J Inf Process Syst, Vol.17, No.2, pp.385~398, April 2021
https://doi.org/10.3745/JIPS.01.0072

ISSN 1976-913X (Print)
ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

Residual Learning Based CNN for Gesture Recognition in Robot Interaction

Hua Han*

Abstract

The complexity of deep learning models affects the real-time performance of gesture recognition, thereby limiting the application of gesture recognition algorithms in actual scenarios. Hence, a residual learning neural network based on a deep convolutional neural network is proposed. First, small convolution kernels are used to extract the local details of gesture images. Subsequently, a shallow residual structure is built to share weights, thereby avoiding gradient disappearance or gradient explosion as the network layer deepens; consequently, the difficulty of model optimisation is simplified. Additional convolutional neural networks are used to accelerate the refinement of deep abstract features based on the spatial importance of the gesture feature distribution. Finally, a fully connected cascade softmax classifier is used to complete the gesture recognition. Compared with the dense connection multiplexing feature information network, the proposed algorithm is optimised in feature multiplexing to avoid performance fluctuations caused by feature redundancy. Experimental results from the ISOGD gesture dataset and Gesture dataset prove that the proposed algorithm affords a fast convergence speed and high accuracy.

Keywords

Convolutional Neural Network, Feature Redundancy, Full Connection Layer, Gesture Recognition, Human-Computer Interaction, Residual Learning

1. Introduction

In recent years, owing to the development of smartphones and the rapid progress of human–computer interaction technology, interaction methods such as touch screens, voice recognition, fingerprint recognition, and gesture recognition have emerged [1-3]. A touch screen can completely replace the traditional mouse and keyboard, whereas voice recognition allows us to control a robot only through our mouth. Furthermore, with the introduction of fingerprint recognition technology, complex and cumbersome passwords are no longer required, and the privacy of users is protected consequently. Computer vision allows us to operate a computer in a state completely detached from fixed peripherals through somatosensory devices, and investigations focusing on the iris, palm prints, voice, gestures, and other human features have been performed to investigate new interaction approaches. Gesture recognition, as a critical topic in human-computer interaction research, has garnered increasing attention in the field of artificial intelligence [4].

Manuscript received July 3, 2020; first revision September 15, 2020; accepted September 20, 2020.

Corresponding Author: Hua Han (kfdxhhjq@163.com)

* School of Mechanical and Automotive Engineering, Kaifeng University, Kaifeng, Henan, China (kfdxhhjq@163.com)

Gesture recognition is based on human hand movements. Human hands are flexible, and gestures are used to measure hand movements. According to changes in gestures, images or syllables are simulated to form certain meanings or words, which are used for communication between people. Body language for communicating ideas is an "important auxiliary tool for voiced language." Specific groups of people with hearing impairments use body language as their primary communication tool; furthermore, body language can be used in a wide range of applications [5,6]. In industrial production, robot teaching is an involved and complicated task. Controlling robot movements through gestures can simplify the teaching process and operation process of industrial robots [7].

Deep learning is generally believed to have originated in 2006. According to Bengio's definition, deep networks comprise multiple layers of adaptive nonlinear units, i.e., a cascade of multiple layers of nonlinear modules. All levels contain trainable parameters. During operation, a deep neural network is typically a five-layer or larger network that contains millions of learnable free parameters [8,9]. In theory, the network model can approximate the internal relationships and essential characteristics of data through functions, regardless of the depth. However, when solving complex real-world problems, exponentially growing computing units are required. Shallow networks often have insufficient function expression capabilities. Deep networks may require fewer computing units. However, in theory, the network is not as deep as one might envision. In addition to the problem of memory occupation due to the excessive number of layers, the problem of gradient disappearance or gradient explosion is encountered [10].

Therefore, an efficient gesture recognition network based on the combination of convolutional neural networks (CNNs) and deep residual learning modules is proposed herein to avoid performance fluctuations caused by feature redundancy.

2. Related Research

Humans are adept at recognising gesture information and providing relevant responses such that they can communicate with gesture-assisted expression effortlessly. In recent years, research regarding the calculation decision classification of gesture information obtained by machine has received significant attention [11-13]. Gesture recognition technology can be used to apply gesture classification and recognition results to electronic devices, intelligent robot manipulation, or the auxiliary transmission of sports medical information. Furthermore, gesture recognition has been investigated intensively in human–computer interaction fields [14,15].

The key aspect in the traditional static gesture recognition algorithm is the extraction of gesture features. Xue et al. [16] segmented gestures in the YCbCr space, extracted Hu moments, combined with Fourier descriptors to obtain desired features, and then performed training and recognition through a back propagation neural network. The two algorithms above can recognise gestures promptly; however, the segmentation effect of gestures has a more significant effect on the recognition rate. Lu et al. [17] improved gesture feature extraction and feature point matching based on data gloves; the results indicated that the accuracy and speed of gesture recognition of data gloves improved, but the recognition rate decreased as the number of gesture types increased. Cai et al. [18] used a support vector machine algorithm to extract gesture features and then combined it with an artificial neural network, hidden Markov model, and dynamic time warping algorithm to recognise gestures. A high recognition rate was achieved, but the combined algorithm demonstrated a slow operation speed and was unsuitable for real-time systems.

