Computer Science and Information Systems 22(4):1509–1532 https://doi.org/10.2298/CSIS241225059Z

Fire Detection Models Based on Attention Mechanisms and Multiscale Features

Shunxiang Zhang[1,⋆], Meng Chen[1], Kuan-Ching Li[2], Hua Wen[1], and Liang Sun[1]

[1] School of Computer Science and Engineering, Anhui University of Science & Technology,
232001 Huainan, China
sxzhang@aust.edu.cn
2260662967@qq.com
1762636707@qq.com
2582132681@qq.com

[2] Department of Computer Science and Information Engineering (CSIE), 13 Providence
University, 43301 Taizhong, Taiwan
kuancli@pu.edu.tw

Abstract. Fire detection is critical in applications such as fire management and building safety, but dispersion and blurring of flame and smoke boundaries can present challenges. Multiple upsampling and downsampling operations can blur the localisation signals, thus reducing accuracy and efficiency. To address this problem, we propose the AMMF(Attention Mechanisms and Multiscale Features) detection model, which integrates an attention mechanism and multi-scale feature fusion to improve accuracy and real-time performance. The model incorporates a dynamic sparse attention mechanism in the backbone network to enhance feature capture and restructures the neck network using CepBlock and MPFusion modules for better feature fusion. MDPIoU loss and Slideloss are then utilised to reduce the bounding box regression error and address the sample imbalance problem respectively. In addition, parameters are shared by merging 3×3 convolutional branches, which optimises the detection head and improves computational efficiency. The experimental results show that AMMF-Detection can significantly improve the detection speed and accuracy on the public dataset.

Keywords: Fire detection,YOLO,Feature fusion,ynamic sparse attention,Multi-scale features

1. Introduction

Detecting objects is essential for analyzing and understanding flame and smoke images, with its main objective being to precisely identify and pinpoint components like the smoke and fire source.The complexity and diversity of fire scenes, the irregular shapes of target objects, and the presence of numerous interfering elements in the images lead to low detection accuracy. Additionally, when processing the original image, up-sampling introduces additional pixels, increasing the sparsity of the original features, while downsampling causes a loss of localization information. This results in the gradual blurring or disappearance of small flame and smoke details, which negatively impacts detection accuracy. Moreover, the current computational efficiency remains a significant challenge, as

[⋆] Corresponding author

fire detection must meet real-time requirements. Consequently, fire detection models must possess strong abstraction and generalization capabilities to handle the complex and dynamic nature of fire scenarios. These demands further complicate target detection, making fire detection a highly challenging task.

Fire detection algorithms are generally divided into three main types: methods based on classifiers, model compression techniques, and deep learning approaches [1,2].Traditional classifier-based techniques rely on manually designed feature extractors to derive image features [3], and then use algorithms like SVM, ID3, or BP neural networks to identify fire and smoke. While model compression methods enhance detection speed, they often face challenges in achieving a balance between accuracy and efficiency. Conversely, fire detection models based on deep learning possess the ability to autonomously extract image features, enabling them to recognize intricate patterns and finer details with greater precision, which leads to enhanced detection accuracy. Mainstream deep learning-based object detection algorithms include SSD [4], YOLO [5,6,7,8], and Transformer-based RT-DETR [9]. Despite advancements in object detection, these methods still face several challenges. First, target features in the image are often scattered with fuzzy boundaries, causing the original features to become more dispersed and sparse during feature fusion, which increases the model's complexity in processing up-sampled features and degrades its performance on certain features. Second, reducing image resolution through down-sampling operations causes the loss of fine details in flame and smoke images. This loss adversely affects the model's capacity to detect small objects or identify localized features. Moreover, increasing model complexity presents a challenge for real-time inference, particularly in resource-constrained environments. Finally, issues such as sample imbalance, poor data quality, and insufficient training data further affect model performance, potentially leading to misdetection or detection failures.

Taking the above considerations into account, this paper introduces the AMMF-Detection model, designed to achieve a balance between detection accuracy, processing speed, and computational efficiency. As illustrated in Figure 1, the model's overall structure comprises three key components: the backbone network, the neck network, and the detection head.The MPfusion module, designed for feature fusion, combines feature maps from three different scales. The CepBlock, a feature extraction module, employs distinct structures during training and inference, ensuring high accuracy during training and fast inference speed. The detection part introduces a novel detection head that integrates seamlessly with the original convolutional block without compromising model performance. This paper aims to enhance the accuracy of fire detection and the speed of inference, all while minimizing the complexity of the model. This goal is accomplished by comprehensively extracting edge characteristics, including color, shape, and texture, from images of flames and smoke to improve the accuracy of fire detection. The model aims to better serve applications in fire safety and emergency response, including fire management, warehousing and logistics, building safety, and other monitoring and detection scenarios.

Our contributions are summarized in the following three aspects.

(1)We propose a method to optimize the backbone network using dynamic sparse attention. This approach aims to enhance the model's feature memory and recognition capabilities, enabling it to focus more effectively on feature selection. By addressing the challenges posed by complex backgrounds, this method effectively mitigates issues of target misdetection and omission.

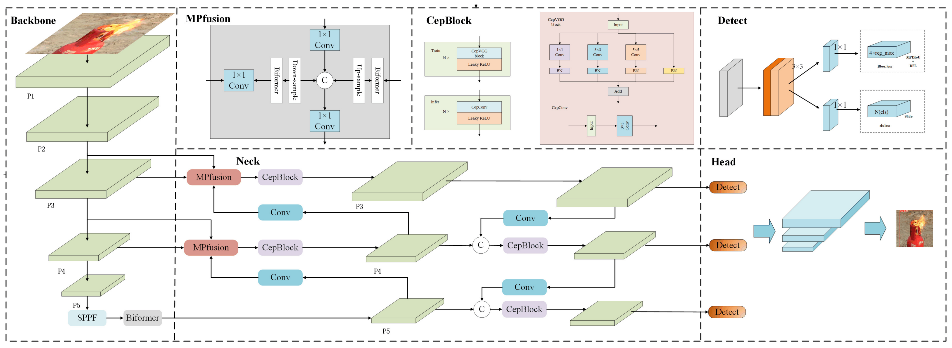


Fig. 1. Overall framework of proposed AMMF

(2)We propose a method that integrates the parameter reconfiguration of the CepBlock module with the MPFusion module to redesign the neck network. This redesign enables the fusion of feature maps across multiple receptive fields, facilitating deeper exploration of features at different levels and improving the overall feature representation.

(3)We propose a novel lightweight detection head design that simplifies the model structure by merging the 3×3 convolutions in the original branches, achieving parameter sharing. This design enhances computational efficiency and makes the model more suitable for deployment on resource-constrained devices.

The structure of the paper is outlined as follows: Section 2 reviews related research on fire detection. Section 3 details the proposed improved methodology. Section 4 explains the experimental setup, training approach, and results. and Section 5 summarizes the key findings and conclusions.

2. Related work

Current fire detection techniques can be categorized into three main approaches: traditional methods relying on image features, target detection algorithms utilizing model.

2.1. Conventional fire detection methods based on image features

Traditional fire detection methods predominantly relied on handcrafted feature extractors, emphasizing characteristics like color, luminance, texture, and edges in images. For instance, Chen et al. [10] proposed an approach based on RGB chromaticity to detect flame and smoke pixels without the need for physical measurements. Similarly, Binti Zaidi et al. [11] leveraged RGB and YCbCr color components, analyzing their specific values to identify fire. Vipin et al. [12] introduced a rule-based model that classified flame pixels by separating luminance and chrominance within the RGB and YCbCr color spaces.

To further enhance detection accuracy, researchers have also investigated texture feature extraction. Dimitropoulos et al. [13] employed background subtraction and color analysis to identify potential ignition regions, followed by modeling fire behavior using spatiotemporal features and dynamic texture analysis. Ye et al. [14] developed a dynamic texture descriptor based on surface waveform transformations combined with a Hidden Markov Tree model, which was employed for smoke detection in video sequences.

Other approaches have combined color and motion features. For example, Chunyu et al. [15] applied an optical flow algorithm to calculate flame motion features, which were then integrated with color features for video-based fire detection. Li et al. [16] introduced a framework that integrates flame color, dynamic motion patterns, and flicker properties. Although these approaches have enhanced the reliability and precision of fire detection, the complexity inherent in fire scenarios frequently constrains their effectiveness. Handcrafted feature extractors struggle to comprehensively represent object features in such scenarios, leading to a decline in feature extraction precision.

