Computer Science and Information Systems 19(3):1371–1388 https://doi.org/10.2298/CSIS220322037L

A Novel Art Gesture Recognition Model Based on Two Channel Region-Based Convolution Neural Network for Explainable Human-computer Interaction Understanding

Pingping Li[1] and Lu Zhao[2]

[1] School of Fine Arts, Zhengzhou Normal University
450000 Zhengzhou, China
910675024@qq.com
[2] School of Fine Arts, Yulin Normal University
537000 Yulin, China
zhaoluvip@163.com

Abstract. The application development of hot technology is both an opportunity and a challenge. The vision-based gesture recognition rate is low and real-time performance is poor, so various algorithms need to be studied to improve the accuracy and speed of recognition. In this paper, we propose a novel gesture recognition based on two channel region-based convolution neural network for explainable human-computer interaction understanding. The input gesture image is extracted through two mutually independent channels. The two channels have convolution kernel with different scales, which can extract the features of different scales in the input image, and then carry out feature fusion at the fully connection layer. Finally, it is classified by the softmax classifier. The two-channel convolutional neural network model is proposed to solve the problem of insufficient feature extraction by the convolution kernel. Experimental results of gesture recognition on public data sets NTU and VIVA show that the proposed algorithm can effectively avoid the over-fitting problem of training models, and has higher recognition accuracy and stronger robustness than traditional algorithms.

Keywords: explainable human-computer interaction understanding, two channel region-based convolution neural network, gesture recognition, softmax classifier, feature fusion.

1. Introduction

The application potential of hot technology in the field of explainable human-computer interaction (EHCI) has begun to show such as geo-spatial tracking technology on smartphones [1-3]. And the motion recognition technology is for wearable computers, stealth technology, immersive games, etc. Tactile interaction technology is for virtual reality, remote robotics, and telemedicine. Speech recognition technology is for call routing, home automation and voice dialing. Silent speech recognition is used for people with speech impairments and eye-tracking technology is used for advertising, websites, product catalogs, and magazine utility tests. The human-machine interface technology based on brainwave is used in the ”mind wheelchair” developed for people with speech and mobility disorders [4,5].

Gestures are an integral part of interpersonal communication. Gesture recognition opens up new ways for humans to interact with machines, devices or computers. With the development of science and technology, gesture recognition technology has developed from the era of data gloves with the help of external auxiliary equipment to the stage of pattern classification based on computer vision.

The gesture is a natural and intuitive means of HCI. It has become a trend to use gestures as computer input. In recent years, gesture recognition has gradually become an important research direction in the field of computer vision, especially in the application prospect of human-robot Interaction (HRI) technology, which greatly promotes the research development of gesture recognition. Gesture recognition refers to the use of certain algorithms to make the computer recognize the gesture of the human body in the picture or lens, and then understand the meaning of the gesture, to achieve mutual communication between the user and the computer. In the process of human-computer interaction to make the computer accurately understand people's intentions, the gesture recognition algorithm must have a highly accurate recognition effect, excellent processing speed, and recognition ability under different light, Angle, background, and other complex environments [6-8].

Currently, the popular visual gesture recognition can be divided into three stages: segmentation, feature extraction and recognition. Gesture segmentation is the basis of gesture recognition. Due to the clustering characteristics of skin color in color space, the majority of gesture segmentation methods at present use the color features of skin color (YUV, HSV [9], YCbCrl [10], etc.) or geometric features (such as elliptic model and graph model [11]). Traditional gesture recognition algorithms mainly include two categories: (1) gesture recognition based on hidden Markov model (HMM) [12], which can be used to express a Markov process with hidden unknown parameters, and gesture recognition process can be regarded as a Markov chain with time series. Therefore, this model is widely used in gesture recognition. (2) Gesture recognition based on set features [13]. This method uses gesture edge, contour, regional distribution and other features to recognize gestures. Both of these two gesture recognition methods require manual feature extraction, which is highly complex and requires high professional knowledge and experience of personnel. At the same time, it also has the problem of poor adaptability to unfamiliar scenes.

Feature extraction based on hand range is the key stage of gesture recognition. At present, influential studies are as follows. Lin et al. [14] extracted low-frequency coefficient features of fuzzy palmprint through Laplace smoothing transform and fused them with geometric features of hand to represent gestures. However, such feature extraction steps were complex and time-consuming. Asaari et al. [15] expressed gestures by integrating geometric features of hands with features such as palmlines, knuckle lines and veins of hands. Due to the complex background of the acquired images, the accuracy of gesture recognition based on texture features was low. Liu et al. [16] proposed a gesture recognition method that combined finger contour features with geometric features. Although it improved the robustness of gesture recognition, it required fingers to be separated from each other, which had certain limitations in practical application. Zhu et al. [17] proposed a two-level detection model, which could detect gestures, and the provided gesture border information could also be used for further gesture recognition analysis. However, the two stages of this method were trained separately, and the second stage required a large amount of label data.

Convolutional Neural Network (CNN) is one of the most widely used models in the field of machine vision and image processing, and has attracted great attention in industry and academia [18,19]. Convolutional neural networks can learn the local and global features of the input image through training, which solves the problem of insufficient feature extraction caused by manual feature extraction. With strong feature extraction and classification capabilities, CNN has been widely applied in many fields of pattern recognition, and has made remarkable achievements in image classification, face recognition, voice recognition and other fields.

In the field of image processing, the application direction of convolutional neural network is mainly image classification, target recognition, image segmentation and so on. In the field of gesture recognition, some scholars have tried. Nguyen et al. [20] combined maximum pooling with convolutional neural network for gesture recognition and obtained a recognition rate of 96.77%. Peng et al. [21] discussed the integration of gesture image pre-processing and gesture recognition process before the input of convolutional network, thus realizing end-to-end gesture recognition and improving the accuracy of recognition. Fang et al. [22] changed the first convolution layer of convolutional neural network into 3d convolution, so that dynamic gesture could be input into the model in the form of stereogram, which successfully solved the problem of regularization of input of dynamic gesture recognition. Singh et al. [23] creatively used the stereoconvolution kernel for gesture recognition and obtained a good gesture recognition accuracy.

