doi:10.3772/j.issn.1006-6748.2022.01.008

An improved micro-expression recognition algorithm of 3D convolutional neural network①

WU Jin(吴 进)②, SHI Qianwen, XI Meng, WANG Lei, ZENG Huadie
(School of Electronic and Engineering, Xi'an University of Posts and Telecommunications, Xi'an 710121, P. R. China)

Abstract

The micro-expression lasts for a very short time and the intensity is very subtle. Aiming at the problem of its low recognition rate, this paper proposes a new micro-expression recognition algorithm based on a three-dimensional convolutional neural network (3D-CNN), which can extract two-dimensional features in spatial domain and one-dimensional features in time domain, simultaneously. The network structure design is based on the deep learning framework Keras, and the discarding method and batch normalization (BN) algorithm are effectively combined with three-dimensional visual geometry group block (3D-VGG-Block) to reduce the risk of overfitting while improving training speed. Aiming at the problem of the lack of samples in the data set, two methods of image flipping and small amplitude flipping are used for data amplification. Finally, the recognition rate on the data set is as high as 69.11%. Compared with the current international average micro-expression recognition rate of about 67%, the proposed algorithm has obvious advantages in recognition rate.

Key Words: micro-expression recognition, deep learning, three-dimensional convolutional neural network (3D-CNN), batch normalization (BN) algorithm, dropout

0 Introduction

Human facial expressions are extremely rich. Through the analysis of micro-expressions, people's inner emotional activities can be understood. Micro-expressions cannot be disguised or concealed. The intensity of expression is very subtle and the duration is very short. A series of special uses make micro-expressions recognition have very important application prospects and value in the field of daily life. Therefore, micro-expression recognition has become one of the important fields of research[1-2]. Due to the particularity of micro-expression, the difficulties and challenges faced by micro-expression recognition are indispensable. How to further improve the accuracy of micro-expression recognition has become one of the issues to be considered. Nowadays, the existing micro-expression feature extraction algorithms mainly use histogram of oriented gradient(HOG)[3], local binary pattern(LBP)[4], and local binary patterns from three orthogonal planes(LBP-TOP)[5], etc. The above methods have very good results in the extraction of salient feature points, but they all have the problem of relatively simple feature description. Due to the particularity of micro-expression recognition, the above methods cannot extract the original features of micro-expressions accurately and quickly. At the same time, micro-expressions are based on video frame sequences[6]. Among the traditional algorithms, the algorithm for extracting temporal features, such as optical flow method, aims at feature tracking on two consecutive frames of images. However, a complete micro-expression contains at least a dozen or more video sequences, and it cannot extract the information of a long video frame sequence, so the recognition rate of micro-expression is not ideal. Therefore, the expected results cannot be achieved when using these methods for micro-expression recognition.

In recent years, as a newly emerging recognition technology, micro-expression has not been studied much. Before 2015, people used traditional image recognition algorithms for feature extraction and recognition of micro-expressions. It was not until 2016 that deep learning was applied to the recognition algorithm of micro expressions. Deep learning was first proposed by in Ref.[7]. After years of development and re-

① Supported by the Shaanxi Province Key Research and Development Project (No. 2021GY-280), Shaanxi Province Natural Science Basic Research Program Project (No. 2021JM-459), the National Natural Science Foundation of China (No. 61834005, 61772417, 61802304, 61602377, 61634004) and the Shaanxi Province International Science and Technology Cooperation Project (No. 2018KW-006).

② To whom correspondence should be addressed. E-mail: wujin1026@126.com.

Received on Feb. 1, 2021

search, deep learning algorithms are widely used in target tracking[8], instance segmentation [9], target detection [10], micro-expression recognition[11], and research in the fields of face recognition. Using the convolutional neural network (CNN) proposed in Ref. [12], in order to effectively extract features, useful information is extracted through feature selection and verified on the CASME II micro-expression data set. Ref. [13] used deep multi-task CNN to realize the location of key points of the face and divide the facial area, in the micro-expression frame sequence, the optical flow information between frames can be better extracted. A micro-expression recognition algorithm combining temporal interpolation model (TIM) and CNN was proposed in Ref. [14]. In reference to the existing methods on the quality and quantity of training data, Ref. [15] proposed a face-enhanced generation confrontation network to reduce the influence distribution of unbalanced deformation attributes. Ref. [16] proposed a dynamic segmentation sparse imaging module. Segmented motion participation in spatio-temporal networks can capture the long-distance spatial relationships of facial micro-expression and enhance the robustness of feature-level subtle movement changes. From a CNN that can only handle two-dimensional spatial features, it is expanded to three-dimensional CNN (3D-CNN). The three-dimensional CNN was designed in Ref. [17], namely, 3D-CNN. 3D-CNN can extract the spatial characteristics of the image and the temporal characteristics of the image frame sequence. In the network layer structure, the network input layer is five-dimensional, including the length and width of the image frame, the number of channels per frame, the length of the image frame sequence, and the batch value of the input image frame sequence. Moreover, the convolutional layer of 3D-CNN does not require the size and length of the input image frame sequence, which effectively overcomes the inability of traditional algorithms such as optical flow to extract the time domain of longer frame sequences information problem. Compared with CNN, the overall network layer structure is consistent with CNN, except that the core sizes of the convolutional layer and pooling layer in 3D-CNN are expanded from CNN's two-dimensional to three-dimensional.