A deep learning algorithm is a specific machine learning algorithm that has revolutionised many data analysis fields. It differs from traditional machine learning methods in that feature extraction is part of the model definition; as such, it is not required to be artificially large. Deep learning algorithms contain self-learning features that enable features to be extracted in a short time. Park and Lee [19] proposed the first end-to-end deep learning architecture and established a surface myoelectric hand motion classification model based on CNNs. Compared with support vector machines, the classification accuracy obtained was higher. Atzori et al. [20] established a simple convolutional network to classify a large number of gestures, and the classification accuracy was comparable to that of classic classification algorithms. Geng et al. [21] improved their results using various data sets as well as transient surface EMG images to obtain an 89.3% accuracy in a set of eight actions. Wei et al. [22] used a multistream CNN architecture to segment the input into smaller images, which were then processed by the convolutional layer and connected to the fully connected layer, resulting in an 85% recognition accuracy on the Ninapro dataset. Although the abovementioned deep-learning-based gesture recognition methods demonstrate high recognition accuracy, they still do not satisfy user requirements.

Fang et al. [23] proposed a gesture recognition algorithm based on a CNN and a deep convolution generative adversarial network (DCGAN). The method, which has been applied to expression recognition, calculation, and output texts, yielded good results. Moreover, this gesture recognition method is not easily affected by illumination or background interference. Tan et al. [24] proposed a static gesture recognition framework based on electromagnetic scattering field data learning, which effectively solved some problems in traditional visual recognition methods. An end-to-end complex-valued attention CNN was designed for training gesture recognisers, where the attention module was used to learn robust perceptual features of the region of interest. A large number of numerical experiments have been performed on a public static gesture dataset, and a high recognition rate was achieved. Alani et al. [25] proposed an adaptive deep convolutional neural network (ADCNN) for gesture recognition tasks. The ADCNN model was initialised using a network comprising a ReLU and softmax as well as L2 regularisation to eliminate data overfitting. Experimental results show that the ADCNN model effectively improved gesture recognition. Although these methods achieved satisfactory recognition accuracy, the complexity of the deep learning models used affected the real-time performance of gesture recognition and limited the application of gesture recognition algorithms in actual scenarios. Hence, a residual learning neural network based on a deep CNN is proposed. This network overcomes the problem of excessive deep network parameters to a certain extent and reduces the possibility of gradient dispersion problems. The main contributions of this study are as follows:

Small-scale networks (3×3 convolution kernels, focusing on local detailed information) are used instead of traditional large-scale networks (7×7 convolution kernels, focusing on global detailed information) to extract features, and the total number of model parameters is reduced.

By constructing a residual structure, the problem of gradient disappearance or gradient explosion caused by the deepening of the network layer is avoided. The deep neural network affords not only a fast convergence speed, but also high accuracy.

3 Residual-Learning-Based Deep CNN

Inspired by the InceptionV3 module in GoogleNet [26,27], an efficient recognition network based on the combination of a CNN and a deep residual learning module (residual block) is proposed to solve the

problem where the abstraction of deep networks results in reduced spatial concentration. In this regard, a gesture image is placed into the deep residual learning network by a random input, and the jump connection is used to fuse shallow and deep features to realise an interaction between different levels of information; subsequently, the output is combined with the CNN to improve the ability to distinguish features. Herein, the training sample is denoted as $\{y_t\}_{t=1}^{N}$, where the input test gesture image is $x \in \Re^{m \times n}$, and N represents the total number of pictures; the proposed network structure is shown in Fig. 1.

The network structure can be categorised into four segments: the first segment is for the initial feature extraction, where a small-scale network (3×3 convolution kernel, focusing on local detailed information) is used to replace the traditional large-scale network (7×7 convolution kernel, focus on global detail information) to extract features, thereby reducing the total number of model parameters; in the second segment, a residual structure is built to learn features and simplify the model optimisation difficulty; in the third segment, a CNN is used to extract the effectiveness of each node from the deep feature information; the fourth segment pertains to identification, where the cascade softmax classifier from the fully connected layer is used to output the predicted category of the model C_t; finally, the loss back propagation is calculated to fine-tune the learning weights and optimise the overall model.

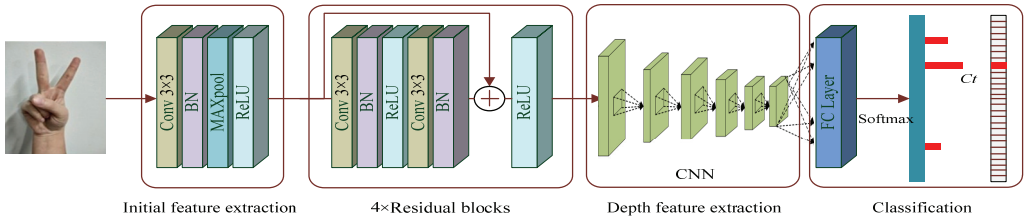

Fig. 1. Proposed deep residual network.

3.1 Deep Residual Learning

The increase in the depth of deep learning networks significantly affects their classification. Researchers discovered that adding a significant number of weight parameters will improve the recognition performance of the network to a certain extent; however, when the conventional CNN is stacked to a certain number of layers, the network gradient often disappears, i.e., the network recognition rate no longer increases. This is because the training of deep networks will cause a gradient dispersion/explosion [28,29]. Using the idea of residual learning, the desired learning goal of the neural network is defined as Y, where X is the feature obtained by convolution, batch normalization (BN), ReLU, and max-pooling of the original image input; $F(X, W_t)$ is the feature extracted after two 3×3 convolutions of X; W_t is the learned weight parameter. The goal that the network must learn is simplified to $F(X, W_t) = Y - X$. The residual learning module is used to effectively alleviate the problem of small-chain derivative gradient values and optimise the training process.