2.2. Fire detection models based on model compression

The primary approaches for model compression and acceleration include network pruning and sparsification [17], lightweight model design [18,19,20], knowledge distillation [21], and compact network architecture development. Techniques such as pruning and sparsification simplify the model by detecting and eliminating redundant parameters or connections, which effectively reduces the computational load and parameter count in neural networks. These methods are often applied to pre-trained models. Alternatively, a compact network architecture can be selected during the initial model design phase.

For instance, C. Szegedy et al. [22] introduced a model architecture grounded in the Hebbian principle and multiscale processing for target detection tasks. Similarly, N. Ma et al. [23] developed ShuffleNetv2, which leverages the ChannelShuffle operation and pointwise grouped convolution to enable efficient feature extraction and information exchange. This design achieves remarkable performance and computational efficiency across multiple computer vision tasks. Furthermore, A. Howard et al. [24] proposed MobileNetv3, which optimizes feature extraction and model compression by employing techniques such as candidate network structure search and network tilting.

These lightweight architectures primarily address the challenges of reducing computational requirements and parameter counts. However, they often come with a trade-off in accuracy, particularly when compared to models like YOLO. While these models excel in specific scenarios, they may struggle to achieve YOLO's level of precision in more complex environments.

2.3. Deep learning based fire detection methods

In recent years, fire detection techniques based on deep learning have primarily employed either single-stage or two-stage strategies. Among these, single-stage methods—such as SSD [4], SPPNet [25], YOLOv3 [26], and YOLOv4 [27] are widely favored due to their ability to quickly and directly predict target categories and locations from input images.In contrast, two-stage methods, including R-CNN [28], Fast R-CNN [29], Faster R-CNN [30], and Mask R-CNN [31], offer higher accuracy but are generally slower, making them widely adopted in target detection tasks. Despite significant progress, these methods face challenges related to high storage and computational resource requirements. To address these limitations, the YOLO series has demonstrated superior performance through continuous iterations and enhancements, particularly in fire detection tasks.

J. Miao et al. [32] introduced an enhanced real-time fire detection algorithm built upon YOLOv5s. This approach incorporates sensory field enhancement along with channel attention mechanisms, aiming to improve both the efficiency and precision of recognizing flames and smoke. Similarly, M. Luo et al. [33] improved YOLOv5s for fire detection by replacing the SPP module with the WASP module and introducing attention mechanisms along with a small-target detection layer, effectively enhancing the detection of small-scale forest fires. Additionally, Li, Pu, and Li, Songbin et al. [34,35] developed fire detection algorithms leveraging target detection CNNs and implicit depth supervision mechanisms, respectively, which addressed the trade-offs between accuracy, model size, and processing speed.Majid et al. [36] Combining EfficientNetB0 with an attention mechanism to propose a fire detection model, real-world fire image dataset achieved good results.Pincott et al. [37] developed a computer vision-based indoor fire and smoke detection system using the Faster R-CNN Inception V2 and SSD MobileNet V2 models, which was initially evaluated with a small training dataset and achieved some results. These approaches addressed prevalent challenges in fire detection algorithms, such as insufficient accuracy and significant latency.

These advancements highlight the potential of deep learning in enhancing fire detection accuracy and reducing false-negative rates. However, existing models still exhibit deficiencies, such as inadequate key feature extraction, limited feature map representation capabilities, suboptimal target loss calculations, and high model complexity. To address these shortcomings, this study adopts YOLOv8n as the benchmark model and aims to improve its capability in detecting flame and smoke boundaries. Key improvements include refining the loss function, incorporating attention mechanisms, optimizing feature fusion, and enhancing feature selection. Simultaneously, efforts are directed toward reducing computational complexity and storage requirements to enable practical deployment and application.

3. Improved methodologies

3.1. Backbone network improve

The backbone network of YOLOv8 uses convolutional and inverse convolutional layers to extract features, using residual connectivity and bottleneck structure to optimise network size and performance. The C2f (Convolution to Fuly Connecied) module is used as the basic building block, but feature redundancy exists after SPPF (Spatial Pyramid Pooling - Fast).

To boost detection accuracy on the fire dataset and refine feature extraction, the dynamic sparse attention mechanism from dynamic sparse attention [38] is applied. Integrated into layer 11 of the backbone network, this mechanism efficiently calculates attention by isolating irrelevant key-value pairs and focusing on the most relevant ones. Leveraging the query-adapted input feature map, the model focuses more on essential key information, reduces the impact of background noise, lowers computational and storage demands, enhances its understanding of the input, and ultimately improves detection accuracy. dynamic sparse attention employs an attention mechanism to capture global feature relationships, offering a superior global perception capability compared to traditional local CNN models. This mechanism functions by encoding the input data sequence, computing and normalizing the dot product between queries and keys, and then applying weighted summation. The attention formula is presented in Equation (1):where $\sqrt{d_k}$ is the scaling factor to prevent concentration of weights and gradient vanishing.

$$Attention(Q,K,V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \quad (1)$$

The attention mechanism proves useful for conducting a global analysis of images, enabling the extraction of features related to small-scale flames and smoke.However, this approach also increases computational complexity and consequently raises computational costs. The CBAM introduced by Woo et al., employs a dual attention mechanism focusing on spatial and channel dimensions. While it demonstrates strong performance, it suffers from significant computational overhead, making it less efficient and not lightweight. In contrast, the ECA (Efficient Channel Attention) module introduced by Wang et al. minimizes model complexity. However, it demonstrates lower effectiveness in fire detection tasks because of its restricted ability to facilitate channel interactions.To address the challenges of high computational complexity and memory usage associated with conventional attention modules, fire detection platforms constrained by resource limitations cannot afford to integrate these modules. To mitigate these issues, sparse queries are proposed as a resource-efficient alternative to global queries. This concept has inspired research into dynamic sparse attention mechanisms, such as Bi-Level Routing Attention. This approach partitions the input feature map into distinct, non-overlapping regions and uses linear mapping to produce the query, key, and value. An adjacency matrix representing region-to-region affinities is computed by multiplying the region-level query with the transposed region-level key through matrix operations. The routing index matrix, which preserves the top-k connections for each region, is utilized to achieve fine-grained token-to-token attention. The dynamic sparse attention module is ultimately integrated into the backbone network at its 11th layer. This integration includes combining two feature vectors, applying depthwise separable convolution, performing layer normalization, and conducting multilayer perceptron computations. In this context, Q , K,and V refer to the query, key, and value, respectively.$W^q, W^k, W^v \in \mathbb{R}^{(C\times C)}$ The projection weights for the query, key, and value are represented accordingly. The corresponding calculation is provided in Equation (2):

$$Q = X^r W^q, K = X^r W^k, V = X^r W^\nu \quad (2)$$

Constructing a directed graph to determine the regions that should be concerned with each given region, we first derive the region-level queries Q^r and K^r by applying Q and the K to the mean value of each region ,which have dimension $\mathbb{R}^{(S^2\times S^2)}$. Then, by computing the matrix multiplication between Q^r and the transposed K^r , we obtain the adjacency matrix A^r of the region-to-region affinity graph with dimension $\mathbb{R}^{(S^2\times S^2)}$.The computation of the adjacency matrix for inter-region correlation can be expressed as shown in Equation (3):

$$A^r = Q^r(K^r)^T \quad (3)$$

In the neighbourhood matrix, the entry A^r is used to measure how semantically related two regions are. The indexes of the top-k connections are retained row by row using the routing index matrix $I^r \in \mathbb{N}^{S^2 \times k}$ Using the region-to-region routing index matrix I^r, fine-grained token-to-token attention can be computed. The ith row of matrix I^r contains the indexes of the first k most relevant regions of the ith region, which is calculated as shown in Equation (4):

$$I^r = topkindex(A^r) \tag{4}$$

Using the region-to-region routing index matrix I^r, fine-grained token-to-token attention can be computed. For each query token in region i, the key-value pairs in the concatenation of all k routing regions located in the index set $I^r_{(i,1)}, I^r_{(i,2)}, \ldots, I^r_{(i,k)}$ are processed Since these routing regions are scattered over the entire feature graph, in order to implement this step efficiently, the tensor of the keys and values needs to be collected first and computed as shown in Equations (5) and (6):

$$K^g = gather(K, I^T) \tag{5}$$

$$V^g = gather(V, I^T) \tag{6}$$

The above formulation uses an attention operation on the collected (gather) key-value pairs and introduces a local context augmentation term LCE(V), where LCE(V) is parameterised using a depth-separable convolution with a convolution kernel size of 5. The computation is shown in Equation (7):

$$O = Attention(Q, K^g, V^g) + LCE(V) \tag{7}$$

The module employs a two-level routing attention mechanism that is incorporated into Layer 11 of the backbone network to enhance the model's attention to critical target details, thereby improving detection accuracy.