The 3D-CNN network can directly process information in the time domain, reducing redundancy. The input data dimension of the 3D-CNN network is five-dimensional, and it can process multiple image frame sequences during training and testing. This paper proposes a micro-expression recognition algorithm that improves the traditional 3D-CNN network. Compared with the traditional CNN network structure, the execution efficiency of this algorithm is higher.

1 Network structure and algorithm design

1.1 3D convolutional neural network

On the basis of CNN structure, the extension of CNN from two-dimensional space to three-dimensional space can be called three-dimensional convolutional neural network (3D-CNN). Micro-expression recognition is a research field based on video sequences. CNN cannot extract temporal information, while 3D-CNN can extract spatial domain features as well as temporal characteristics of image frame sequences, preserving temporal information and achieving better results. Fig. 1 is the main difference between CNN and 3D-CNN, where Fig. 1(a) and (b) are 2D convolution for single-channel images and multi-channel images (here multi-channel images can refer to the same picture 3 color channels, also refers to multiple pictures stacked together, that is, a short video). For a filter, the output is a two-dimensional feature map, which completely compresses the information of the multiple channels. The output of the 3D convolution in Fig. 1(c) is still a 3D feature map.

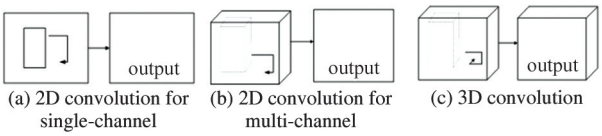

Fig. 1 The main difference between CNN and 3D-CNN

It can be seen from Fig. 1 that, in general, for the input video data, CNN recognizes the images in the video frame by frame. This operation ignores the relative motion between image frames in the video data in the time dimension information, however, for the spatial and temporal information in video data, 3D-CNN can extract them well.

In the common operation methods of the pooling layer, in addition to extracting input data, the three-dimensional pooling operation methods include average pooling and maximum pooling. Compared with the two-dimensional pooling operation, the pooling size of the three-dimensional pooling layer is $2 \times 2 \times 2$. During the pooling operation, the average or maximum value of 8 values in each $2 \times 2 \times 2$ cube is taken. As Eq. (1) is used to calculate the output of 3D convolution and 3D pooling.

$$N = \frac{W - F + 2P}{S} + 1 \tag{1}$$

where, N is the output, W is the length, width or

height of the input, the default kernel size is the same, F is the kernel size, P is the number of zeros added around the input unit, and S is the stride.

In summary, the main differences between 3D-CNN and CNN network structure are as follows. Firstly, the research object is different, the former is video (frame sequence), the latter is picture. Secondly, the input layer is different, the input of 3D-CNN is five. In addition to the size, the number of channels, and the batch size of the video frame sequence, the number of frames of one-dimensional video frame sequence is increased. Thirdly, the core size of convolutional layer and pooling layer is different. The size of the core in 3D-CNN is one more time length, and the structures of other layers are similar. The basic flow chart of the 3D-CNN network processing video frame sequence is shown in Fig. 2.

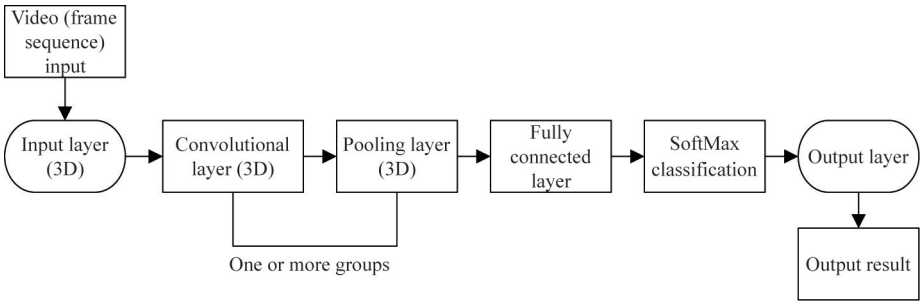

Fig. 2 The basic flow of 3D-CNN processing video frame sequence

1.2 3D-VGG network structure

VGGNet was proposed in Ref. [18], and won the first runner-up in the Image Challenge image recognition competition. It has six different network structures. Each network structure contains five groups of convolutions, and the size of the convolution kernel is 3×3. After each group of convolutions, the maximum pooling of 2×2 is performed. The biggest innovation of VGGNet is to use multiple 3×3 convolution stacks in sequence to obtain a larger receptive field effect. It can be seen that on the premise of the same receptive field, a smaller convolutional kernel stack can be used to replace a larger convolutional kernel, which can ensure that the number of parameters is greatly reduced, thus increasing the depth of the network.