3.2 Convolutional Neural Network

The hidden layer in a CNN is an important layer. The classic CNN comprises an input layer, a convolutional layer, a pooling layer (down-sampling layer), a fully connected layer, and an output layer. A CNN exhibits the characteristics of local vision sharing, weights and pooling sharing, etc. [30].

The convolutional layer is the core layer structure of the CNN. The quality of the image feature map extracted by the convolutional layer directly affects the processing of subsequent layers. The convolutional layer is primarily connected to the input two-dimensional image data through the convolution kernel and yields the local characteristics of the image data, and the feature map is obtained via the weight-sharing method. Each neuron in the convolutional layer is connected to the local region (local receptive field) of the previous layer of feature maps through the convolution kernel. The role of the convolution kernel is to extract the features of the local region. Once the local features are extracted, by sliding the convolution kernel on the previous layer of feature maps and traversing all regions, the characteristics of each local region are determined. Weight sharing refers to the fact that each neuron in the convolutional layer is locally connected to the previous layer with the same set of connection weight parameters, i.e., the same convolution kernel is used to convolve the image of the previous layer such that the network is trained. The weight parameter at that time is reduced. Different convolution kernels are used to traverse the feature map of the previous convolution layer (plus the bias), and the current neuron is obtained using the activation function to form different feature maps. The formula to calculate the convolution layer can be expressed as

$$y_j^l = f\left(\sum_{i=1}^{N_j^{l-1}} w_{i,j} \otimes x_i^{l-1} + b_j^l\right), j = 1, 2, \cdots, M \tag{1}$$

In the formula, l represents the current network layer number, and $l-1$ is the previous layer network; y_j^l represents the jth feature map of the current convolution layer; x_i^{l-1} represents the ith feature map of the previous layer; $w_{i,j}$ is the convolution kernel between i and j. x_i^{l-1} shown in Fig. 1 is a 10×10 feature map; the convolution kernel $w_{i,j}$ has dimensions of 5×5 and is convolved with the 5×5 area in x_i^{l-1}, adds biases b_j^l, and passes through the activation function $f(x)$ before the value of the corresponding area in y_j^l is obtained. The same convolution kernel $w_{i,j}$ slides on x_i^{l-1} with a step size of 1; subsequently, the entire feature map y_j^l is obtained through the steps above.

The pooling layer is also known as the down-sampling layer, which often follows the convolution layer. The increase in the number of convolutional layers will cause the number of feature maps to increase accordingly; therefore, the learned feature dimension will increase rapidly. The role of the pooling layer is to reduce the feature dimension of the feature map after convolution to reduce the network complexity and reduce the amount of calculation. For different requirements, different pooling methods are used in the pooling layers. Maximum pooling and mean pooling are the two most frequently used pooling methods. Assuming that the sample size is 2×2, i.e., the feature map input by the convolutional layer is segmented into 2×2 small blocks, the maximum pooling operation was used to extract the largest value in each small block to form a new feature. The mean pooling operation used the mean of the parameters in the small block. Through the pooling operation, if the image is slightly shifted, the pooling result will not change accordingly; therefore, the robustness of the system will improve. The general expression for pooling is

$$y_j^l = down\left(y_j^{l-1}\right) \tag{2}$$

In the formula, l indicates the current layers, j the feature map, $down(g)$ the pooling function.

The fully connected layer is a type of convolutional layer as well; however, it differs from the local

connection of the convolutional layer. Each neuron in the fully connected layer is connected to all neurons in the previous layer, but the neurons in this layer are not mutually connected. A full connection function is a dimensional transformation that transforms the high-dimensional matrix data of the previous layer into a low-dimensional matrix as well as extracts and integrates the distinguishing features. Additionally, it expresses the implicit semantics and maps the original features to each implicit semantic node. The general expression of the fully connected layer is

$$y_j^l = f\left(\sum_{i=1}^{n} x_i^{l-1} \bullet w_{i,j}^l + b_j^l\right) \tag{3}$$

Similar to the convolution layer, where n is the number of neurons in the previous layer, l represents the current layer, w_{ji}^l the connection weight parameter between the current layer y_j^l and previous layer x_i^{l-1}, b_j^l the biases, and $f(g)$ the activation function.

The third segment of the proposed deep convolutional residual network uses a CNN to extract the effective information of each node in the deep features. As shown in Fig. 2, without an input, the CNN contained eight layers, including three convolutional layers, three pooling layers, one fully connected layer, and one softmax regression layer. The network input contained 224×224 gesture pixels. In the matrix, the size of convolution layer 1 convolution kernel was 5×5, the number of convolution kernels was 32, and 32 feature maps were obtained. The sampling size of pooling layer 1 was 2×2 (non-overlapping sampling), corresponding to convolution layer 1, and 32 feature maps were obtained. Similarly, the size of the convolution kernels of convolution layers 2 and 3 was 5×5; therefore, the number of convolution kernels was 64, corresponding to pooling layers 2 and 3. The number of neurons in the fully connected layer was set to 500, and pooling layer 3 was completed. The fully connected layer followed by the softmax regression layer contained six neurons. The features output by the fully connected layer were classified, and a total of six gestures from 0 to 5 were obtained.