Most of the existing works use pattern classification algorithms to realize gesture recognition. Rahman et al. [24] proposed an artificial neural network method for Bengali sign language static gesture letter recognition, but this method had high requirements on the number of samples and extracted data characteristic values. It also had low mean average precision (mAP) recognition accuracy. Panwar et al. [25] proposed a bit-coding sequence based on shape parameter features to achieve gesture classification, but this method had limitations for gesture direction placement and low robustness. Dominio et al. [26] proposed a gesture recognition method based on depth information with the help of Kinect equipment, but this method had high requirements on equipment, complex algorithm and low accuracy of experimental results. Yang et al. [27] proposed a gesture recognition method based on gesture main direction and Hausdorff-like distance template matching. The 2D Cartesian coordinate system was constructed to extract the gesture feature vector through the main direction of the gesture. However, this method had a high limitation on the main direction of the gesture. When the main direction of the gesture obtained was inconsistent with the main direction of similar gestures in the training library, it was prone to error recognition. In view of the complexity of current gesture feature extraction methods and the low gesture recognition rate in complex background, the research and practice of references [28,29] found that convolutional neural network (CNN) had scale invariance to flip, pan and scale. It was better than other machine vision methods in gesture detection and recognition applications. In references [30,31], R-CNN was used for target detection framework, which reached the world's leading level in face detection and pedestrian detection. Le et al. [32] proposed a multi-scale R-CNN method, which used the RPN network to detect a single object first, then extract geometric information. Finally, it determined whether a driver used a mobile phone or how many hands he had on the steering wheel. This method achieved the highest accuracy in the famous VIVA data

set. But when detecting the number of hands on the steering wheel, the detection rate was only 65%, which still had a large missed detection rate.

It can be seen from the above analysis that the current research work is to separate gesture detection and recognition. However, since CNN method can reach the world's leading level in gesture detection and recognition, can CNN be used to realize both detection and recognition? In this paper, a new gesture recognition method is proposed by extending R-CNN algorithm, which can detect and recognize gestures simultaneously. This paper uses the strategy of modifying key parameters to improve the existing R-CNN framework. In order to achieve higher recognition accuracy, a two-channel network is proposed to avoid the over-fitting problem of the training model. For NTU-Microsoft-Kinect-Hand Posture Dataset (NTU) and the Vision for Intelligent vehicles and Experiments on Applications (VIVA) gesture datasets, experiment results show that the proposed algorithm has higher accuracy and stronger robustness than traditional algorithms.

2. Convolutional neural network (CNN)

Convolutional neural network has a unique advantage in the field of image processing, its understanding of image is from local to global. Generally speaking, the pixels in the part of the image are closely related, while the parts that are far apart are weakly related. Convolutional neural network firstly perceives local features, and then integrates these local features at a high level to obtain the global features and topological structure of the image, and then judge the attributes and categories of the image [33,34]. Therefore, convolutional neural networks are highly invariant to shape translation, scale scaling, tilt or other forms of deformation.

Convolutional neural networks have two typical characteristics: one is that the two layers of neurons are locally connected rather than fully connected through the convolution kernel. Therefore, the convolution layer connected with the input image is a local link established for pixel blocks, rather than the traditional full link based on pixel points. Second, in the same layer, the weight parameters of the convolution kernel in each convolution layer are shared. These two features greatly reduce the number of parameters of the deep network, reduce the complexity of the model, and speed up the training speed of the model, so that the convolutional neural network has a great advantage in the image processing with pixel values as the processing unit. The main components of convolutional neural network include convolutional layer, pooling layer, activation function, full connection layer and classifier. Generally, the first layer directly connected to the input image is the convolution layer. This layer is responsible for connecting images directly. The input is transformed into a form that can be understood by the convolutional network through the processing of pixel values, and then propagated forward. The pooling layer and activation function are usually connected behind the convolutional layer, alternating with the convolutional layer.

2.1. Convolutional layer

The convolutional layer is the core component of convolutional neural network. Its main function is to extract local features of input through the fixed step movement of the convolutional kernel. The output of the convolution layer is the feature graph, and each element

in the feature graph is the output of a neuron. The input of this neuron connection is a local region of the previous layer's output feature map, also known as the local receptive field. The input in the sensory field is calculated by a set of synaptic weights and the output of the neuron is obtained by the activation function. By sharing this set of synaptic weights, also known as the convolution kernel, the number of parameters can be greatly reduced during feature graph generation. For a network, the size of the convolution kernel is fixed. However, the weight parameters of the convolution kernel, namely the convolution template, are obtained through the training of training samples.

The convolution kernel is the core of the convolution layer. Convolution kernel is a mapping relation of image features extracted by local receptive fields. The convolution kernel can also be viewed as an eigenmatrix. During the convolution operation, the convolution kernel moves in turn on the input, and the product accumulation operation is carried out between the convolution kernel and the elements at the corresponding positions on the receptive field to obtain the convolution value of the receptive field. After the moving, the eigenmatrix of the input is obtained, which is also called the eigengraph. A single convolution kernel can only extract a certain type of image features. Therefore, multi-convolutional kernels are generally used in practical convolutional neural networks. The mathematical expression of convolution operation is as follows:

$$x_j^n = f(\sum_{i \in M_i} x_i^{n-1} \cdot k_{ij} + b_j^n). \tag{1}$$

Where x_j^n is the j-th feature graph of the n-th convolution layer. x_i^{n-1} is the i-th output feature graph of layer n-1. $f(\cdot)$ represents the activation function. M_i represents the set of input graphs. k_{ij} is the convolution kernel between the i-th feature graph of the previous layer and the j-th feature graph of the current layer. b_j^n is the bias of the n-th layer, i.e. the current layer. Convolutional neural networks generally contain multiple convolutional layers to globalize extracted features.