Generally speaking, the design principle of VGGNet is to use a smaller 3×3 convolution kernel, and the design ideas are mainly continuation of AlexNet's ideas. On this basis, try to build a network with more layers and deeper. The main difference lies in the convolution kernel of VGGNet and the size of each convolution layer, instead of only performing one operation, convolution is performed consecutively for many times. Therefore, by analyzing a variety of commonly used convolutional neural network models and weighing the calculation amount and performance of the experiments in this paper, VGGNet is selected as the improved model of the algorithm to realize the feature extraction of micro expressions.

The improved 3D-CNN network structure in this paper is three-dimensional, while the VGGNet network is two-dimensional. Therefore, the size of the convolutional layer and the pooling layer core needs to be modified first. As shown in Fig. 3, a VGG module is changed to a three-dimensional VGG module. It can be seen from Fig. 3 that the size of the convolution kernel and the size of the pooling window have been changed from two-dimensional 3×3, 2×2 to $3 \times 3 \times 3$, $2 \times 2 \times 2$.

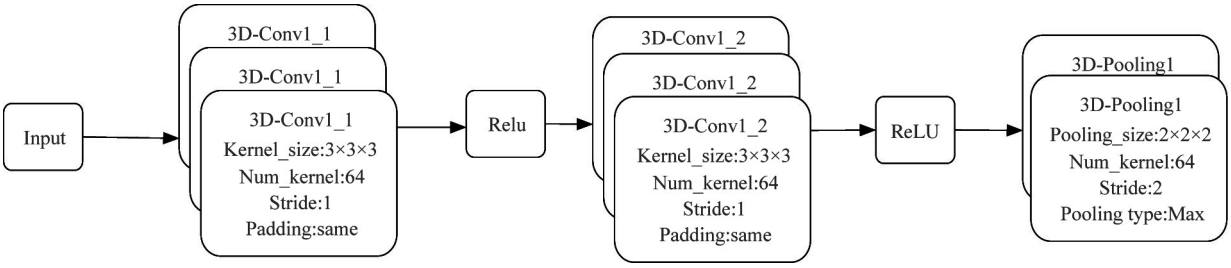

Fig. 3 3D-VGG module

1.3 Overall network design plan

This paper completes the overall design of the corresponding network structure based on the deep learning framework Keras. The network structure model uses a linear graph model, and the back-end is implemented by TensorFlow. The overall network algorithm diagram is shown in Fig. 4, where the input of the network is a sequence of micro-expression images. The improved 3D-CNN network realizes the joint extraction of spatial and temporal features, and uses the Softmax function to classify the extracted feature vectors, completing the entire algorithm flow.

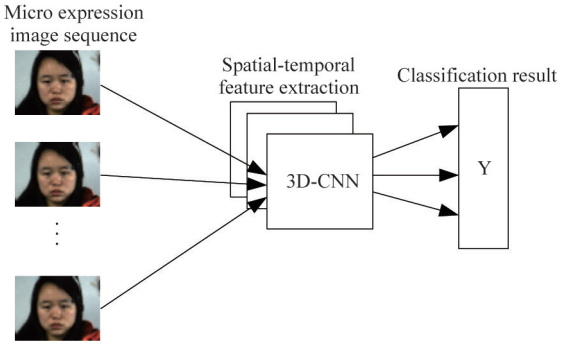

Fig. 4 Schematic diagram of the overall network algorithm

Since the training of the network requires artificial adjustment of the hyperparameters of the network, such as learning rate, parameter initialization, number of iterations, weight attenuation coefficient, dropout ratio, most of the time when training the network is used to adjust the hyperparameters. When network training starts, the change of parameters will also have an impact on the distribution of input data at each layer. Therefore, during training, in order to match the new data distribution in time, each layer of the network needs to be updated, which will cause slow convergence and hinder the emergence of network training situations, and as the network depth increases, it will become more and more difficult to train. In order to reduce the impact of the above problems and speed up the training speed, this article applies the batch normalization (BN) algorithm to each layer of input data, and the data distribution is more stable after the normalization process. At the same time, in the process of neural network model training, there will be overfitting. This situation occurs because there are too few training samples and more model parameters. In order to effectively alleviate the model over-fitting, this paper adopts the Dropout algorithm, commonly known as the drop method.

2 Overall network structure and performance analysis

Based on the 3D-VGG-Block design, the overall network structure is completed. The pre-network structure of 3D-CNN and the fully connected layer network structure are shown in Fig. 5 and Fig. 6, respectively. Model training difficulties and over-fitting are common problems in deep learning network training. In order to effectively alleviate these problems, the BN layer and the Dropout layer are sequentially added to the structure.