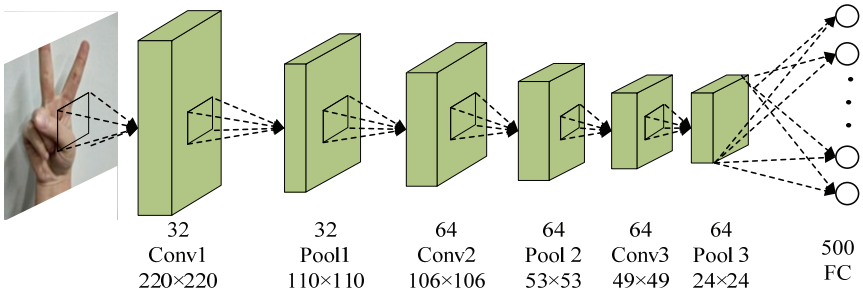

Fig. 2. Convolutional neural network structure for gesture recognition.

3.3 Loss function

The training of CNNs generally comprises two stages: the forward propagation stage and the back propagation stage. In the forward propagation stage, the sample x and its label y are extracted from the sample set, and those samples are input into the CNN network. After convolution, pooling, and other operations such as layer-by-layer transfer, the category attributes of the output gesture image in softmax layer C_t is obtained. When the input passes through the convolutional layer, the output is obtained from

the activation function. In the designed network, the ReLU function with a faster convergence speed was adopted as the activation function. The mathematical expression of the ReLU function is as follows:

$$f(x) = \max(x, 0) \tag{4}$$

The cross-entropy loss function was used as the network loss function; it is expressed as

$$Loss(\theta) = -\sum_{t=1}^{N} p_t \cdot \log(\hat{p}_t) \tag{5}$$

where p_t is the probability distribution of the real class labels, $p_k = 1$, $p_t = 0(j \neq k)$, $\hat{p}_t$ is the probability distribution of the predicted labels of the network. Setting the output of the fully connected layer network to θ_t yields

$$\hat{p}_t = \text{soft}\max(\theta_t) = \frac{e^{\theta_t}}{\sum_j^N e^{\theta j}} \tag{6}$$

Back propagation primarily optimises the weight and bias terms of the entire network through the cost function and error obtained by forward propagation. Generally, the smaller the cost function, the better is the performance of the entire network. At this time, the network weights and bias terms have reached a more ideal state. Therefore, the goal of the entire network training is to reduce the cost function; this is generally realised using optimisation algorithms.

During network optimisation, the adaptive moment estimation (Adam) optimisation algorithm was used for back propagation. The Adam algorithm uses gradient first-order moment estimation and second-order moment estimation to dynamically adjust the learning rate of each parameter. Its main advantage is that after an offset correction, a certain is achieved for the learning rate, rendering the parameters more stable. In addition, the stochastic gradient descent (SGD) optimisation method [31] and the Nesterov gradient acceleration method (NAG) were investigated for comparison. A comparison of the network loss is shown in Fig. 3 based on the difference in the training time of different network layers. Therefore, a network structure with the same number of network layers was configured for the loss comparison. As

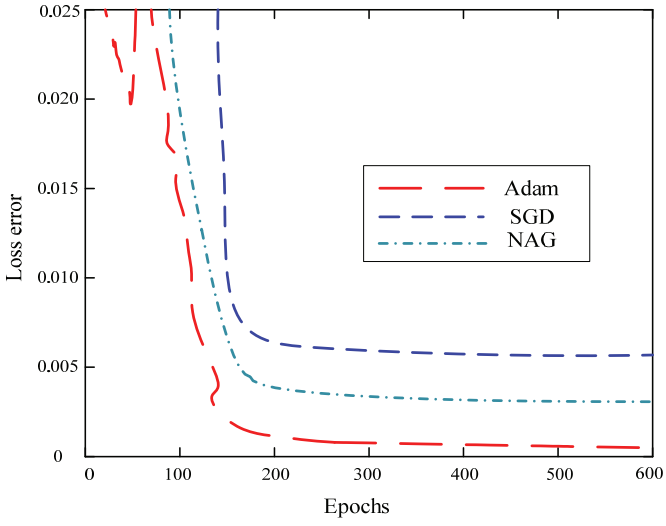

Fig. 3. Training loss of different optimization methods.

shown in Fig. 3, when the training period was short, although the overall convergence trends were similar, the Adam optimisation method yielded a smaller loss value at first; however, after training, its loss value was much smaller than those of the SGD and NAG optimisation methods, thereby proving the rationality of using the Adam optimisation algorithm combined with a cross-entropy loss function.

In summary, the gesture recognition algorithm based on the deep convolutional residual network is summarized as follows:

Algorithm 1. Gesture recognition algorithm based on deep convolution residual network

Input: training sample library $\{y_i\}_{i=1}^{N}$ form a training set, test samples $\{x_i\}_{i=1}^{N}$ Make up a test set.

Output: category attribute of gesture image C_t .

Training part:

1. Use the network training sample library $\{y_i\}_{i=1}^{N}$ shown in Fig. 1 to obtain the probability distribution of the model prediction category;
2. Calculate the loss value according to formula (5), fine-tune the learning weights by backpropagation, and iterate until the training is completed to obtain the network weights model.