3.2. Optimisation of neck network

Drawing on the EfficientCepBiPAN idea of YOLOV6 [39], a new neck network is designed, which includes two key components: the MPFusion module and the CepBlock module.The MPFusion module is optimised for the up- and down-sampling part of the original model, which incorporates an attention mechanism before the up- and down-sampling, and the three adjacent layers in a cascade operation to fuse the low-level features in the trunk to the high-level features in the neck, so that more accurate position signals are retained in the process of feature fusion, and efficient fusion of multi-scale feature maps and large and small target information is achieved. This process not only strengthens the model's capacity to detect targets but also improves its comprehension of image content.The design of the MPFusion module helps to cope with the scale differences of diverse targets in the real scene, so as to capture the target's features in a more comprehensive way. Inspired by RepBlock, the CepBlock module is designed, which is mainly optimised for the large perceptual field of the model. During the training phase, a 5×5 convolutional kernel is introduced to expand the perceptual field while maintaining the network's depth. The four branches in this phase are employed separately for feature extraction, with each branch undergoing a distinct reparameterization process. Specifically, the 1×1 convolution is reconfigured using padding with 3×3 and 5×5 convolutional kernels, with multiple instances of the 5×5 kernel applied. convolution kernel, 5 × 5 convolution kernel through the weighted average of neighbouring weights compressed into a 3 × 3 convolution, there is no convolution kernel of the residual channel to construct a class of convolutional input and output, that is, multiply a unit matrix can be, after the convolution layer and the BN layer fusion of the addition operation and then the output.CepConv is a 3 × 3 convolutional and the LeakyReLU activation function of the stack, Leaky ReLU can effectively solve the 0-gradient problem in the case of negative input, compared to the ordinary convolution block, less BN layer, the core idea is the fusion of Conv2d and depth-separable convolution, and finally directly add the parameters of these three convolutional layers to fuse them into an equivalent 3×3. Since 3×3 convolution has a high degree of optimisation on mainstream GPUs and CPUs, and has a high computational density, this design can greatly accelerate the inference speed. In the inference stage, the CepVGG block is transformed into CepConv, which can effectively accelerate the inference process using single branching. The MPFusion and CepBlock modules collaborate effectively, complementing each other to enhance the accuracy of information localization during the neck network's feature fusion process. The feature representation capability is enhanced by strengthening feature interactions and filtering out irrelevant information, allowing the model to better capture and understand target features. This improvement boosts the model's target discrimination ability. Additionally, the inference speed is increased without compromising accuracy, offering a practical approach for real-time target detection tasks. The designs of the two modules are illustrated in Figure 2, parts (a) and (b).

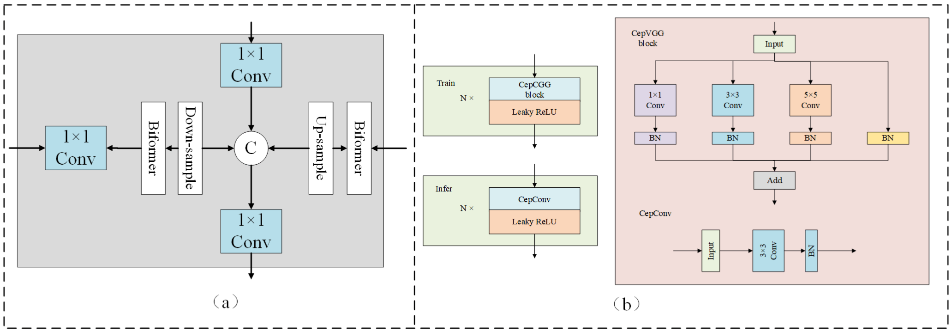


Fig. 2. The structure of MPFusion and CepBlock

3.3. Lightweight detection head reconfiguration

In the redesigned YOLOv8, the detection head has been optimized with a focus on lightweighting, aiming to overcome challenges related to model storage and execution. The original YOLOv8's detection head adopts a decoupled head structure and has been changed from Anchor-Based to Anchor-Free by removing the objectness branch and adopting two parallel branches, which are responsible for the extraction of category features and positional features respectively. However, this results in an increased parameter count, which significantly raises the demand for storage and computational resources. Our new detection head design maintains high accuracy while keeping lightweight. Specifically, we have designed the original 3×3 convolution of the two branches to merge and share parameters, while employing a layer of 1x1 convolution for both classification and localisation tasks. This design helps decrease the model's parameter count, which in turn lowers storage demands, making the model more efficient to deploy and operate. By employing parameter sharing and applying optimization techniques, we effectively improve the overall performance of the model. The new detection head structure is shown in Figure 3, with two parallel branches performing classification and localisation tasks through a layer of 1x1 convolution, which is designed to remain lightweight while still being able to quickly process edge feature information and extract classification features. Compared to the original detection head of YOLOv8, our design reduces the storage footprint while still ensuring high detection accuracy. This improvement not only makes the model more suitable for resource-limited environments, but also accelerates the training and inference speed and improves the overall performance.

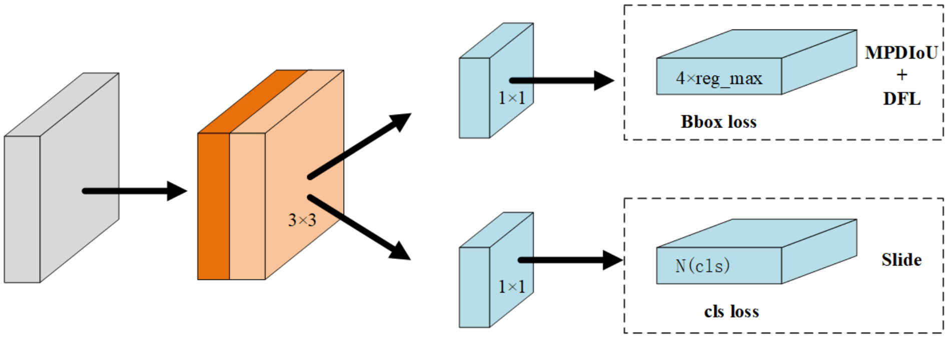


Fig. 3. The structure of the lightweight detection head

3.4. Optimisation of loss function

YOLOv8 adopts an anchorless design with a significant change in the loss function compared to the YOLOv5 series. The optimization goal is divided into two primary aspects: regression and classification. The classification component utilizes the sample weighting function (Slide Loss), whereas the regression process relies on the Distributional Keypoint Loss (DFL) along with the bounding box regression loss (MDPIoU). The complete loss function calculation is shown in Equation (8):

$$F_{loss} = \alpha_1 F_{Slideloss} + \alpha_2 F_{DFL} + \alpha_3 F_{MPDIoU} \tag{8}$$

To tackle the problem of sample imbalance in target detection tasks, Slide Loss has been introduced. Slide Loss primarily aims to balance samples with different difficulty levels by dynamically modifying their weights. The difficulty for each sample is evaluated based on the IoU values calculated between the predicted bounding boxes and the ground truth. To minimize the inclusion of additional hyperparameters, the average IoU value across all bounding boxes is used as the threshold, denoted as μ with IoU values below μ are classified as negative, while those with IoU values above μ are categorized as positive. The calculation process is detailed in Equation (9):

$$\mathrm{F(x)} = \begin{cases} 1, x \leq \mu - 0.1 \\ e^{1-\mu}, \mu < x < \mu - 0.1 \leftarrow \\ e^{1-x}, x \geq \mu \end{cases} \tag{9}$$

To fully utilize samples with ambiguous classifications and those located near decision boundaries, Slide Loss is introduced. This method addresses challenging samples by categorizing them as positive or negative based on the parameter μ. Additionally, the Slide weighting function assigns greater importance to boundary samples by giving them higher weights, thereby enhancing the model's attention to classification-challenging cases.