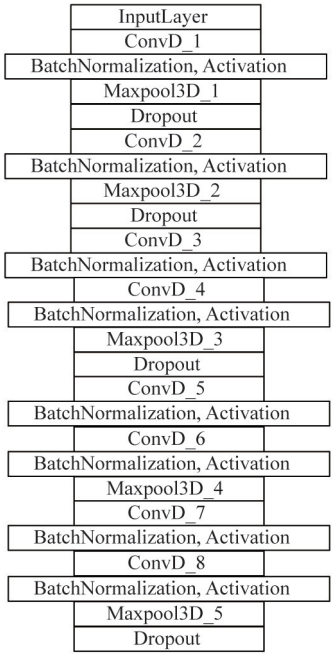

Fig. 5 3D-CNN pre-network structure

For the input layer of the network, the input size is (None, 16, 128, 128, 3), where the number of training samples is represented by None, the number of original image channels is 3, and the image resolution size after normalization processing is 128 × 128, the sliding window step length and padding are both 1, 16 represents the length of the video sequence in the time dimension. The number of Conv3D_1 and Conv3D_2 layer convolution kernels is set to 32; the number of Conv3D_3 and Conv3D_4 layer convolution kernels is 64, Conv3D_5, Conv3D_6, Conv3D_7, and Conv3D_8 layer convolution kernels are 128. Considering that some valuable information may be lost due to downsampling, and at the same time trying to reduce the amount of calculation, therefore, the setting of the first pool layer is (1, 2, 2). For all the time-domain sequence information, only the sampling procedure is

performed on the two-dimensional space domain, pooling layer processing is performed, sampling and dimensionality reduction operations are performed on the time-domain sequence features, and the time-domain sequence features are extracted and compressed. The pooling window sizes are all (2, 2, 2); the activation function ReLU is applied to all activation layers; the dropout discard rate is both 0.25, and the fully connected layer is 0.5.

In Fig. 6, after multiple convolutional layers and pooling layers, the output of the last convolutional layer is (None, 1, 4, 4, 128), which is commonly used between the convolutional layer and the fully connected layer transition layer. The input of the Flatten layer is (None, 1, 4, 4, 128), the output is (None, 2048), and then entering the first fully connected layer (Dense1), the input length value is 2048 feature vector. In order to obtain more valuable information results, the length of the feature vector is shortened, the output value is set to 1024, and finally, in the second fully connected layer (Dense2), the Softmax function is used to classify the output vector of length 1024. For the CASME data set, the number of neurons is 4, and for the CASME II data set, the number of neurons is 5.

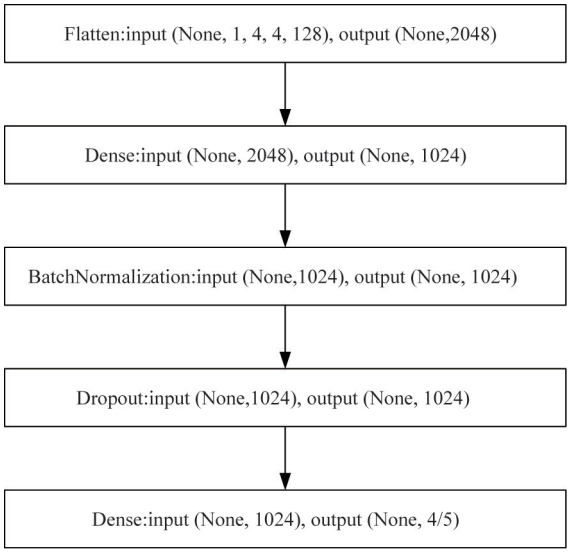

Fig. 6 Fully connected layer network structure

3 Experiment and analysis

3.1 Experimental environment

This article uses the following experimental environment (Table 1) in the research process.

3.2 Data set and preprocessing

3.2.1 Data set selection

The micro expression data sets used in this experiment are CASME and CASME II. In 2013, Ref. [19] researched and established the CASME data set. Nineteen subjects participated in the normal shooting, a total of 195 micro-expression samples were generated. The CASME data set contains two parts, CASME_A and CASME_B. CASME_A is under natural light conditions, the resolution of the micro-expression camera is 1280×720 and the frame rate is 60 fps. CASME_B uses two symmetrical LED lights and uses a camera with a resolution of 640×480 and a frame rate of 60 fps. The resolution of the captured face image area is 150×190, which is divided into eight categories, namely, tense, disgust, repression, surprise, happiness, sadness, contempt and fear. Fig. 7 is a micro-expression image belonging to the surprised category in CASME.