Test part:

1. Input test set $\{x_i\}_{i=1}^{N}$;
2. Extract the expression features of the gesture image based on the weight model obtained by training;
3. Output the predicted category label label C_t ,using formula (6), calculate the overall recognition rate.

4. Experimental Results and Analysis

4.1 Experimental Setup

The ISOGD gesture dataset and Gesture dataset were selected as experimental samples, and the recognition rate (recognition rate) was used to evaluate the recognition performance of different networks. The ISOGD gesture dataset contained 47,933 pictures, and they were segmented into two parts. In the first part, 12,400 pictures were used for deep learning training, and the remainder was used for classification tests. A total of 16,989 Gesture datasets were selected. In the experiment, 2,280 frontal gesture images under different lighting changes were selected, of which 1,330 were randomly selected as training samples and the remaining 950 were used as test samples.

The algorithm network is a tandem network structure, and each subnetwork must perform 150 iterations of the overall sample training. In this algorithm, the learning rate decayed gradually, the initial learning rate was 0.001, the learning rate was reduced to 1/3 of the original value after every 20 iterations, and the weight decay was 0.0001. To avoid overfitting the network, a residual learning method was used. In the deep residual learning network, the first-layer subnetwork step size was set to 1 and filled with 1, whereas the last three-layer subnetwork step size was set to 2 and filled with 1 such that the learned residual features can be compared with the last three layers. The features obtained by the subnet training cannot be fused. Before the fusion, the residual features were subjected to a convolution kernel with a size of 3 and a step size of 2 to perform feature map down-sampling. In this experiment, the image was first input to the convolutional layer, and maximum pooling was performed to extract the initial features.

Subsequently, the global separation convolutional network based on deep residual learning was cascaded, and the fully connected layer was used for classification to obtain the final recognition result.

The experiment in this study was implemented in a Ubuntu 16.04.4 LTS environment based on the TensorFlow 1.7 platform, and the computer used comprised an Intel Xeon CPU E5-2630 v4 @ 2.20 GHz and two Nvidia GTX 1080 GPUs with 64 GB of memory and 12 GB of graphics card memory.

4.2 The Impact of the Number of Fully Connected Layer Neurons on the Network

Using gesture samples as input data, the network training sample library shown in Fig. 1 was used. Using the Adam optimisation method, the 500 neurons of the fully connected layer in the original network were increased to 800 and 1,000, and iterative training was performed 600 times each. The changes in the number of cost function iterations during the training process are shown in Fig. 4. The correct recognition rate of the model trained based on the number of neurons in the fully connected layer on the test set is shown in Table 1. As show in Table 1 and Fig. 4, the Adam optimisation method was used. Under the same conditions, the rate of decline of the cost function value was affected by the change in the number of neurons in the fully connected layer. When the number of neurons in the fully connected layer was 1,000, it decreased. However, after 5,000 iterations of training, the recognition rate of the test set did not improve. In fact, the recognition rate was the highest when the number of neurons was 500. It was demonstrated that increasing the number of fully connected layer neurons can improve the training speed to a certain extent, albeit not to the maximum.

Table 1. Comparison of gesture recognition rate after the number of fully connected neurons changes

Number of fully connected neurons	Test set recognition rate (%)
500	98.65
800	96.75
1,000	97.16

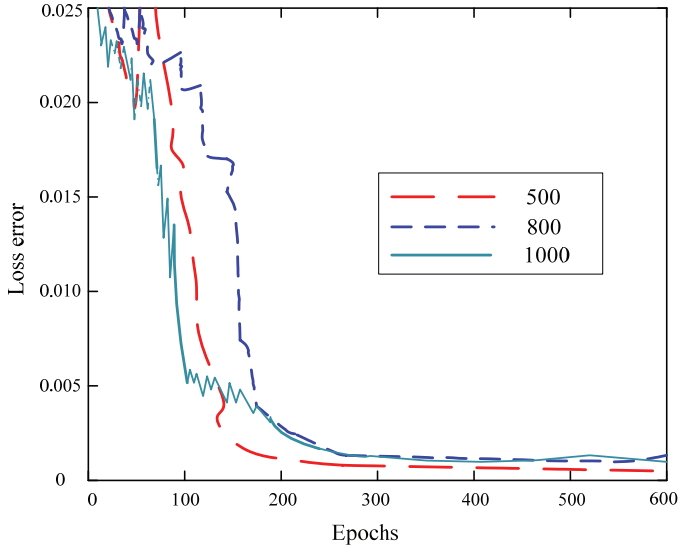

Fig. 4. Changes in cost function after changing number of neurons in fully connected layer.

4.3 Comparison with Other Methods

To verify the performance of the proposed algorithm, the proposed algorithm was compared with the algorithms in [23-25]. In each group of experiments, the final iteration result of the network was selected as the final recognition rate. As shown in Fig. 5, the recognition rate of the other algorithms fluctuated significantly with the period. Meanwhile, the proposed algorithm obtained the best recognition effect in both gesture image datasets. When the number of training iterations was small, the recognition rate of the proposed algorithm was higher than those of the other algorithms. After 50 training cycles, the recognition rate stabilized. Even though the number of training cycles was increased, the proposed algorithm still yielded a good recognition effect. In the ISOGD database, the proposed algorithm was second only to the algorithm in reference [23]. Compared with the algorithm ranked third [24], the recognition rate of the proposed algorithm was higher by 2.8%, thereby proving its better recognition performance.