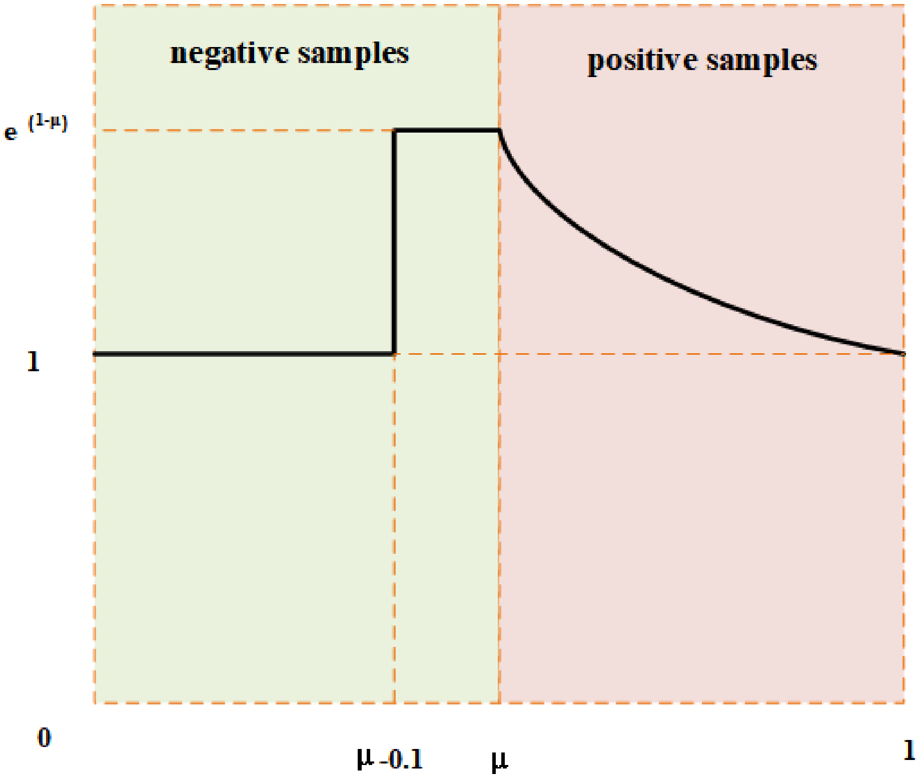


Fig. 4. Slide weighting function image

Slideloss [40], illustrated in Figure 4, represents a sliding loss function that adaptively determines the threshold parameters: μ for positive samples and μ for negative samples.By setting higher weights around μ the loss of difficult, incorrectly categorised examples can be increased,which approach significantly enhances the model's classification performance, particularly for boundary cases and challenging samples.

There are many targets in the dataset with the same aspect ratio but inconsistent scaling. To solve this problem, this paper introduces MDPIoU [41] as an optimisation method for bounding box regression loss. For any convex shapes A and B, the widths and heights are denoted as w and h . The coordinates represent the upper-left and lower-right corner points of shapes A and B, respectively. $(x_1^A, y_1^A), (x_2^A, y_2^A)$ and $(x_1^B, y_1^B), (x_2^B, y_2^B)$ respectively,The derivation process of MDPIoU is shown in Equations (10) to (12):

$$d_1^2 = (x_1^B - x_1^A)^2 + (y_1^B - y_1^A)^2 \tag{10}$$

$$d_2^2 = (x_2^B - x_2^A)^2 + (y_2^B - y_2^A)^2 \tag{11}$$

$$MDPIoU = \frac{A \cap B}{A \cup B} - \frac{d_1^2}{w^2 + h^2} - \frac{d_2^2}{w^2 + h^2} \tag{12}$$

In the training phase, the set of predicted values for each bounding box predicted by the model is forced to approximate the set of true bounding boxes by minimising the loss function, hence for MPDIoU based loss function is defined as shown in Equation (13):

$$L_{MDPIoU} = 1 - MDPIoU \tag{13}$$

The coordinates of the four points can be used to derive all components of the current bounding box regression loss function. The transformation steps are outlined in Equations (14) to (18):

$$\begin{aligned} C_x &= \max(x_2^{gt}, x_2^{prd}) - \min(x_1^{gt}, x_1^{prd}) \\ C_y &= \max(y_2^{gt}, y_2^{prd}) - \min(y_1^{gt}, y_1^{prd}) \\ |C| &= C_x \cdot C_y \end{aligned} \tag{14}$$

$$x_c^{gt} = \frac{x_1^{gt} + x_2^{gt}}{2}, \quad y_c^{gt} = \frac{y_1^{gt} + y_2^{gt}}{2} \tag{15}$$

$$x_c^{prd} = \frac{x_1^{prd} + x_2^{prd}}{2}, \quad y_c^{prd} = \frac{y_1^{prd} + y_2^{prd}}{2} \tag{16}$$

$$w^{gt} = x_2^{gt} - x_1^{gt}, \quad h^{gt} = y_2^{gt} - y_1^{gt} \tag{17}$$

$$w^{prd} = x_2^{prd} - x_1^{prd}, \quad h^{prd} = y_2^{prd} - y_1^{prd} \tag{18}$$

Here, $|C|$ denotes the area of the smallest rectangle that encloses both B_{gt} and P_{rd} , (x_c^{gt}, y_c^{gt}) and (x_c^{prd}, y_c^{prd}) represent the coordinates of the centre points of the groundtruth bounding box and the prediction bounding box, respectively, ${}_{w}gt$ and ${}_{h}gt$ represent the width and height of the groundtruth bounding box, ${}_{W}prd$ and ${}_{h}prd$ represent the width and height of the prediction bounding box.According to From the coordinates of the upper-left and lower-right points, all factors present in existing loss functions—such as non-overlapping areas, distances between centroids, and variations in width and height—can be derived, as shown in Equations (16) to (18).This demonstrates that the MDPIoU utilized in our approach is both thoughtfully designed and computationally efficient. Figure 5 illustrates the loudness factor.

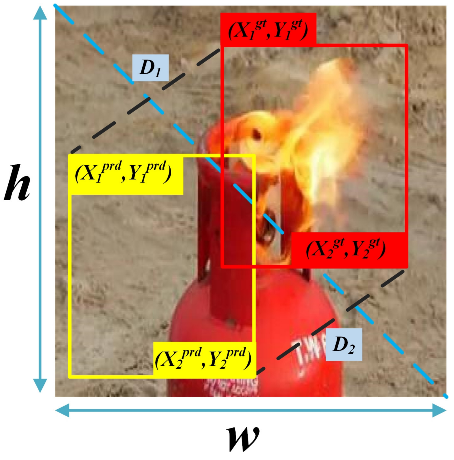


Fig. 5. Factors affecting MDPIoU

4. Experiments

4.1. Experimental environment

The environment and hardware platform parameters for the training phase of the experiment are shown in Table 1:

Table 1. Experimental environment configuration

parameter	configure
CPU	Intel Xeon Silver 4214R
GPU	NVIDIA GeForce RTX 3080 Ti
operating system	Ubuntu 18.04.5
Architecture	torch-1.9.0+cu111

During the training process, we set the key parameters according to Table 2.

4.2. Introduction to datasets

To comprehensively assess the performance of our enhanced algorithm, we chose two publicly accessible datasets: one from the Roboflow platform and the D-Fire fire detection dataset. Both datasets offer extensive annotations for flames and smoke across various scenarios, enabling a robust assessment of our algorithm's performance in complex environments.

The D-Fire dataset consists of 21,527 images featuring flames and smoke, averaging 2.52 bounding boxes per image. Conversely, in the categories labeled as ”Smoke” and

Table 2. Experimental parameter configuration

parameter	settings
Epochs	150
Momentum	0.937
Initial learning rate	0.01
Final learning rate	0.01
Weight decay	0.005
Input image size	640 × 640
Optimizer	SGD
Data enhancement	Mosaic
Box Loss decay	7.5
Cls Loss decay	0.5
Batch size	32

”Fire and Smoke,” the average number of smoke-labeled frames is 1.13 bounding boxes per image. Altogether, the dataset comprises 26,557 bounding boxes, with 11,865 annotated as smoke and 14,692 identified as fire.