Table 1 Hardware and software environment

Name	Model (version)	Description
TensorFlow	1.2.0	As the backend of Keras
Keras framework	2.1.1	Use TensorFlow as backend
CUDA	CUDA10.1	Applied in the underlying software platform of GPU
Operating system	Linux	Ubuntu 18.04
GPU	GTX 1080Ti	Video memory 11 GB
RAM	Kingston HyperX Savage DDR4	Main frequency 2400 MHz, 8 GB
Hard disk	Seagate	1 TB

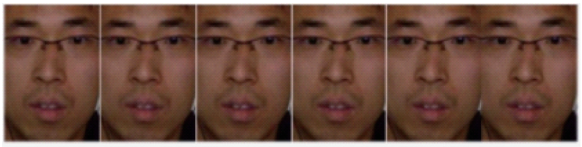

Fig. 7 Surprise class

In 2014, Ref. [20] researched on the basis of the CASME data set and created the CASME II data set, which is now the most scientific and reasonable micro-expression data set recognized by researchers from various countries. A total of 255 micro-expression sequences were taken by 26 subjects. The camera frame rate was 200 fps and the image resolution was 280×340. All samples are spontaneous, and the light is sufficient and stable when shooting the video, including sadness, happy, fear, surprise, disgust, depression, and others. Fig. 8 is a micro-expression image belonging to the happy category in CASME II.

Fig. 8 Happy class

According to the distribution of the number of

samples of each type of micro-expression in the two data sets of CASME and CASME II, it is known that the number of samples of individual types is too small, and the research is of little significance. The data set of the experiment in this article finally uses the four types of sample data in CASME (tension, surprise, depression, disgust) and the five types of sample data in CASME II (surprise, depression, happiness, disgust, and others).

3.2.2 Data set preprocessing

When training a deep neural network, a large amount of training data is often required. Sufficient sample data will improve the training effect of the network, otherwise it is prone to overfitting. However, one of the main problems faced by micro-expression recognition research is the lack of data sets, and the number of data set samples that can be used as research applications is currently very small. Therefore, before the network training and testing, this article firstly performs image data enhancement processing on the two experimental data sets.

In deep learning, commonly used image data enhancement methods include image flipping, rotation, zooming, cropping, translation, and adding noise, etc. However, different from general expression features, the micro-expression duration is short and the action intensity is weak. It is ensured that the dynamic feature information and motion information of the micro-expression are not affected as much as possible. Therefore, this paper adopts two data enhancement methods, image flipping and small amplitude flipping. The image flip used in this article is different from rotating 180 °, but a way of flipping the image symmetrically about the Y axis, similar to mirror flipping. Fig. 9 is a partial result after flipping. The data set is expanded to twice the original. Next, take 5 °, −5 °, 10 °, and −10 ° image rotations for the original sample and the flipped sample, respectively. Fig. 10 shows the partial results after rotation. Finally, the processed data is sorted into the original sample, the original sample rotated by four angles, the flipped sample, and the flipped sample rotated by four angles. In summary, the number of samples has been expanded to ten times the original.

Fig. 9 Partial results of image flip

Fig. 10 Partial results of image rotation

Finally, in the network data input module, directly call the cvtColor function in OpenCv to grayscale the input RGB image, and then call the resize function in OpenCv, and the normalized image size is 128 × 128 as the data input. When performing the data reading operation during the training process, first read the frame sequence with a fixed length of 16 each time, and then read the length value of each frame sequence in the obtained data set. If one of the frame sequences is the length of is R, then randomly obtain a number M in the range of $(0, R-16)$ and set it as the starting frame. The entire frame is within the range of $(M, M+16)$, and the above operation is continuously executed until all the sample data is traversed, and the purpose of data amplification is achieved.

3.3 Analysis of experimental result

Because it often takes as little as 10 h to complete the entire neural network training, and as many as tens of hours, it is necessary to carefully and thoroughly record the experimental data. Based on the research and analysis of the records, it is found that for existing problems, adjust the parameters in the network in time and update the record results. This article uses the History module in the Keras callback function Callbacks to record the experimental data. Once the training meets the conditions of a certain stage, the function set can be called. In the process of network training, it can be used the Callback function to record the statistical information of the network and its internal conditions. This article uses the Sequential network model, which includes the fit function. This function will transmit logs information to the Callback function, which contains on _ epoch _ end data to record the experimental data in this article.

3.3.1 Network training

According to the overall network structure design in subsection 1.3, the network is trained and tested based on the deep learning Keras framework. At the beginning of this section, the network is trained first. The training set uses the augmented data of the CASME

and CASME II data sets. The network configuration parameters during training are determined after multiple experiments based on the experimental software and hardware environment and other factors. The learning rate (Base _ lr) is 0.0001, the learning rate decay is 1e-10, the momentum is 0.9, and the batch size is 16. When the number of iterations continues to increase, in order to effectively avoid divergence of the loss function, especially the number of iterations is limited. When the number of iterations reaches 20 000, the learning rate is one-half of the original, and the same operation is continued until ten loop operations are completed. When the number of iterations reaches 200 000, set the iteration epoch to 500. In this experiment, 400 epochs were performed, and the loss function value and accuracy value were recorded in detail. Fig. 11 and Fig. 12 show the accuracy curve and loss function curve of the training process on the CASME data set. Fig. 13 and Fig. 14 show the accuracy curve of the training process on the CASME II data set. The training process diagram records the state of the entire network during training. According to the accuracy curve and loss function curve, the convergence speed and convergence state of the network are known as the number of iterations of the network increases.