2.2. Pooling layer

The pooling layer performs function transformation on the non-overlapping rectangular areas in the output feature map of the previous layer to obtain higher level invariant features [13]. Its function is to aggregate the feature graph obtained from the convolution layer, reduce the dimension of the feature graph, and reduce the sensitivity of the output to tilt, displacement and other forms of deformation, thus enhancing the generalization ability of the model. In the process of forward calculation of input image and feature image, the local features of the image are gradually expanded and integrated into global features through pooling operation. The commonly used pooling methods include mean pooling and max pooling. Pooling layer can keep original feature information while reducing feature dimension.

$$x_j^n = f(\beta_j^{n-1} down(x_j^{n-1}) + b_j^n). \tag{2}$$

Where x_j^n is the j-th feature map of the n-th pooling layer. $F()$ represents the activation function. $Down()$ indicates the pooling process. β_j^n is multiplicative bias. b_j is additive bias.

2.3. Fully connection layer

Because the convolution layer usually uses multiple convolution kernel templates. Therefore, the output is also juxtaposed feature maps of the same size. In order to fuse these feature maps together for classification, one or more fully connected layers are required. The full connection layer is generally connected between the pooling layer and the classifier, which is used to fuse different features expressed by multiple feature graphs. The powerful feature extraction capability of convolutional neural network comes from the convolution operation of multi-convolutional kernel template. The output of each convolution kernel template represents a feature expression from different Angles. Therefore, the integration of these features becomes an important process. Each neuron of the full connection layer is connected with all neurons of the output feature graph of the upper layer, which can be expressed as:

$$h(x) = f(W^T x + b). \tag{3}$$

Where x is the input of the feature graph of the previous layer. $h(x)$ is the output of the full connection layer. W is the connection weight and b is the bias. $F()$ is the activation function. The full connection layer combines all the features of the previous layer's feature map and then inputs them into the softmax classifier. The common activation functions include Sigmoid, Tanh and Relu (Rectified Linear Unit).

3. Proposed Two Channel R-CNN

R-CNN algorithm consists of two networks: (1) a region proposal network (RPN) candidate box extraction network, which can extract the candidate regions of interest (RoIs) that may contain the target; (2) R-CNN network is used to classify RoIs (target or background) and refine the bounding box (BBox) of target area. Using RPN network to generate BBox is the main innovation of R-CNN compared with other detection algorithms. During the training, the method of alternating training RPN and R-CNN is adopted.

Step1. Train RPN.

Step2. Train R-CNN with candidate region extracted by RPN.

Step3. R-CNN is used to initialize the convolution layer common to RPN network.

Step4. Select the generation to perform Step1-Step 3 until the end of the training.

This is the training method used in reference [35]. In the first generation selection, the model obtained by ImageNet is used to initialize the parameters of the convolution layer in RPN and R-CNN. Starting from the second generation selection, when training RPN, the shared convolutional layer parameters of R-CNN are used to initialize the shared convolutional layer parameters in RPN, and then only the convolutional layer not shared by fine-tuning and corresponding parameters of other layers. When R-CNN is trained, its convolution layer parameters shared with RPN are kept unchanged. Only the parameters corresponding to the layer not shared by fine-tuning can be realized, so that the feature sharing training of two network convolution layers can be realized.

3.1. RPN

R-CNN uses the RPN network to extract gesture candidate regions, which is essentially a sliding window. RPN obtains a series of target candidate regions with target scores from

images of any size. The specific process is as follows. A small network is used for sliding scanning on the feature graph obtained in the last convolutional layer. This network is fully connected with the $N \times N$ window on the feature graph every sliding. Then it maps to a lower-dimensional vector, such as 512 dimensions. Finally, the low-dimensional vector is sent into two fully connected layers, namely, box-regression Layer (REG) and box-classification Layer (CLS). For each position, CLS outputs the probability belonging to the foreground and background from the 512-dimensional features. REG outputs four panning scaling parameters from the 512-dimensional feature.

RPN anchors consider k possible reference windows for each slide location, which means that each slide location will predict up to nine candidate areas at once. For a $W \times H$ feature map, $W \times H \times k$ candidate regions are generated. RPN's anchor has translation invariance. Its principle is to sample the slide-to-height ratio of multi-scale anchor points located in the area of $N \times N$ with the window as the center. The base area size is 16×16, and the width to height ratio is 2:1, 1:1 and 1:2 respectively. The window scale of the center point [8,16,32] is sampled so that nine anchors are created in each sliding window location.

The choices for anchors are as follows. We have two kinds of anchors that are categorized as detection accuracy, that is, the intersection-over-union (IoU) between the object box generated by the model and any one of the tagged boxes. Positive sample calibration rules are as follows.

Rule 1. If the IoU value of anchor box and Ground Truth corresponding to anchor has the maximum value, they will be marked as positive samples.

Rule 2. If the IoU of the candidate frame and marker frame corresponding to anchor is greater than 0.7, it is marked as a positive sample. In fact, rule 2 can basically find enough positive samples, but for some extreme cases, for example, the IoU of candidate boxes and marker boxes corresponding to all anchors is not greater than 0.7, so rule 1 can be used to generate them. Negative sample calibration rules are as follows:

Rule 3. If the IoU of candidate box corresponding to anchor and marker box is less than 0.3, it is marked as negative sample. The rests are neither positive nor negative samples for final training. Candidate boxes that cross image boundaries are also discarded. IoU calculation formula is:

$$IoU = \frac{S_{anchorBox \cap S_{groundTruth}}}{S_{anchorBox \cup S_{groundTruth}}}. \tag{4}$$