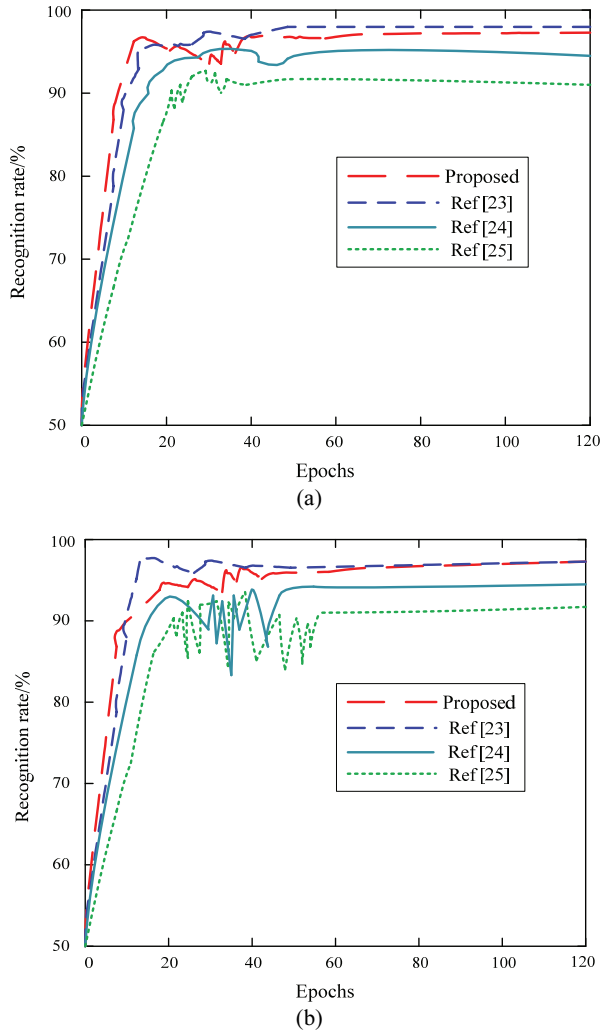

Fig. 5. Comparison of recognition rates of different algorithms in different training cycles: (a) ISOGD gesture dataset and (b) Gesture dataset.

In the Gesture dataset, the recognition rate of the algorithms from [24] and [25] fluctuated significantly. This is because the current deep learning framework is not optimised for dense connections and only relies on repeated stitching operations between feature maps. The previously extracted feature maps were spliced together and then passed on to the lower layer network. Therefore, during training, dense connections occupied a large amount of memory and resulted in significant fluctuations in the network performance. By contrast, the proposed algorithm converged at the 45th training cycle, the loss value stabilised at approximately 0.001, and the recognition rate stabilised, thereby reflecting its considerable robustness.

The test times of different recognition algorithms are shown in Fig. 6. Based on a comparative analysis, it was discovered that the algorithm in [23-25] used a more complex deep learning model that affected the test time. The algorithm in this study used a deep global separable convolution operation combined with a residual learning module to reduce the redundancy of multiplexed features and accelerate the initial feature extraction process for training.

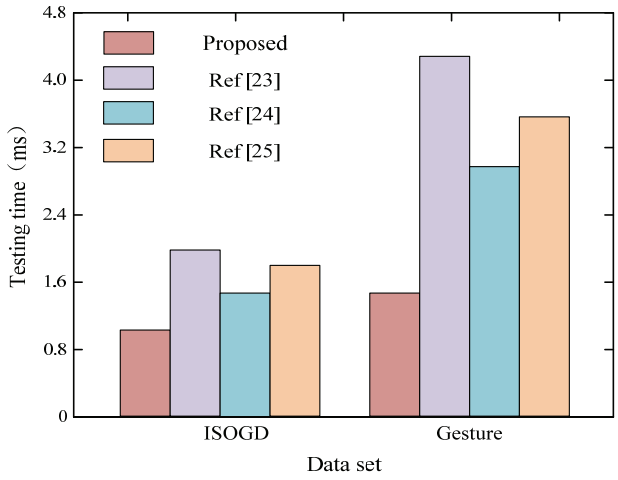

Fig. 6. The average test time of each image by several algorithms.

In the framework of [23], the DCGAN is a deep network recognition algorithm that improves recognition performance by deepening the number of CNN layers and increasing the number of model parameters. This method can improve the accuracy of label prediction based on multiscale feature information fusion; however, when the network layer is deeper, degradation problems will occur, thereby resulting in saturation or a decline in the accuracy of the training set. The feature dimension of the fully connected layer of the DCGAN is large, thereby rendering the calculation time cost much greater than those of other comparison algorithms. The proposed algorithm builds a residual learning network and extracts features with different depths of semantic information for fusion; this not only reduces the side effects caused by the increased depth of the network, but also reduces the overall number of parameters of the model and improves the overall network by combining global separable convolution operations.

With the deepening of the network structure presented in reference [24], the algorithm will increase the number of parameters in the network. Consequently, the amount of calculation and test time will increase. In real application scenarios, this will render it difficult to perform gesture recognition quickly and efficiently. The algorithm has been verified based on a significant amount of experimental results to

obtain the optimal number of residual learning module layers combined with a global separable convolution. It uses fewer convolutional network layers and fewer parameters to reduce the operation time.