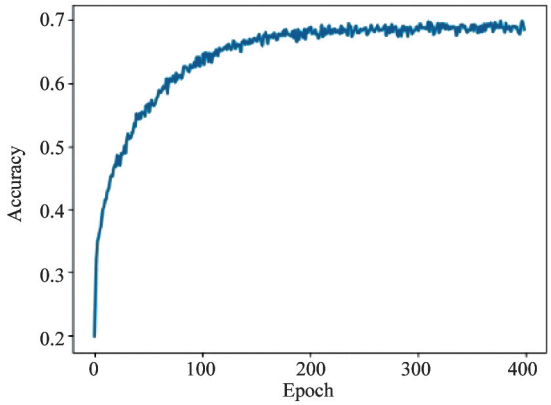

Fig. 11 Accuracy curve during the CASME training process

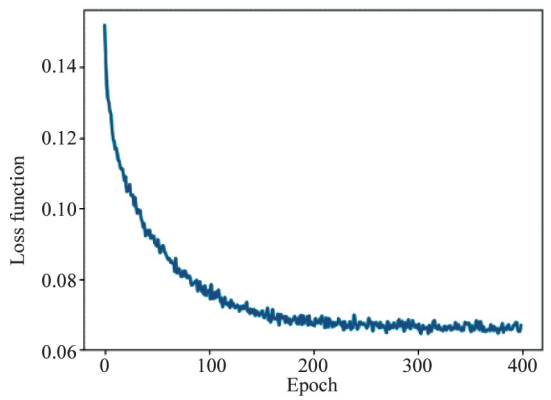

Fig. 12 The loss function curve during the CASME training

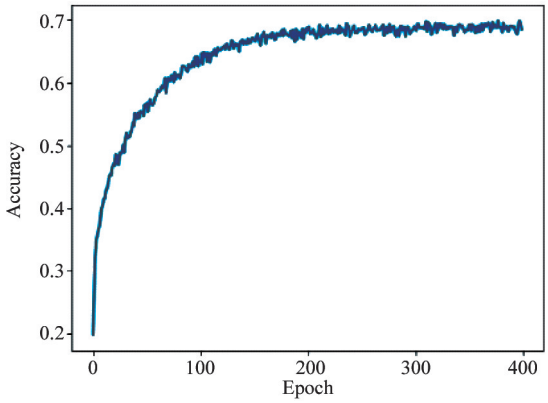

Fig. 13 Accuracy curve during the CASME II training process

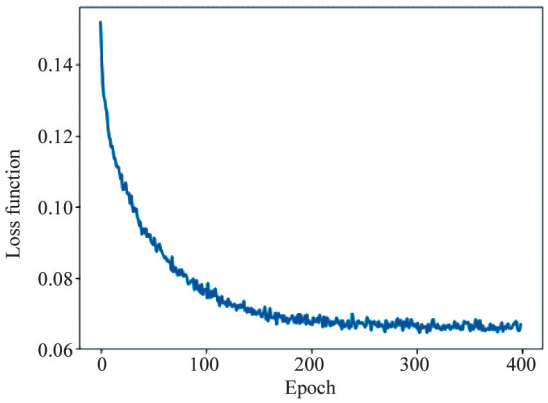

Fig. 14 The loss function curve during the CASME II training process

3.3.2 Network test and analysis

After the network training process is over, in order to verify the quality of the network model, the trained model is tested, and the test sample uses the original sample of the data set. Through the network test, the confusion matrices on the two data sets of CASME and CASME II were obtained, as shown in Fig. 15 and Fig. 16. From the confusion matrix, it can