For each anchor, a binary classifier is first attached behind, and two score outputs are used to represent the probability that it is an object and the probability that it is not an object. It then attaches a REG output representing the four coordinate positions of this anchor. The loss function of RPN is defined as:

$$L(p_i, t_i) = s\frac{1}{N_{cls}}\sum_i L_{cls}(p_i + p_i^*) + \lambda\frac{1}{N_{reg}}p_i^* L_{reg}(t_i + t_i^*). \tag{5}$$

Where i represents the index of each RoI. p_i^* is the label representing the category (positive sample=1, negative sample=0). $t_i = t_x, t_y, t_w, t_h$ indicates the offset of the suggestion box relative to the candidate box. t_i^* represents the offset of the marker box with respect to the candidate box. The goal of learning is to make the former close to the value of the latter, and the calculation formula is:

$$t_x = \frac{x - x_\alpha}{w_\alpha}. \tag{6}$$

$$t_w = log(w/w_\alpha). \tag{7}$$

$$t_h = log(h/h_\alpha). \tag{8}$$

$$t_h = log(h/h_\alpha). \tag{9}$$

Where x, y, w and h represent the central coordinates of the proposed area and its width and height respectively. The suggestion box is generated by the candidate box fine-tuning through the regression process until it approaches the marker box.

The classification loss L_{cls} represents the logarithmic loss of the candidate box predicted as the target and background respectively. Regression loss $L_{reg}(t, t'_i) = R(t_i - t^*_i)$. Where, the loss function is:

$$R(x) = 0.5x^2, |x| < 1. \tag{10}$$

N_{cls} represents the number of mini-batches extracted from an image. A mini-batch is made up of 256 candidate regions randomly selected from an image. Among them, the ratio of positive and negative samples is 1:1. If the positive samples are less than 128, more negative samples should be used to meet the requirement that there are 256 candidate regions for training. N_{reg} stands for the total number of anchors. λ is the balance factor of loss in the CLS layer and REG layer, usually $\lambda = 1$. In the detection process, the setting rule is to combine the prediction boxes whose probability is greater than a certain threshold and IoU is greater than a certain threshold by the non-maximum suppression method.

3.2. Two channel R-CNN

The network structure of gesture recognition used in this paper is shown in Figure 1. Images of any size are input to CNN and propagated forward to the last shared convolution layer through CNN. On the one hand, RPN is used to output candidate regions; on the other hand, R-CNN is used to detect and identify targets in candidate regions extracted by RPN.

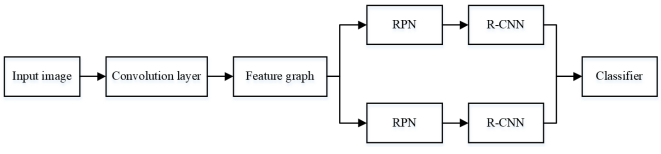

Fig. 1. Proposed network structure diagram of gesture recognition

1. Input parameters of CNN. The gesture data size of the dataset used is 640×480. To improve the recognition rate, we set the input size as 640×480 for both the training and test phases. The purpose of this paper is to identify 10 gestures in the NTU dataset and detect hands in the VIVA dataset. The number of categories in the NTU and VIVA datasets is set to 11 and 2 (including background), respectively.
2. Anchors parameters. In this paper, the size of gesture area in VIVA and NTU data sets is analyzed. In order to improve the convergence speed during training and the recognition rate in test stage, the benchmark area is set as 8×8. The ratio of width to height is 2:1, 1:1 and 1:2 respectively. The scale used in this paper is [8,12,16]. The nine anchors examples generated at each slide window location are shown in figure 2.

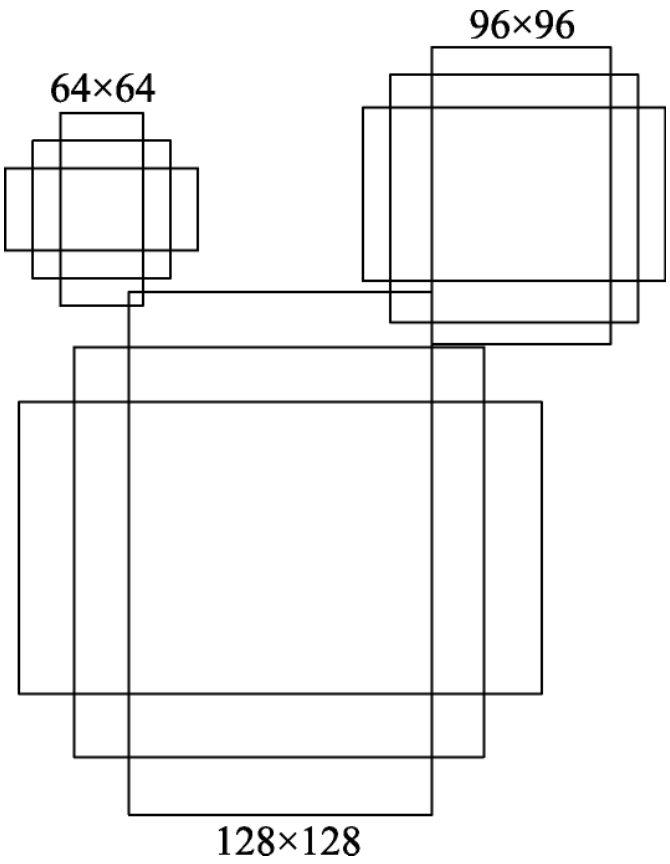

Fig. 2. Anchors scale example diagram

3. Hyperparameters of training. During training, the publicly trained ImageNet classification model is used to initialize the network layer shared by RPN and R-CNN. The remaining layers are randomly initialized with a Gaussian distribution with a

mean value of 0 and a standard deviation of 0.01. After some experimental analysis and comparisons, the learning rate of the first 60K selection is set at 0.001, and the learning rate of the subsequent 40K selection is set at 0.0001. After 104 times of selection training, good results can be achieved. Momentum and weight decay factors use empirical values of 0.9 and 0.0005.