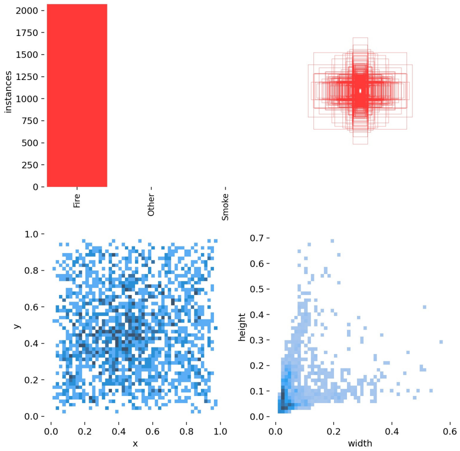


Fig. 6. Label distribution of the dataset

The dataset from the Roboflow platform features images of fire smoke captured in diverse environments, including both indoor and outdoor scenarios. The dataset is split randomly into three subsets: training, validation, and testing, following a 7:2:1 ratio. Specifically, the training set consists of 4,620 images, the test set includes 1,320 images, and the validation set comprises 660 images. There are three distinct types of annotations, as illustrated in Figure 6. The first subfigure highlights the quantity of various fire-related objects. The second subfigure presents the bounding box sizes, with all their center points aligned at a single location, suggesting a prevalence of small object regions within the dataset. The third subfigure depicts the distribution of bounding box center coordinates, revealing that the majority of center points are clustered around the central region of the image.Finally, the fourth subfigure presents a scatter plot showing the widths and heights of bounding boxes. The darkest area in the lower-left corner highlights that the dataset primarily consists of small objects.

From the analysis of the dataset, it can be concluded that it predominantly consists of numerous small objects with a dense yet uneven distribution. Compared to traditional datasets used in computer vision tasks, this dataset is significantly larger and includes a variety of scales, scenes, and angles, making it more challenging than standard computer vision datasets. To enhance the model's performance and refine its development, this paper employs data augmentation techniques such as cropping, scaling, and color perturbation to improve data quality and increase diversity. The YOLOv8n model served as the baseline, with several ablation experiments performed to assess how each improvement strategy affected its performance, leading to the identification of the best configuration. Moreover, mosaic data augmentation was utilized in the last 10 training epochs to enhance the speed of model convergence.

4.3. Evaluation indicators

This experiment uses both the model itself metrics and TIDE metrics to measure the performance of the model in this paper at the same time, calculated as shown in Equations (19) to (21):

$$Precision = \frac{TP}{TP + FP} \tag{19}$$

$$Recall = \frac{TP}{TP + FN} \tag{20}$$

$$AP = \int_0^1 \text{Precision(Recall)d(Recall)} \tag{21}$$

The mean Average Precision (mAP) across all categories is derived by calculating the weighted average of the AP values for each sample category. This metric evaluates the model's detection performance across all categories and is computed as illustrated in Equation (22):

$$mAP = \frac{1}{K}\sum_{i=1}^{K} AP_i \tag{22}$$

APi in Equation (22) denotes the relationship between the value and the value of the category index.K denotes the number of categories of the samples in the trained dataset, and the value in this paper is 3.

To compare model runtime, this paper adopts Frames Per Second (FPS) as the performance metric. FPS, which indicates the number of image or video frames the model can process each second, is used to evaluate runtime efficiency. In order to further capture the more valuable error distributions in mAP, all FPs and FNs are grouped into six types, and FPs and FNs can be paired in some cases, and IoUmax is used to denote the overlap of the maximum IoU of an FP with the ground truth of a given category, with the foreground

IoU threshold denoted as tf and the background threshold denoted as tb. The following are the definitions of the six types of errors and the rules of determining them .

Classification error (**Cls**), i.e., for misclassification, IoU$_{\text{max}} \geq t_f$ (i.e., localisation is correct but misclassified).

Localisation error (**Loc**), i.e., for correct classification, $t_b \leq$ IoU$_{\text{max}} \leq t_f$ (i.e., classification is correct but localisation is incorrect).

Both classification and localisation error (**Cls+Loc**), i.e., for misclassification, $t_b \leq$ IoU$_{\text{max}} \leq t_f$ (i.e., classification is incorrect and incorrectly localised).

Duplicate Detection Error (**Duplicate**), i.e., correctly classified and GTIoU$_{\text{max}} \geq t_f$, but another higher-scoring test has already matched the GT (i.e., would have been correct if not for the higher-scoring test).

Background Error (**Bkgd**), i.e., IoU$_{\text{max}} \leq t_b$ for all GTs (detecting the background as the foreground).

Undetected GT errors (**Missed**), i.e., all undetected ground truth (FN) not covered by classification or localisation errors.

Smaller values of the above six metrics represent smaller errors and superior model performance.

4.4. Experiment 1:experimental results and analysis

Optimal improved position experiments with dynamic sparse attention The first few layers of the backbone network usually have smaller feature map size and depth, and it may be relatively faster to apply the attention mechanism at these layers to reduce the impact on the overall inference speed. To achieve optimal performance and support subsequent ablation experiments, comparative experiments were carried out in this study. Specifically, by enhancing the neck network, the dynamic sparse attention module was integrated into different layers of the backbone network, with the resulting experimental data summarized in Table 3. Specifically, dynamic sparse attention modules were applied individually to layers 3, 5, 7, 9, and 11 of the backbone network. The data in Table 3 demonstrate that the dynamic sparse attention11 model achieves superior detection performance, exhibiting a notable accuracy improvement and the lowest scores across all six error type indices when compared to other models. As a result, dynamic sparse attention11 was chosen as part of this paper's proposed improvement strategy. The experimental results demonstrate that incorporating the dynamic sparse attention module allows the network to more effectively identify and highlight important features within the image.

Table 3. Comparison of YOLOv8 with different dynamic sparse attention configurations

Model	AP	AR	mAP50	mAP50-95	FPS/bs32	Size/MB	GFlops	Cls	Loc	Both	Dupe	[c]Bkg	[c]Miss
YOLOv8+dynamic sparse attention3	0.951	0.912	0.961	0.879	**419**	**2.86**	3.01	0.11	1.38	**0.02**	0.09	0.76	0.55
YOLOv8+dynamic sparse attention5	0.948	0.921	0.959	0.881	412	**2.86**	3.01	0.12	1.45	0.06	0.08	0.73	0.61
YOLOv8+dynamic sparse attention7	0.957	0.915	0.948	0.880	409	**2.86**	3.01	**0.09**	1.31	0.03	**0.06**	0.88	0.58
YOLOv8+dynamic sparse attention9	0.946	**0.935**	0.964	0.882	406	**2.86**	3.01	0.11	1.35	0.04	**0.06**	0.86	0.71
YOLOv8+dynamic sparse attention11	**0.971**	0.919	**0.967**	**0.888**	400	**2.86**	**3.01**	**0.09**	**1.29**	**0.02**	**0.06**	**0.67**	**0.49**

Analysing Table 3 shows that the performance of adding dynamic sparse attention to the backbone network is superior, with most of the metrics improved, the recall has decreased probably because the features extracted at layer 11 are not discriminative enough, and despite the decrease in recall, the overall increase in performance shows the positive impact of the addition of dynamic sparse attention to the 11th layer of the backbone network. The improvement in other metrics suggests that it has advantages in improving detection precision and accuracy.

Experiment 2: ablation experiment This paper further explores the impact of different improvement strategies to evaluate the enhancement of the improvement model by gradually adding each module. The specific programmes are as follows.

(1) YOLOv8n: original model for target detection.

(2) YOLOv8n+A: Replace the original BCElosss with Slideloss.

(3) YOLOv8n+B: Use MPDIoU to replace the original CIoU.

(4) YOLOv8n+D: Improve the neck network.

(5) YOLOv8n+C+D: Improve the backbone network by adding dynamic sparse attention to (4).

(6) YOLOv8n+C+D+E: using optimised lightweight detection head on top of (5).

(7) YOLOv8n+C+E: removing improvements to the neck network from (6).

(8) YOLOv8n+A+B+C+D+E: combining (2),(3),and (6).