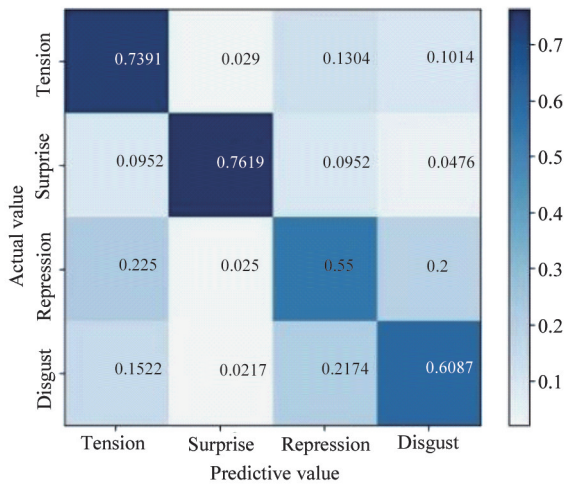

Fig. 15 Confusion matrix in CASME

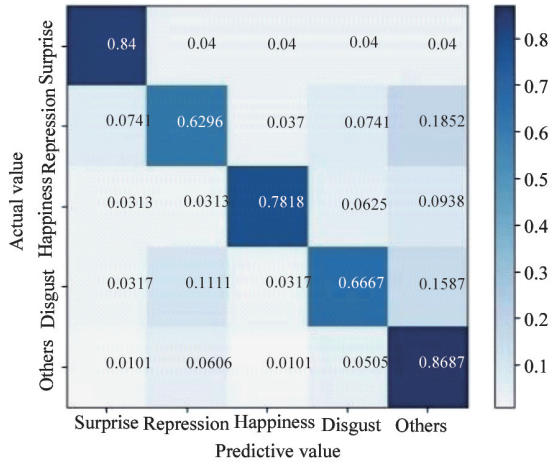

Fig. 16 Confusion matrix in CASME II

be seen that in CASME, the model has the highest recognition rate for the surprised category and the lowest recognition rate for the depressed category. In CASME II, the model has the highest recognition rate for other categories and the lowest recognition rate for the depressed category. This may be that there are too few training samples in these categories, which leads to insufficient dynamic feature extraction capabilities, and the imbalance of training samples leads to different recognition rates.

According to the confusion matrix of the test results, the accuracy of each category and the overall accuracy of the test on the two data sets are sorted, as shown in Fig. 17. Surprise, happiness and other categories have higher accuracy rates, and the recognition rates on the CASME II dataset are 84%, 78.13% and 86.87%, respectively. The accuracy rates of the repression and disgust categories are far from the higher three categories. The recognition rates on the CASME II dataset are 62.96% and 66.67%, respectively. The recognition rates of tension and surprise categories

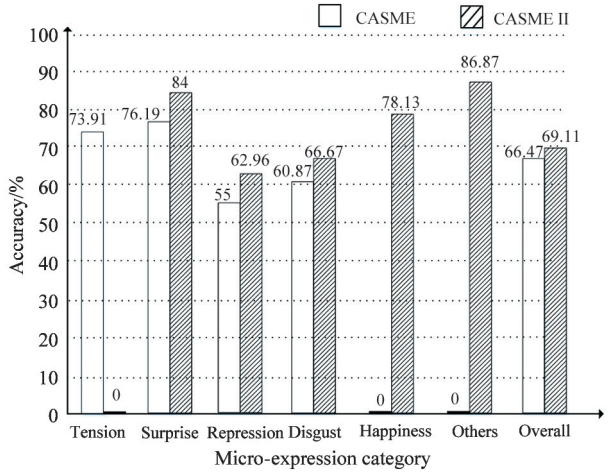

Fig. 17 Summary of test results

on the CASME dataset are higher, reaching 73.91% and 76.19%. The repression category has the lowest recognition rate, with an accuracy rate of 55% on the CASME data set. In the final comparison and analysis of various recognition rates, the overall accuracy rate on the CASME data set is 66.47%, and that on the CASME II data set is 69.11%.

In order to illustrate the effectiveness of data preprocessing before training the network, this article adds a comparison experiment at the end to compare the data sets before and after data enhancement processing. The final recognition rate results of the overall class on the CASME and CASME II data sets are shown in Table 2. In the CASME data set, the recognition rate without data amplification processing is 60.76%, while the recognition rate after processing is 66.47%; In the CASME II data set, the recognition rate before data amplification processing is 62.95%, and the recognition rate after processing is 69.11%.

Table 2 Comparison of accuracy before and after data amplification

	Before data amplification/%	After data amplification/%
CASME	60.76	66.47
CASME II	62.95	69.11

Finally, the improved algorithm in this paper is compared with other micro-expression recognition algorithms. The results are listed in Table 3. From Table 3, it can be seen that the test results of the improved 3D-CNN algorithm proposed in this paper are higher than that of other algorithms. This verifies the effectiveness of the algorithm in this paper, and also proves the importance of extracting spatial and temporal features at the same time for feature extraction based on video frame sequence for micro-expression data amplification.

Table 3 Comparison of accuracy between the algorithm in this paper and other algorithms

Algorithms	Accuracy/%
2D-CNN[14]	64.90
S-LRCN[21]	65.70
3D-CNN[17]	66.67
CNN + GEME[22]	67.48
Algorithm in this paper	69.11

4 Conclusion

Aiming at the problem of the low detection rate of current micro-expression recognition algorithms, espe-

cially when the amount of sample data is insufficient, an improved three-dimensional convolutional neural network micro-expression recognition algorithm is proposed. While improving the structure of the 3D-CNN model, this algorithm introduces a BN layer to increase the training speed. In order to improve the generalization ability of the model and reduce the risk of overfitting, this algorithm introduces Dropout layer. The improved 3D-CNN network was trained and tested on the data sets of CASME and CASME II. The experimental results show that the micro-expressions recognition effect of 3D-CNN algorithm is improved to some extent. The maximum recognition rate of the experimental results is 69.11%, which verifies the feasibility of the algorithm. However, how to design and implement a more robust and accurate feature extraction algorithm, establish a real-time automatic recognition system for micro-expression, and apply it to complex real-life fields require the further research.