3.3. Algorithm implementation

Due to the large amount of data required for training with CNN, it is easy to overfit when there are few images in the data set. It is a major challenge to obtain a robust CNN model to avoid the problem of over-fitting during training. At present, it is mainly through strengthening regularization of loss layer to avoid over-fitting, which means that the degree of model fitting to training data is much higher than test data. In this paper, a variety of over-fitting methods (data enhancement, weight attenuation, k-fold cross validation, etc.) are tried and found that the improvement effect is not obvious. Reference [36] used noise data to train the performance of CNN, in which some free labels might be correct or wrong for each training image. However, this method was not applicable to training R-CNN network. The Newlabling algorithm proposed in this paper adds disturbance labels to training data to reduce the degree of fitting. During the training, 10 images are randomly selected for each generation. Since positive and negative samples are not specified in the production of data sets, but are determined in training according to the IoU value and the label of the target real box, this paper randomly selects part of the IoU in every 1000 generations according to probability and set it as 0.5. The rest of the IoU is set as 0.7. In essence, when the IoU is set very low, the original positive label may become a negative label, and the negative label may also become a positive label, thus generating disturbance labels. Noise is added to the loss layer by the disturbance tag, and this noise gradient is propagated in the RPN back propagation stage.

The steps of $Newlabling$ algorithm are as follows:

In the RPN training stage, the label data sent into RPN is $D = (p_n^*, t_n^*)_{n=1}^L$. Among them, the $p_n^* \in R^{C+1}$, C is category number, $p_n^* = [0, 1, 2, \cdots, C]$, 0 is background. $1, 2, \cdots, C$ represents C target category markers to be identified. Data labels are 4-dimensional vectors $t_n^* = [x^*, y^*, w^*, h^*]$, they represent the center coordinates of the target on the original image and the width and height of the BBox respectively. L represents the number of images used in each RPN training network. In this paper, $L = 10$. Its aim is to train a model $M : f(p, t, \theta) \in R^2$, θ is a model parameter, which is usually initialized with white noise θ_0, and then the stochastic Gradient Descent (SGD) algorithm is used for updating. At the m-th iteration,

$$\theta_{m+1} = \theta_m + \gamma_m \cdot \frac{1}{|D_m|} \sum (p, t) \in D_m \cdot \nabla_{\theta_t}[L(p_i, t_i)]. \tag{11}$$

Wherein, $L(p_i, t_i)$ is calculated by formula (2). $\nabla_{\theta_t}[L(p_i, t_i)]$ is used to back-propagate the gradient. γ_m stands for learning rate. D_m randomly selects images from the total data set. In the training (test) stage, RPN firstly outputs the categories, positions and probability scores of 6K candidate regions, and finally selects the top 300 candidate regions with probability scores from these 6K candidate regions, and then feeds the information to R-CNN network.

$Newlabling$ generates disturbance which primarily affects the labeling of 12K candidate area categories. For each candidate region, the disturbance labeling is expressed as $p = p[p_0, p_1]$. Where, p is generated by the input data according to the positive and negative sample calibration rules (p_0 represents the probability that the candidate box is the background, p_1 represents the probability that the candidate box is the target). The size of the IoU is decisive.

$$IoU = \begin{cases} 0.7 & \tilde{I}_j = 1 \\ 0.5 & \tilde{I}_j = 0 \end{cases} \tag{12}$$

$$\tilde{I} = [\tilde{I}_1, \cdots, \tilde{I}_N]. \tag{13}$$

N is the number of selected generations each time. In this paper, $N = 1000$. $\tilde{I}$ follows a Bernoulli distribution.

$$j \sim \phi_j(\alpha), \tilde{I}_j = 1, \tilde{I}_i = 0. \tag{14}$$

Where, the $\phi_j = \frac{1}{N} \cdot \alpha$. α is the noise rate.

The role of the noise rate in the $Newlabling$ algorithm is as follows. The noise rate α determines the number of possible error labels in each training set. When $\alpha = 0$, there is no noise, and the detection algorithm will be proceeded according to the normal situation. When $\alpha = 100\%$, the accuracy rate of all labels is 50%, that is, random labels at this time are completely unreliable. This $Newlabling$ algorithm is applied to NTU and VIVA data sets in this paper, and loss rate results and mAP of VGG_M model are shown in figure 3, figure 4, table 1, and table 2. It can be seen that when $\alpha = 10\%$, higher accuracy can be achieved, and the convergence speed is also improved to a certain extent. The Newlabling algorithm can improve the generalization ability of CNN model. When α is higher (up to 15%), the convergence speed and accuracy rate of the network will be reduced. The reason is that the label of training data is not reliable at this time. By comparing figure 3 and figure 4, it can be seen that $Newlabling$ algorithm has a more obvious effect on NTU dataset than VIVA dataset. The reason is that the hand of VIVA data set is in a complex background, and its background light intensity changes greatly and the occlusion is serious. However, the gestures in the NTU dataset are in a simple background, and the light intensity basically does not change. There is no gesture occlusion. Therefore, the loss rate jitter of VIVA data set is larger and the convergence rate is slower in each iteration.