The detection performance and model parameters of each model are listed in the Table 4, where YOLOv8n is the baseline model. The tables and images are briefly analysed below: YOLOv8n:The baseline model, with most of its metrics, is only ranked in the middle of the pack in the ablation experiments, which suggests that even if a model does not have a high number of model parameters in it, it still leads to a long inference time due to the redundant network layers. The final FPS index of the improved model reaches 434, which can ensure the real-time requirement in real deployment. YOLOv8n+A:Using the strategy of improvement A on top of the baseline model, the recall and map50 scores of the model are slightly improved to 0.975 and 0.944 respectively, which demonstrates the reasonableness of the improvement in solving the problem of imbalance between difficult and simple samples, while the the decrease in Cls indicates a substantial improvement in the classification error. YOLOv8n+B: Adopting the B improvement strategy based on the baseline model, the map50-95 , AR and AP of this model slightly decreased respectively, which may be due to the large differences between targets affecting the overall performance of recall and accuracy. However, map50 is improved, while its Loc and Miss metrics are decreased indicating the superiority of the frame loss improvement. YOLOv8n+D:The D improvement strategy was used on the basis of the baseline model, which fused feature representations of different layers, reduced the number of channels and convolutional kernel size of some layers, and increased the convolutional layers, with the number of parameters and the amount of computation reduced by 10% and 11%, respectively. All other indexes are improved, which indicates that the model after the improved neck network makes a substantial improvement in the detection performance and running speed of the model. YOLOv8n+A+B+C+D+E: This model has improved all indexes compared with YOLOv8n, and the error rate is obviously reduced, which indicates that there is a large improvement in the detection performance, and the overall performance is higher, therefore, we identified this model as the optimal improved model, which shows that the improvement of the baseline model is feasible and effective and reasonable considering the accuracy and speed of special equipment and detection scenarios.

Table 4. Comparison of ablation experiment indicators

Models	AP	AR	mAP50	mAP50-95	FPS/bs32	Size/MB	GFlops	Cls	Loc	Both	Dupe	Bkg	Miss
YOLOv8n	0.968	0.938	0.970	0.898	400	2.86	8.1	0.18	0.77	0.01	0.03	0.71	0.42
YOLOv8n+A	0.975	**0.944**	**0.977**	0.902	400	2.86	8.1	**0.03**	0.86	0.02	0.04	0.74	0.16
YOLOv8n+B	0.966	0.928	0.971	0.890	384	2.86	8.1	0.08	0.70	0.01	0.07	0.86	0.27
YOLOv8n+D	0.968	0.937	0.972	0.897	434	**2.57**	7.2	0.10	0.75	0.01	0.05	0.68	0.13
YOLOv8n+C+D	0.971	0.919	0.967	0.888	384	2.82	7.2	0.09	1.29	0.02	0.06	0.67	0.49
YOLOv8n+C+D+E	0.959	0.922	0.961	0.880	416	2.88	6.5	0.20	**0.72**	0.01	0.11	1.01	0.09
YOLOv8n+C+E	0.966	0.935	0.971	0.893	**454**	3.91	8.1	0.08	0.78	0.01	0.06	0.72	0.21
YOLOv8n+A+B+C+D+E	**0.981**	0.940	0.974	**0.907**	434	2.88	**6.5**	0.08	0.76	**0.01**	0.04	**0.49**	**0.08**

Experiment 3:comparative experiments n the field of target detection, deep learning methods are classified into level 1 and level 2 categories, distinguished by their anchor generation mechanisms. In real engineering scenarios, real-time processing of fire and smoke images is more in line with practical needs. Therefore, in order to combine both accuracy and hardware dependency considerations, it is more practical to choose the level 1 target detection method. In this experiment, we selected the YOLO series as the object of comparison test, including advanced and general models such as YOLOv5-s, YOLOv3, YOLOv3-tiny and YOLOv6. These models have been widely used in a variety of embedded scenarios and published in several papers. To emphasize the advantages of the models used in this experiment, we selected the enhanced versions developed in this study for comparison. Comparison experiments are conducted with YOLOv5-s, YOLOv3, YOLOv3-tiny and YOLOv6. To ensure fairness, no pre-training weights were used in all model training processes.The outcomes of the comparative experiments are presented in detail in Table 5 and Table 6. Review these tables to gain insights into the findings.

(1) YOLOv3 has the highest number of parameters, totaling 103,666,553 bytes. While its overall performance metrics are impressive, its real-time speed is limited to 66 FPS, making it unsuitable for the real-time detection tasks required in this study.

(2) The YOLOv5-s model is highly lightweight. however, its precision is low on this dataset, suggesting a high false detection rate. Despite this, it demonstrates better performance in terms of recall. The primary reason for these observations lies in the complexity of the dataset's background and the significant variation in target sizes. These factors negatively impact the precision, but the improved version of the model is better suited for the task at hand.

(3) YOLOv3-tiny is also a lightweight model, most of the indicators are low, compared to the performance of YOLOv5-s is still insufficient, which is also YOLO after several versions of iteration much led to make YOLOv5 performance has been greatly improved.

(4) YOLOv6 has the lowest recall rate, indicating that the model has a certain leakage rate, fire smoke detection task, not only requires high real-time, in the leakage rate requirements are also more stringent, the poor performance of this important indicator, making the model is not suitable for the task in this paper.

(5) Our model comprehensive comparison of other models in the same series, the overall performance of the best, especially in the FPS this indicator is excellent, while the model in basically does not change the number of far away model parameters, in the deployment difficulty and real-time and other convenient to meet the real-life engineering needs, while the model also has good robustness, accuracy and practicality.

Table 5. Comparison of experimental indexes of data sets of each model on roboflow

Models	AP	AR	mAP50	mAP50-95	FPS/bs32	Size/MB	GFlops	Cls	Loc	Both	Dupe	Bkg	Miss
YOLOv5-s	0.963	0.962	0.972	0.894	270	2.86	7.2	0.11	1.02	0.01	0.03	0.86	0.71
YOLOv3	0.968	0.961	0.981	0.920	66	98.86	282.2	0.44	0.83	0.23	0.19	0.65	0.28
YOLOv3-tiny	0.959	0.935	0.973	0.895	285	4.04	11.8	0.58	1.19	0.19	0.21	0.67	0.25
YOLOv6	0.955	0.917	0.960	0.853	277	11.56	18.9	0.16	1.35	0.10	0.13	0.66	0.32
TOOD	0.972	0.931	0.978	0.901	386	31.8	125.9	0.12	0.86	0.03	0.08	0.59	0.16
Our	0.981	0.938	0.974	0.907	434	2.88	6.5	0.08	0.83	0.01	0.06	0.49	0.08

Table 6. Comparison of the experimental metrics for each model on the D-Fire dataset

Models	AP	AR	mAP50	mAP50-95	FPS/bs32	Size/MB	GFlops	Cls	Loc	Both	Dupe	Bkg	Miss
YOLOv5-s	0.761	0.731	0.772	0.458	270	2.86	7.2	0.14	1.16	0.04	0.05	0.92	0.84
YOLOv3	0.777	0.759	0.781	0.481	66	98.86	282.2	0.49	0.95	0.29	0.24	0.74	0.36
YOLOv3-tiny	0.765	0.721	0.773	0.453	285	4.04	11.8	0.68	1.27	0.24	0.24	0.79	0.23
YOLOv6	0.753	0.711	0.762	0.414	277	11.56	18.9	0.19	1.44	0.14	0.17	0.73	0.42
TOOD	0.768	0.727	0.766	0.451	386	31.8	125.9	0.13	0.78	0.05	0.07	0.62	0.19
Our	0.786	0.728	0.789	0.467	434	2.88	6.5	0.12	0.93	0.03	0.09	0.55	0.17

Experiment 4:individual comparison with YOLOv8 To verify the impact of the enhanced model on detection performance, we carried out a comparative experiment between the improved model and the baseline model, YOLOv8n. Figure 7 shows the trend analysis of precision, recall, mAP50 and mAP50-95 for AMMF-Detection (orange curve) and YOLOv8n (blue curve) on the validation dataset. The metrics show rapid improvement throughout the iterations and gradually approach stable values, and AMMF-Detection ends up with higher convergence values.

Experiment 5:visualisation and analysis The interpretability of deep learning models is a key issue limiting their application and development, thus becoming a research hotspot in artificial intelligence. We evaluated the model performance in terms of confusion matrix, feature map visualisation and inference experiments through comparative experiments. The confusion matrix intuitively reflects the classification accuracy, the feature map visualisation demonstrates the distribution of the model's attention to the target, and the inference experiment verifies the model's generalisation ability and robustness on a new fire image dataset. These methods comprehensively reveal the strengths and limitations of the model.

As illustrated in Figure 8, the diagonal indicator region within the confusion matrix for AMMF-Detection is notably higher compared to YOLOv8n. This suggests an improved capability of our model in accurately classifying target categories. Also, the proportion of objects whose backgrounds are judged to be flames has been reduced, which means that the improved model reduces the miss detection rate for this category, but the accuracy for smoke has been reduced, which is due to the complexity and polygonal shape of the smoke itself. A comparison of the heat maps reveals that both models exhibit a certain degree of false detection, with the flame frequently being misclassified as background. To address this, we selected three representative images to visualize their feature maps.