References

[1] LI J, WANG Y D, SEE J, LIU W H. Micro-expression recognition based on 3D flow convolutional neural network [J]. *Pattern Analysis and Applications*, 2019, 22 (4): 1331-1339

[2] HE J C, HU J F, LU X, et al. Multi-task mid-level feature learning for micro-expression recognition[J]. *Pattern Recognition*, 2017, 66: 44-52

[3] LI Q, YU J, KURIHARA T, et al. Deep convolutional neural network with optical flow for facial micro-expression recognition [J]. *Journal of Circuits, Systems and Computers*, 2020, 29(1): 1-18

[4] TAMMINA S. Transfer learning using VGG-16 with deep convolutional neural network for classifying images [J]. *International Journal of Scientific and Research Publications*, 2019, 9(10): 143-150

[5] PFISTER T, LI X B, ZHAO G Y, et al. Recognising spontaneous facial micro-expressions [C] // The 2011 IEEE International Conference on Computer Vision, Barcelona, Spain, 2011: 1449-1456

[6] BEN X, JIA X, YAN R, et al. Learning effective binary descriptors for micro-expression recognition transferred by macro-information [J]. *Pattern Recognition Letters*, 2018, 107: 50-58

[7] HINTON G E, OSINDERO S, TEH Y W. A fast learning algorithm for deep belief nets[J]. *Neural Computation*, 2014, 18(7): 1527-1554

[8] NAM H, HAN B. Learning multi-domain convolutional neural networks for visual tracking[C] // Computer Vision and Pattern Recognition, Las Vegas, USA, 2016: 4293-4302

[9] HE K, GKIOXARI G, DOLLAR P, et al. Mask R-CNN [C] // IEEE International Conference on Computer Vision, Venice, Italy, 2017: 2980-2988

[10] LI Z, LI F, ZHU L, et al. Vegetable recognition and classification based on improved VGG deep learning network model [J]. *International Journal of Computational Intelligence Systems*, 2020, 13(1): 559-564

[11] KIM D H, BADDAR W J, RO Y M. Micro-expression recognition with expression-state constrained spatio-temporal feature representations [C] // The 24th ACM International Conference on Multimedia, Amsterdam, The Netherland, 2016: 382-386

[12] PATEL D, HONG X, ZHAO G. Selective deep features for micro-expression recognition [C] // International Conference on Pattern Recognition, Amsterdam, Netherlands, 2017: 2258-2263

[13] LI X, YU J, ZHAN S. Spontaneous facial micro-expression detection based on deep learning [C] // 2016 IEEE 13th International Conference on Signal Processing, Chengdu, China, 2016: 1130-1134

[14] MAYYA V, PAI R M, PAI M MM. Combining temporal interpolation and DCNN for faster recognition of micro-expressions in video sequences [C] // International Conference on Advances in Computing, Communications and Informatics, Jaipur, India, 2016: 699-703

[15] LUO M, CAO J, MA X, et al. FA-GAN: face augmentation GAN for deformation-invariant face recognition [J]. *IEEE Transactions on Information Forensics and Security*, 2021, doi:10.1109/TIFS.2021.3053460

[16] PAN H, XIE L, LV Z, et al. Hierarchical support vector machine for facial micro-expression recognition [J]. *Multimedia Tools and Applications*, 2020, 79(3): 1-15

[17] JI S, XU W, YANG M, et al. 3D convolutional neural networks for human action recognition [J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013, 35(1): 221-231

[18] ABBAS A, ABDELSAMEA M M, GABER M M. Classification of COVID-19 in chest X-ray images using DeTraC deep convolutional neural network [J]. *Applied Intelligence*, 2021, 51(2): 854-864

[19] YAN W J, WU Q, LIU Y J, et al. CASME database: a database of spontaneous micro-expressions collected from neutralized faces [C] // IEEE International Conference and Workshops on Automatic Face and Gesture Recognition, Shanghai, China, 2013:1-7

[20] YAN W J, LI X, WANG S J, et al. CASME II: an improved spontaneous micro-expression database and the baseline evaluation[J]. *PLoS ONE*, 2014,9(1):1-8

[21] LI X H, HU S Q, SHI Z G, et al. Micro-expression recognition algorithm based on separate long-term recurrent convolutional network [J]. *Chinese Journal of Engineering*, 2022, 44(1): 104-113 (In Chinese)

[22] NIE X, TAKAlKAR M A, DUAN M , et al. GEME: dual-stream multi-task GEnder-based micro-expression recognition[J]. *Neurocomputing*, 2021, 427(5):13-28

WU Jin, born in 1975. She received her B. S. degree from Xi'an Jiaotong University in 1998, and she also received her M. S. degree from Xi'an Jiaotong University in 2001. Her research focuses on key techningues for signal and information processing.