Table 1. mAP with different α on NTU data sets/%

α	mAP
0	98.2
5	98.4
10	99.0
15	97.9

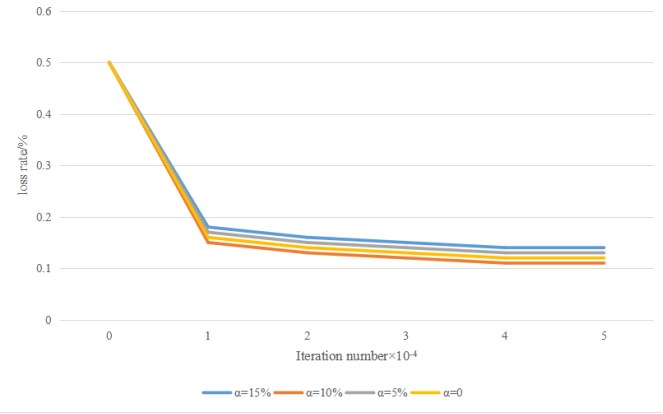

Fig. 3. Loss rate of gesture recognition using different α on NTU datasets

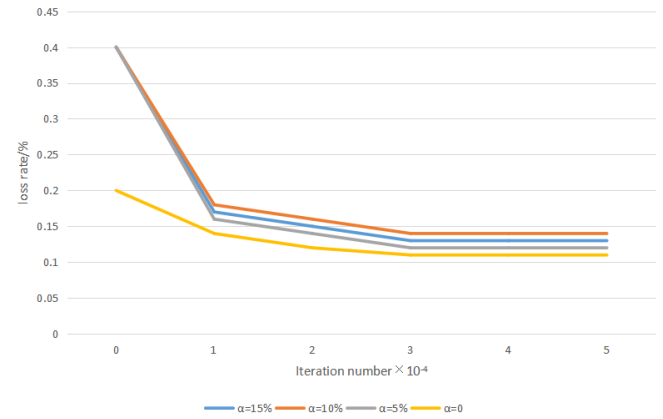

Fig. 4. Loss rate of gesture recognition using different α on VIVA datasets

Table 2. mAP with different α on VIVA data sets/%

α	mAP
0	83.8
5	84.6
10	84.7
15	83.2

3.4. Experiments and analysis

Experimental environment is Ubuntu14.04 with the py-RCNN (https://github.com/rbgirshick/py-rcnn). NTU and VIVA gesture data sets are adopted. In this paper, only color images are used, and the depth images in the dataset are not used. The original gesture images in the database are 248-256 or 128-128 pixels. The image contains a large amount of data, and gestures occupy a small area in the whole image with a lot of background redundancy. If the original image is directly used as the input of the convolutional network, the amount of data to be processed will be very large and the classification results will be easily affected by the complex background. Therefore, the image is preprocessed and then used as the input of the model. The image information content after preprocessing is reduced to 1/64 of the original image.

In order to verify the practical application effect of two-channel convolutional neural network in gesture recognition, two groups of experiments are designed here. The first group compares the recognition effects of single channel and two-channel convolutional neural networks. At the same time, the recognition accuracy of two-channel convolutional neural networks with different convolution kernel sizes is compared. The second experiment compares the recognition accuracy of the proposed algorithm with that of previous gesture recognition algorithms to verify the improvement of feature extraction ability and recognition accuracy of two-channel convolutional neural network.

According to the principle of the proposed algorithm in this paper, the improvement of feature extraction ability of two-channel convolutional neural network mainly comes from convolution kernels with different scales. However, for different processing objects and different application scenarios, the selection of the convolution kernel size is not fixed. Only with the most appropriate convolution kernel, the best classification effect can be achieved. Therefore, the networks selected for this experiment include single-channel convolutional neural network with convolution kernel size of 5-5 and two-channel convolutional neural network with convolution kernel size of 3×3, 5×5, which are matched with four sizes. Experiments are conducted on two static gesture databases respectively.

Table 3 shows the recognition results of different convolution kernel sizes in the data sets. After using two-channel convolutional neural network, gesture recognition accuracy is significantly improved compared with single-channel convolutional neural network. At the same time, the convolution kernels of different sizes match each other, and the recognition effect is also different. According to the image sizes of the data sets and the network input after pre-processing in this experiment, the best recognition effect can be obtained by the collocation of convolution kernels with the size of 5×5 and 7×7. The experimental results show that the proposed model uses two independent convolutional channels to extract features from input images, and can obtain richer feature information than single-channel networks. The feature information is reflected in local features with different sizes. Combining these features to classify images can effectively improve the accuracy of static gesture recognition of convolutional neural network.

Figure 5 compares the classification errors of single-channel and double-channel convolutional neural networks with the increase of iteration number during training process. The experimental results show that the classification accuracy of the model tends to be stable after more than 20 iterations. The error rate of two-channel convolutional neural network is obviously better than that of single-channel convolutional neural network and has a more stable convergence process.

Table 3. Gesture recognition accuracy of two-channel model and single-channel model with different sizes of convolution kernel/%

Model	NTU	VIVA
Single channel RCNN(5×5)	96.99	96.05
Two-channel RCNN(3×3, 5×5)	97.76	96.76
Two-channel RCNN(5×5, 5×5)	97.75	97.23
Two-channel RCNN(7×7, 5×5)	98.21	97.67
Two-channel RCNN(3×3, 7×7)	96.71	96.48

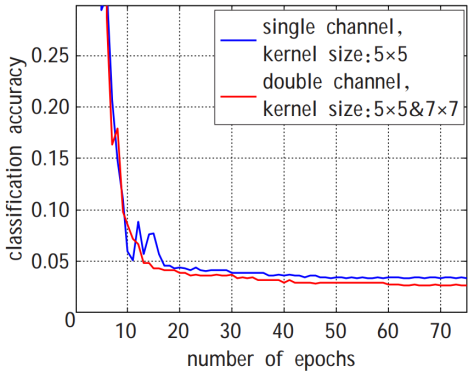

Fig. 5. The curve of recognition rate changes with the number of iterations

In this experiment, the representative traditional gesture recognition algorithm and the recognition method based on convolutional neural network are selected and verified on the static gesture database successively. Based on the above experimental results, the two-channel convolutional neural network model chooses the convolution kernel of 7×7, 5×5.

The results of comparative experiments are listed in Table 4. Big and Deep MPCNNP methods combine maximum pooling with convolutional neural network to form deep convolutional neural network, achieving a recognition rate of 96.88%. Bottom-up structured DCNN is an end-to-end deep convolutional neural network with a gesture recognition accuracy of 88.89%. The recognition accuracy of the proposed algorithm reaches 98.21%, which is higher than the traditional convolutional neural network model.