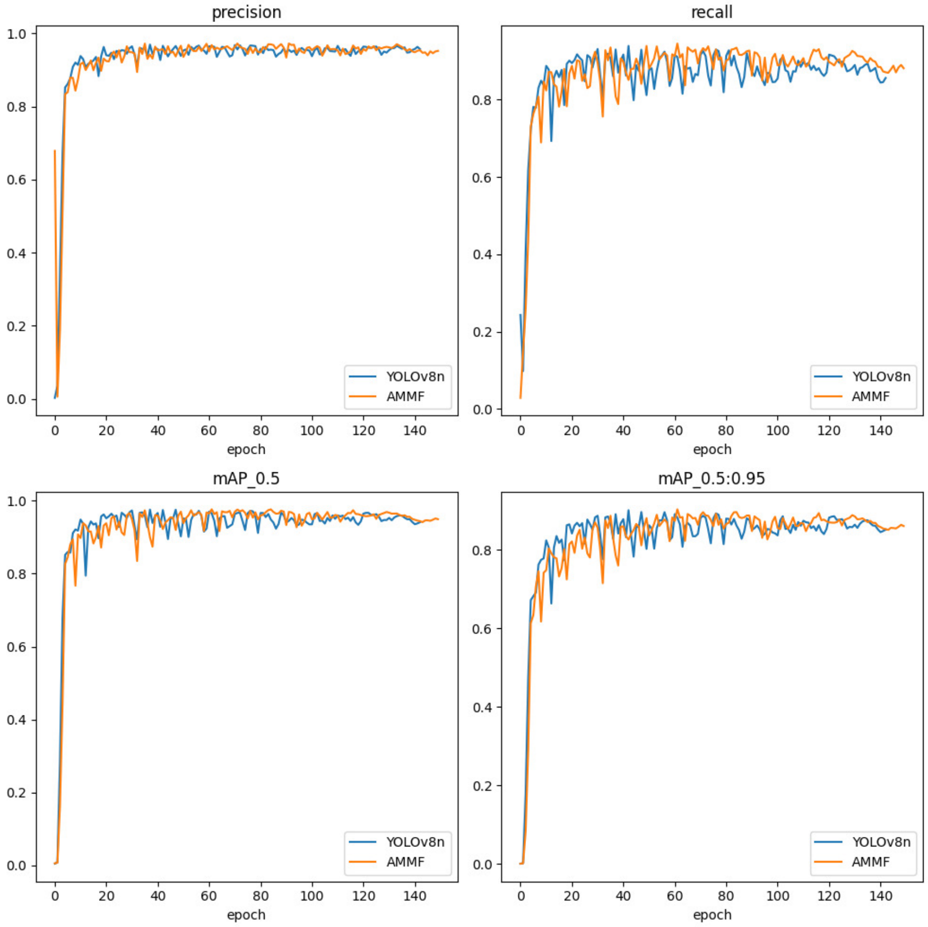
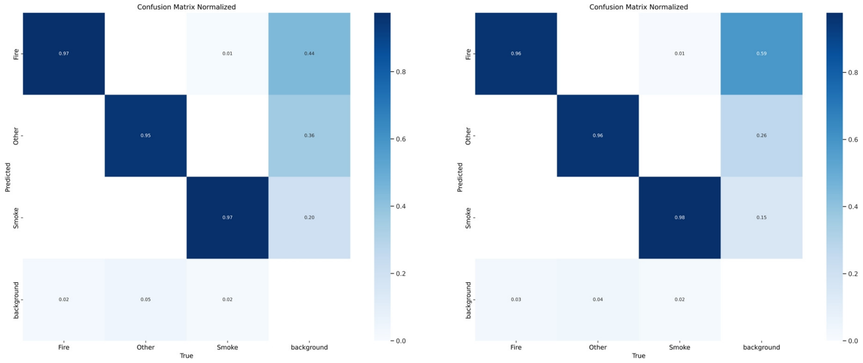


Fig. 7. Trend analysis of the indicators of the model training process

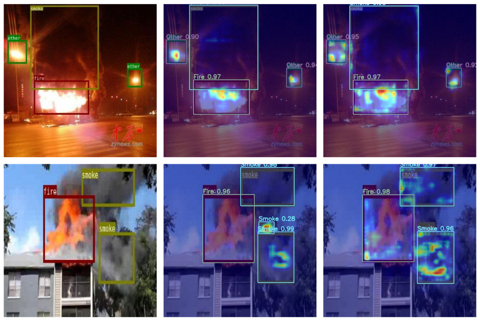
This visualization offers a clearer and more intuitive way to observe the model's focus, which is highly beneficial for improving the overall detection performance. In the same image, it is evident that AMMF-Detection focuses on a broader range, including similar target objects. The original model, however, demonstrates a notable false detection rate and insufficient confidence levels for the detected targets. In contrast, the improved model exhibits a more precise focus, reduces the error-prone regions, and achieves higher confidence scores. A detailed comparison of the feature maps is presented in Figure 9.

To evaluate the generalization ability of the proposed method, a diverse dataset of images was collected, and inference experiments were performed. These images feature numerous small objects, posing significant challenges for detection. The inference experiment results, presented in Figure 10, include detection outcomes for indoor and outdoor targets of varying sizes. As shown in Figure 10, our method achieves high-quality detection performance across diverse and complex environments. The model demonstrates



YOLOv8n AMMF-Detection

Fig. 8. Comparison of confusion matrices



original YOLOv8n AMMF-Detection

Fig. 9. Comparison of heatmap visualizations

minimal instances of missed detections, effectively showcasing the adaptability and robustness of the proposed approach across different scenarios.

From the experimental results, it can be concluded that the proposed approach successfully identifies target objects in various and challenging environments, encompassing both indoor and outdoor scenarios. Additionally, it demonstrates the capability to handle target objects of various sizes. These findings highlight the method's strong adaptability and generalization capabilities, making it suitable for application in numerous real-world scenarios.

5. Conclusion

In this paper, we introduce an effective and streamlined fire detection model, termed AMMF-Detection, which is an optimization based on YOLOv8. This model addresses the challenges of bounding box optimization and sample imbalance in fire detection tasks by incorporating the MPDIoU bounding box distance metric and the SlideLoss classification loss function. Moreover, the integration of the dynamic sparse attention mechanism

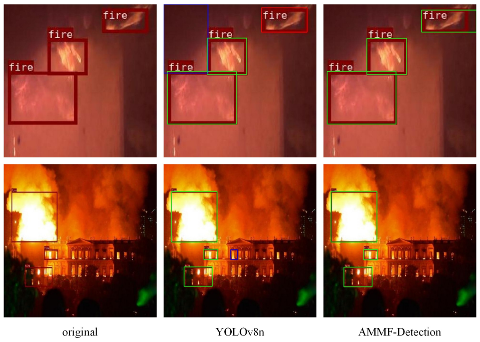


Fig. 10. Inference experiment results

improves the model's ability to capture global contextual information and understand image content. Additionally, the neck network is redesigned by incorporating the CepBlock module and the MPFusion module, further refining the overall architecture. Finally, the detection head is restructured to achieve a lightweight design, reducing the model's computational complexity. Experimental findings reveal that the optimized model attains an average precision of 97.4% at a 50% recall rate and 90.7% across a recall range of 50% to 95%.Additionally, the frames per second (FPS) metric improves from 400 to 434. The fire detection model presented in this paper holds significant practical applications. Future studies can further optimize the model's performance and validate its application in various other domains and tasks.The reconstruction of the neck network in our improved model introduces complexity and a relatively high number of feature fusion steps, leading to variations in model size and inference time. There is still potential to optimize the model's computational consumption. Future work will prioritize investigating distillation and pruning techniques to compress the model's parameters and structure, aiming to strike a balance between complexity and performance, thereby improving its efficiency and overall performance.

Acknowledgments. This work was supported by the National Natural Science Foundation of China (Grant NO.62076006), the Opening Foundation of the State Key Laboratory of Cognitive Intelligence, iFLYTEK (Grant NO.COGOS-2023HE02), and the University Synergy Innovation Program of Anhui Province (Grant NO.GXXT-2021-008).