The algorithm in [25] eliminates the data overfitting problem through network initialisation (ReLU and softmax) and L2 regularisation; furthermore, it uses the BN layer for feature normalisation after the fully connected layer, thereby increasing the total test time of the model.

In summary, compared with other algorithms, the algorithm proposed herein can extract more discriminative feature information to distinguish different types of face images while affording better recognition effects and a shorter test time. For example, compared with traditional deep networks, the proposed algorithm exhibits a simplified network structure and affords low time complexity. Compared with densely connected multiplexed feature information networks, the proposed algorithm is optimised for feature reuse to avoid feature redundancy. The experimental results of this study prove that the use of a residual network based on global depth separation convolution results in an efficient recognition of gesture images.

5. Conclusion

In recent years, the field of deep-learning gesture recognition has realised remarkable achievements. However, the complexity of the deep learning model has affected the real-time performance of gesture recognition and limited the application of gesture recognition algorithms in actual scenarios. Hence, a residual learning neural network based on a deep CNN was proposed in this study. Compared with traditional machine leaning methods, the proposed network is advantageous in terms of the feature extraction stage. The layered network adopts the method of sharing weights to a certain extent to overcome the problem of excessive deep network parameters, thereby reducing the possibility of gradient dispersion problems. Compared with the dense connection multiplexing feature information network, the proposed algorithm is optimised for feature multiplexing to avoid performance fluctuations caused by feature redundancy. The experimental results in the ISOGD gesture dataset and Gesture dataset proved that the proposed algorithm not only afforded a fast convergence speed, but also high accuracy.

In future studies, we plan to reconstruct the residual unit of the residual network such that the optimised network weight update is more robust. In addition, some auxiliary modules will be added during detection to improve the detection rate of the detection algorithm as well as improve the effectiveness of the gesture algorithm in practical applications.

References

[1] D. Jiang, Z. Zheng, G. Li, Y. Sun, J. Kong, G. Jiang, et al., “Gesture recognition based on binocular vision,” *Cluster Computing*, vol. 22, no. 6, pp. 13261-13271, 2019.

[2] K. Vaesen, A. Visweswaran, S. Sinha, A. Bourdoux, B. van Liempd, and P. Wambacq, “Integrated 140 GHz FMCW radar for vital sign monitoring and gesture recognition,” *Microwave Journal*, vol. 62, no. 6, pp. 50-58, 2019.

[3] X. A. Huang, Q. Wang, S. Zang, J. Wan, G. Yang, Y. Huang, and X. Ren, “Tracing the motion of finger joints for gesture recognition via sewing RGO-coated fibers onto a textile glove,” *IEEE Sensors Journal*, vol. 19, no. 20, pp. 9504-9511, 2019.

[4] J. H. Sun, T. T. Ji, S. B. Zhang, J. K. Yang, and G. R. Ji, "Research on the hand gesture recognition based on deep learning," in *Proceedings of 2018 12th International Symposium on Antennas, Propagation and EM Theory (ISAPE)*, Hangzhou, China, 2018, pp. 1-4.

[5] D. Zhu, R. Wei, W. Zhan, and Z. Hao, "Individual soldier gesture intelligent recognition system," in *Proceedings of 2019 IEEE International Conference on Power, Intelligent Computing and Systems (ICPICS)*, Shenyang, China, 2019, pp. 231-235.

[6] A. Ananthakumar, "Efficient face and gesture recognition for time sensitive application," in *Proceedings of 2018 IEEE Southwest Symposium on Image Analysis and Interpretation (SSIAI)*, Las Vegas, NV, 2018, pp. 117-120.

[7] S. Sharma and S. Jain, "A static hand gesture and face recognition system for blind people," in *Proceedings of 2019 6th International Conference on Signal Processing and Integrated Networks (SPIN)*, Noida, India, 2019, pp. 534-539.

[8] J. P. Sahoo, S. Ari, and S. K. Patra, "Hand gesture recognition using PCA based deep CNN reduced features and SVM classifier," in *Proceedings of 2019 IEEE International Symposium on Smart Electronic Systems (iSES)(Formerly iNiS)*, Rourkela, India, 2019, pp. 221-224.

[9] T. Du, X. Ren, and H. Li, "Gesture recognition method based on deep learning," in *Proceedings of 2018 33rd Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, Nanjing, China, 2018, pp. 782-787.

[10] I. Dhall, S. Vashisth, and G. Aggarwal, "Automated hand gesture recognition using a deep convolutional neural network model," in *Proceedings of 2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, Noida, India, 2020, pp. 811-816.

[11] T. H. Nascimento, F. A. A. Soares, H. A. Nascimento, M. A. Vieira, T. P. Carvalho, and W. F. de Miranda, "Netflix control method using smartwatches and continuous gesture recognition," in *Proceedings of 2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE)*, Edmonton, Canada, 2019, pp. 1-4.

[12] X. Zhang and X. Wu, "Robotic control of dynamic and static gesture recognition," in *Proceedings of 2019 2nd World Conference on Mechanical Engineering and Intelligent Manufacturing (WCMEIM)*, Shanghai, China, 2019, pp. 474-478.