Table 4. Recognition rate comparison with different gesture recognition algorithms

Method	Recognition rate/%
Spatial Pyramid [37]	85.43
bottom-up structured DCNN [38]	88.89
Tiled CNN [39]	90.59
Big and Deep MPCNN [40]	96.88
Proposed Method	98.21

Based on the above results, the following conclusions can be drawn:

1. Two-channel convolutional neural network uses two convolutional channels with convolution kernels of different sizes to process the input image, so that the network can learn more features. The adequacy of feature extraction is higher than that of traditional single-channel convolutional neural network, so the accuracy of gesture recognition is better.
2. The proposed algorithm in this paper extends the traditional convolutional neural network, but also adopts the supervised learning method for network training. The process of feature extraction does not need human participation, which reflects the excellent scalability of convolutional neural network. At the same time, it also reflects the great potential of the structural expansion of convolutional neural network to improve performance.

4. Conclusion

In this paper, a two-channel convolutional neural network model is proposed, which uses different convolution sizes to check the original gesture images for feature extraction, so as to obtain richer local information and overall topology of gestures. After pooling, features are fused at the full connection layer to extract deeper classification information. The experimental results show that the two-channel convolutional neural network can classify 24 kinds of gestures and adapt to various background forms such as simple and complex, bright and dark, with strong generalization ability. At the same time, two-channel convolutional neural network has a lot of research and development space, mainly including the following three aspects: (1) try to introduce more hierarchical and scale features to further improve the adaptability of the model to complex background; (2) At present, the accuracy of dynamic gesture recognition still has a lot of room for improvement, and the model can be applied to the field of dynamic gesture recognition; (3) The convolutional neural network model for gesture recognition needs a large number of labeled image data for training. In the future, network training can be carried out through unsupervised or semi-supervised learning to reduce the model's dependence on a large number of labeled data.

Availability of data and materials. The data used to support the findings of this study are available from the corresponding author upon request.

Competing interests. The authors declare that they have no conflicts of interest.

References

1. Nguyen K A. "Utilizing a Human-Computer Interaction Approach to Evaluate the Design of Current Pharmacogenomics Clinical Decision Support," *Journal of Personalized Medicine*, vol. 11, 2021.
2. Zhong Q, Yang Q. "Analyzing the Mental States of the Sports Student Based on Augmentative Communication with Human-Computer Interaction," *Journal of Interconnection Networks*, 2021.

3. Jing Yu, Hang Li, Shoulin Yin. "Dynamic Gesture Recognition Based on Deep Learning in Human-to-Computer Interfaces," *Journal of Applied Science and Engineering*, vol. 23, no. 1, pp. 31-38, 2020.
4. Chaaba Ne S, Etien Ne A M, Schyns M, et al. "The Impact of Virtual Reality Exposure on Stress Level and Sense of Competence in Ambulance Workers," *Journal of Traumatic Stress*, 2021.
5. X. Zhang, F. Zhang and C. Xu. "Joint Expression Synthesis and Representation Learning for Facial Expression Recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 3, pp. 1681-1695, 2022.
6. Feng T. "Mask RCNN-based Single Shot Multibox Detector For Gesture Recognition In Physical Education," *Journal of Applied Science and Engineering*, vol. 26, no. 3, pp. 377-385, 2022.
7. Gasteiger N, Hellou M, Ahn H S. "Factors for Personalization and Localization to Optimize Human-Robot Interaction: A Literature Review," *International Journal of Social Robotics*, 2021:1-13.
8. Eskofier B M. "A Smart Capacitive Sensor Skin with Embedded Data Quality Indication for Enhanced Safety in Human-Robot Interaction," *Sensors*, vol. 21, 2021.
9. Hamuda E, Ginley B M, Glavin M, et al. "Automatic crop detection under field conditions using the HSV colour space and morphological operations," *Computers & Electronics in Agriculture*, vol. 133(Complete), pp. 97-107, 2017.
10. Udoh N, Ekpenyong M. "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, 2022.
11. Bhattacharjee H, Anesiadis N, Vlachos D G. "Regularized machine learning on molecular graph model explains systematic error in DFT enthalpies," *Scientific Reports*, vol. 11, no. 1, 2021.
12. J. Wan, Q. Ruan, G. An and W. Li, "Gesture recognition based on Hidden Markov Model from sparse representative observations," *2012 IEEE 11th International Conference on Signal Processing*, 2012, pp. 1180-1183, doi: 10.1109/ICoSP.2012.6491787.
13. Q. Chen, N. D. Georganas and E. M. Petriu, "Real-time Vision-based Hand Gesture Recognition Using Haar-like Features," *2007 IEEE Instrumentation & Measurement Technology Conference IMTC 2007*, 2007, pp. 1-6, doi: 10.1109/IMTC.2007.379068.
14. Lin S, Yuan W, Jing L, et al. Blurred palm-print recognition based on fusion of Laplacian smoothing transform and geometric features of hand," *Chinese Journal of Scientific Instrument*, vol. 34, no. 2, pp. 415-422, 2013.
15. Asaari M S M, Suandi S A, Rosdi B A. Fusion of Band Limited Phase Only Correlation and Width Centroid Contour Distance for finger based biometrics," *Expert Systems with Applications*, vol. 41, no. 7, pp. 3367-3382, 2014.
16. Liu F, Liu H Y, Gao L, et al. Hand shape recognition based on fusion features of fingers and particle swarm optimization," *Optics & Precision Engineering*, vol. 23, no. 6, pp. 1774-1782, 2016.
17. X. Zhu, W. Liu, X. Jia and K. -Y. K. Wong, "A two-stage detector for hand detection in egocentric videos," *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1-8, 2016, doi: 10.1109/WACV.2016.7477665.
18. Qingwu Shi, Shoulin Yin, Kun Wang, Lin Teng and Hang Li. Multichannel convolutional neural network-based fuzzy active contour model for medical image segmentation," *Evolving Systems*, 2021. https://doi.org/10.1007/s12530-021-09392-3
19. Wu S. "Simulation of classroom student behavior recognition based on PSO-kNN algorithm and emotional image processing," *Journal of Intelligent & Fuzzy Systems*, vol. 40, no. 4, pp. 7273-7283, 2021.
20. Nguyen-Trong K, Vu H N, Trung N N, et al. "Gesture Recognition Using Wearable Sensors With Bi-Long Short-Term Memory Convolutional Neural Networks," *IEEE Sensors Journal*, vol. 21, no. 13, pp. 15065-15079, , 2021.