References

1. Y. Wang, Y. Han, Z. Tang, et al. A Fast Video Fire Detection of Irregular Burning Feature in Fire-Flame Using in Indoor Fire Sensing Robots. *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–14, 2022.
2. L. Zhang, C. Lu, H. Xu, et al. MMFNet: Forest fire smoke detection using multiscale convergence coordinated pyramid network with mixed attention and fast-robust NMS. *IEEE Internet of Things Journal*, 2023.
3. M. Mueller, P. Karasev, I. Kolesov, et al. Optical flow estimation for flame detection in videos. *IEEE Transactions on Image Processing*, vol. 22, no. 7, pp. 2786–2797, 2013.

4. W. Liu, D. Anguelov, D. Erhan, et al. SSD: Single shot multibox detector. In *Computer Vision–ECCV*, Springer, pp. 21–37, 2016.
5. J. Redmon, S. Divvala, R. Girshick, et al. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, 2016.
6. J. Redmon and A. Farhadi. YOLO9000: Better, faster, stronger. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7263–7271, 2017.
7. C. Y. Wang, I. H. Yeh, H. Y. Mark Liao.Yolov9: Learning what you want to learn using programmable gradient information.In *European Conference on Computer Vision*. Springer, Cham, 2025, pp. 1-21.
8. A. Wang, H. Chen, L. Liu, et al. Yolov10: Real-time end-to-end object detection.*arXiv preprint arXiv:2405.14458*, 2024.
9. W. Lv, S. Xu, Y. Zhao, et al. DETRs beat YOLOs on real-time object detection. *arXiv preprint arXiv:2304.08069*, 2023.
10. T. H. Chen, P. H. Wu, Y. C. Chiou. An early fire-detection method based on image processing. In *Proceedings of the 2004 International Conference on Image Processing (ICIP)*, vol. 3, pp. 1707–1710, IEEE, 2004.
11. N. I. binti Zaidi, N. A. A. binti Lokman, M. R. bin Daud, et al. Fire recognition using RGB and YCbCr color space. *ARPN Journal of Engineering and Applied Sciences*, vol. 10, no. 21, pp. 9786–9790, 2015.
12. V. Vipin. Image processing based forest fire detection. *International Journal of Emerging Technology and Advanced Engineering*, vol. 2, no. 2, pp. 87–95, 2012.
13. K. Dimitropoulos, P. Barmpoutis, N. Grammalidis. Spatio-temporal flame modeling and dynamic texture analysis for automatic video-based fire detection. *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 2, pp. 339–351, 2014.
14. W. Ye, J. Zhao, S. Wang, et al. Dynamic texture based smoke detection using Surfacelet transform and HMT model. *Fire Safety Journal*, vol. 73, pp. 91–101, 2015.
15. Y. Chunyu, F. Jun, W. Jinjun, et al. Video fire smoke detection using motion and color features. *Fire Technology*, vol. 46, pp. 651–663, 2010.
16. Z. Li, L. S. Mihaylova, O. Isupova, et al. Autonomous flame detection in videos with a Dirichlet process Gaussian mixture color model. *IEEE Transactions on Industrial Informatics*, vol. 14, no. 3, pp. 1146–1154, 2017.
17. S. Han, H. Mao, W. J. Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.
18. M. Courbariaux, Y. Bengio, J. P. David. BinaryConnect: Training deep neural networks with binary weights during propagations. In *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 28, 2015.
19. M. Courbariaux, I. Hubara, D. Soudry, et al. Binarized neural networks: Training deep neural networks with weights and activations constrained to +1 or -1. *arXiv preprint arXiv:1602.02830*, 2016.
20. I. Hubara, M. Courbariaux, D. Soudry, et al. Quantized neural networks: Training neural networks with low precision weights and activations. *Journal of Machine Learning Research*, vol. 18, no. 187, pp. 1–30, 2018.
21. J. Yim, D. Joo, J. Bae, et al. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4133–4141, 2017.
22. C. Szegedy, W. Liu, Y. Jia, et al. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9, 2015.
23. A. G. Howard, M. Zhu, B. Chen, et al. MobileNets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04825*, 2017.
24. A. Howard, M. Sandler, G. Chu, et al. Searching for MobileNetV3. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1314–1324, 2019.

25. K. He, X. Zhang, S. Ren, et al. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.
26. J. Redmon and A. Farhadi. YOLOv3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.
27. A. Bochkovskiy, C. Y. Wang, and H. Y. M. Liao. YOLOv4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.
28. R. Girshick, J. Donahue, T. Darrell, et al. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580-587, 2014.
29. R. Girshick. Fast R-CNN. In *Proceedings of the IEEE international conference on computer vision*, pages 1440-1448, 2015.
30. S. Ren, K. He, R. Girshick, et al. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137-1149, 2016.
31. K. He, G. Gkioxari, P. Dollár, et al. Mask R-CNN. In *Proceedings of the IEEE international conference on computer vision*, pages 2961-2969, 2017.
32. J. Miao, G. Zhao, Y. Gao, et al. Fire detection algorithm based on improved YOLOv5. In *2021 International Conference on Control, Automation and Information Sciences (ICCAIS)*, IEEE, pages 776-781, 2021.
33. M. Luo, J. Huang, X. Sun, et al. Small Target Forest Fire Recognition Method based on Deep Learning. In *2023 IEEE 3rd International Conference on Information Technology, Big Data and Artificial Intelligence (ICIBA)*, IEEE, volume 3, pages 593-597, 2023.
34. P. Li and W. Zhao. Image fire detection algorithms based on convolutional neural networks. *Case Studies in Thermal Engineering*, 19:100625, 2020.
35. S. Li, Q. Yan, and P. Liu. An efficient fire detection method based on multiscale feature extraction, implicit deep supervision and channel attention mechanism. *IEEE Transactions on Image Processing*, 29:8467-8477, 2020.
36. S. Majid, F. Alenezi, S. Masood, M. Ahmad, E. S. Gündüz, and K. Polat. Attention based CNN model for fire detection and localization in real-world images. *Expert Systems with Applications*, 189:116114, 2022.
37. J. Pincott, P. W. Tien, S. Wei, and J. K. Calautit. Indoor fire detection utilizing computer vision-based strategies. *Journal of Building Engineering*, 61:105154, 2022.
38. L. Zhu, X. Wang, Z. Ke, et al. dynamic sparse attention: Vision transformer with bi-level routing attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10323-10333, 2023.
39. C. Li, L. Li, Y. Geng, et al. YOLOv6 v3.0: A full-scale reloading. *arXiv preprint arXiv:2301.05586*, 2023.
40. Z. Yu, H. Huang, W. Chen, et al. YOLO-Facev2: A scale and occlusion aware face detector. *arXiv preprint arXiv:2208.02019*, 2022.
41. M. Siliang and X. Yong. MPDIoU: A loss for efficient and accurate bounding box regression. *arXiv preprint arXiv:2307.07662*, 2023.

Shunxiang Zhang received the Ph.D. degree from the School of Computing Engineering and Science, Shanghai University, Shanghai, China, in 2012. He is a Professor at the Anhui University of Science and Technology, Huainan, China. His current research interests include web mining, semantic search, and complex network

Meng Chen received her Bachelor's degree from Fuyang Normal University in 2023. She is currently a master's student in the School of Computer Science and Engineering, Anhui University of Science and Technology. Her current research interests include computer vision, image segmentation, object detection, 3D object detection, and multimodal networks.

Kuan-Ching Li is currently a Distinguished Professor at Providence University, where he also serves as the Director of the High-Performance Computing and Networking Center. He published more than 450 scientific papers and articles and is the coauthor or coeditor of more than 50 books published by well-known publishers. He is the Editor-in-Chief of IJCSE and IJES and serves as an associate editor for several leading journals. His research interests include parallel and distributed computing, big data, and emerging technologies. He is a fellow of the IET and a Senior Member of the IEEE.

Hua Wen received his Bachelor's degree from Huainan Normal College in 2023. He is currently a master's student in the School of Computer Science and Engineering, Anhui University of Science and Technology, China. His current research interests include multimodal sentiment analysis, sentiment extraction in multiple scenarios, sarcasm detection, and multimodal networks.

Liang Sun received the Bachelor's degree from Anhui University of Technology in 2024. He is currently a master's student in the School of Computer Science and Engineering, Anhui University of Science and Technology, China. His current research interests include multimodal sentiment analysis, large language modeling, graphic dialog generation, machine translation.

Received: December 25, 2024; Accepted: June 27, 2025.