated learning, and other diverse areas of the field, along with previous experience in the industry. His current research focuses on geospatio-temporal data mining.

Received: February 29, 2024; Accepted: August 06, 2024.