21. Y Peng, Wang J, Pang K, et al. "A Physiology-Based Flexible Strap Sensor for Gesture Recognition by Sensing Tendon Deformation," *IEEE Sensors Journal*, vol. 21, no. 7, pp. 9449-9456, 2021.
22. Fang Y, Zhang X, Zhou D, et al. "Improve Inter-day Hand Gesture Recognition Via Convolutional Neural Network-based Feature Fusion," *International Journal of Humanoid Robotics*, 2021.
23. Singh D K. "3D-CNN based Dynamic Gesture Recognition for Indian Sign Language Modeling," *Procedia Computer Science*, vol. 189, pp. 76-83, 2021.
24. Rahman M A. "Recognition of Static Hand Gestures of Alphabet in Bangla Sign Language," *IOSR Journal of Computer Engineering*, vol. 8, no. 1, pp. 07-13, 2012.
25. M. Panwar, "Hand gesture recognition based on shape parameters," *2012 International Conference on Computing, Communication and Applications*, pp. 1-6, 2012, doi: 10.1109/ICCCA.2012.6179213.
26. Dominio F, Donadeo M, Zanuttigh P. "Combining multiple depth-based descriptors for hand gesture recognition," *Pattern Recognition Letters*, vol. 50, pp. 101-111, 2014.
27. Yang X, Feng Z, Huang Z, et al. "Gesture Recognition Based on Combining Main Direction of Gesture and Hausdorff-like Distance," *Journal of Computer-Aided Design & Computer Graphics*, 2016.
28. X. Zhang, F. Zhang and C. Xu, "Joint Expression Synthesis and Representation Learning for Facial Expression Recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 3, pp. 1681-1695, March 2022, doi: 10.1109/TCSVT.2021.3056098.
29. Y. Xia, W. Zheng, Y. Wang, H. Yu, J. Dong and F. -Y. Wang, "Local and Global Perception Generative Adversarial Network for Facial Expression Synthesis," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 3, pp. 1443-1452, March 2022, doi: 10.1109/TCSVT.2021.3074032.
30. H. Zhang, W. Su, J. Yu and Z. Wang, "Identity-Expression Dual Branch Network for Facial Expression Recognition," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 13, no. 4, pp. 898-911, Dec. 2021, doi: 10.1109/TCDS.2020.3034807.
31. Chen C., Liu MY., Tuzel O., Xiao J. "R-CNN for Small Object Detection," *Computer Vision - ACCV 2016. ACCV 2016. Lecture Notes in Computer Science*, vol. 10115, 2017. Springer, Cham.
32. T. H. N. Le, Y. Zheng, C. Zhu, K. Luu and M. Savvides, "Multiple Scale Faster-RCNN Approach to Driver's Cell-Phone Usage and Hands on Steering Wheel Detection," *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 46-53, 2016, doi: 10.1109/CVPRW.2016.13.
33. D. Poux, B. Allaert, N. Ihaddadene, I. M. Bilasco, C. Djeraba and M. Bennamoun, "Dynamic Facial Expression Recognition Under Partial Occlusion With Optical Flow Reconstruction," *IEEE Transactions on Image Processing*, vol. 31, pp. 446-457, 2022, doi: 10.1109/TIP.2021.3129120.
34. F. Zhang, T. Zhang, Q. Mao and C. Xu, "Geometry Guided Pose-Invariant Facial Expression Recognition," *IEEE Transactions on Image Processing*, vol. 29, pp. 4445-4460, 2020, doi: 10.1109/TIP.2020.2972114.
35. Ahmad M, Ahmed I, Jeon G. "An IoT-enabled real-time overhead view person detection system based on Cascade-RCNN and transfer learning," *Journal of Real-Time Image Processing*, vol. 6, 2021.
36. Sukhbaatar S, Bruna J, Paluri M, et al. Training convolutional networks with noisy labels[OL]. [2017-06-01]. https://arxiv.org/abs/1406.2080
37. Wang J, Lv P, Wang H, et al. "SAR-U-Net: squeeze-and-excitation block and atrous spatial pyramid pooling based residual U-Net for automatic liver CT segmentation," *Computer Methods and Programs in Biomedicine*, vol. 208, 2021.
38. Jin Y, Bhatia A, Wanvarie D. Seed Word Selection for Weakly-Supervised Text Classification with Unsupervised Error Estimation. 2021. https://doi.org/10.48550/arXiv.2104.09765

39. M Trusca, Spanakis G. "Hybrid Tiled Convolutional Neural Networks (HTCNN) Text Sentiment Classification." 2020. https://doi.org/10.48550/arXiv.2001.11857
40. Zhang C., He D., Li Z., Wang Z. "Parallel Connecting Deep and Shallow CNNs for Simultaneous Detection of Big and Small Objects," *Pattern Recognition and Computer Vision. PRCV 2018. Lecture Notes in Computer Science*, vol. 11259, 2018. Springer, Cham.

Pingping Li is with College of Fine Arts, Zhengzhou Normal University. Her research interests include: art image processing and digital art research.

Lu Zhao is with School of Fine Arts, Yulin Normal University. Main research direction: sports art.

Received: March 22, 2022; Accepted: September 12, 2022.