[13] A. Kanawade, S. Varvadekar, D. R. Kalbande, and P. Desai, "Gesture and voice recognition in story telling application," in *Proceedings of 2018 International Conference on Smart City and Emerging Technology (ICSCET)*, Mumbai, India, 2018, pp. 1-5.

[14] K. M. Kim and J. I. Choi, "Passengers' gesture recognition model in self-driving vehicles: gesture recognition model of the passengers' obstruction of the vision of the driver," in *Proceedings of 2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS)*, Singapore, 2019, pp. 239-242.

[15] S. A. Hoque, M. S. Haq, and M. Hasanuzzaman, "Computer vision based gesture recognition for desktop object manipulation," in *Proceedings of 2018 International Conference on Innovation in Engineering and Technology (ICIET)*, Dhaka, Bangladesh, 2018, pp. 1-6.

[16] J. Xue, Y. Zong, and Z. Yang, "Gesture recognition based on improved YCBCR space and multi-feature integration," *Computer Applications and Software*, vol. 33, no. 1, pp. 151-155, 2016.

[17] L. Lu, J. Zhang, Y. Zhu, and H. Liu, "A static gesture recognition method based on data glove," *Journal of Computer-Aided Design & Computer Graphics*, vol. 27, no. 12, pp. 2411-2416, 2015.

[18] Z. Cai, S. Wu, and J. Song, "Study on Hand Gesture Recognition and Portfolio Optimization Model Based on SVM," *Journal of System Simulation*, vol. 28, no. 8, pp. 1812-1817, 2016.

[19] K. H. Park and S. W. Lee, "Movement intention decoding based on deep learning for multiuser myoelectric interfaces," in *Proceedings of 2016 4th International Winter Conference on Brain-Computer Interface (BCI)*, Gangwon, South Korea, 2016, pp. 1-2.

[20] M. Atzori, M. Cognolato, and H. Muller, "Deep learning with convolutional neural networks applied to electromyography data: a resource for the classification of movements for prosthetic hands," *Frontiers in Neurorobotics*, vol. 10, article no. 9, 2016. https://doi.org/10.3389/fnbot.2016.00009

[21] W. Geng, Y. Du, W. Jin, W. Wei, Y. Hu, and J. Li, "Gesture recognition by instantaneous surface EMG images," *Scientific Reports*, vol. 6, article no. 36571, 2016. https://doi.org/10.1038/srep36571

[22] W. Wei, Y. Wong, Y. Du, Y. Hu, M. Kankanhalli, and W. Geng, "A multi-stream convolutional neural network for sEMG-based gesture recognition in muscle-computer interface," *Pattern Recognition Letters*, vol. 119, pp. 131-138, 2019. https://doi.org/10.1016/j.patrec.2017.12.005

[23] W. Fang, Y. Ding, F. Zhang, and J. Sheng, "Gesture recognition based on CNN and DCGAN for calculation and text output," *IEEE Access*, vol. 7, pp. 28230-28237, 2019. https://doi.org/10.1109/ACCESS.2019.2901930

[24] M. Tan, J. Zhou, K. Xu, Z. Peng, and Z. Ma, "Static hand gesture recognition with electromagnetic scattered field via complex attention convolutional neural Network," *IEEE Antennas and Wireless Propagation Letters*, vol. 19, no. 4, pp. 705-709, 2020. https://doi.org/10.1109/LAWP.2020.2977995

[25] A. A. Alani, G. Cosma, A. Taherkhani, and T. M. McGinnity, "Hand gesture recognition using an adapted convolutional neural network with data augmentation," in *Proceedings of 2018 4th International Conference on Information Management (ICIM)*, Oxford, UK, 2018, pp. 5-12.

[26] Z. Zhu, J. Li, L. Zhuo, and J. Zhang, "Extreme weather recognition using a novel fine-tuning strategy and optimized GoogLeNet," in *Proceedings of 2017 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, Sydney, Australia 2017, pp. 1-7.

[27] T. Fang, "A novel computer-aided lung cancer detection method based on transfer learning from GoogLeNet and median intensity projections," in *Proceedings of 2018 IEEE International Conference on Computer and Communication Engineering Technology (CCET)*, Beijing, China, 2018, pp. 286-290.

[28] K. Zhang, M. Sun, T. X. Han, X. Yuan, L. Guo, and T. Liu, "Residual networks of residual networks: multilevel residual networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 6, pp. 1303-1314, 2018.

[29] X. Liu, J. Chen, Y. Wu, Y. Cui, Z. Ding, and S. Yang, "An optimized residual network with block-soft clustering for road extraction from remote sensing imagery," in *Proceedings of 2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, Chengdu, China, 2019, pp. 2767-2772.

[30] S. Bulusu, Q. Li, and P. K. Varshney, "On convex stochastic variance reduced gradient for adversarial machine learning," in *Proceedings of 2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Ottawa, Canada, 2019, pp. 1-5.

[31] H. T. Elshoush and E. A. Dinar, "Using adaboost and stochastic gradient descent (SGD) algorithms with R and orange software for filtering e-mail spam," in *Proceedings of 2019 11th Computer Science and Electronic Engineering (CEEC)*, Colchester, UK, 2019, pp. 41-46.

Hua Han **https://orcid.org/0000-0001-7549-6870**

She has got Master's degree of Mechatronic Engineering from Hefei University of Technology in 2011. She is a lecturer at Kaifeng University. Her research interests include intelligent control and industrial robot.