Automatic Facial Expression Recognition Based on a Deep Convolutional-Neural-Network Structure

Ke Shan, Junqi Guo[*], Wenwan You, Di Lu, Rongfang Bie
College of Information Science and Technology
Beijing Normal University
Beijing, P.R.China
guojunqi@bnu.edu.cn

***Abstract*—Facial expression recognition, which many researchers have put much effort in, is an important portion of affective computing and artificial intelligence. However, human facial expressions change so subtly that recognition accuracy of most traditional approaches largely depend on feature extraction. Meanwhile, deep learning is a hot research topic in the field of machine learning recently, which intends to simulate the organizational structure of human brain's nerve and combine low-level features to form a more abstract level. In this paper, we employ a deep convolutional neural network (CNN) to devise a facial expression recognition system, which is capable to discover deeper feature representation of facial expression to achieve automatic recognition. The proposed system is composed of the Input Module, the Pre-processing Module, the Recognition Module and the Output Module. We introduce both the Japanese Female Facial Expression Database(JAFFE) and the Extended Cohn-Kanade Dataset(CK+) to simulate and evaluate the recognition performance under the influence of different factors (network structure, learning rate and pre-processing). We also introduce a K-nearest neighbor (KNN) algorithm compared with CNN to make the results more convincing. The accuracy performance of the proposed system reaches 76.7442% and 80.303% in the JAFFE and CK+, respectively, which demonstrates feasibility and effectiveness of our system.**

Keywords—Facial Expression Recognition; Deep Learning; Convolutional Neural Network

I. Introduction

Early in the 1990s, Picard predicted that Affective Computing would be an important direction for future artificial intelligence research [1]. In 1971, the American psychologist Ekman and Friesen defined seven categories of basic facial expression, which are Happy, Sad, Angry, Fear, Surprise, Disgust and Neutral [2]. In 1991, A.Pentland and K.Mase held the first attempt to use optical flow method to determine the direction of movement of facial muscles. Then, they extracted the feature vectors to achieve four kinds of automatic expression recognition including Happy, Angry, Disgust, Surprise and got nearly 80% accuracy [3].

In 2006, Hinton and Salakhutdinov published an article in "Science" [5], opening the door to a deep learning era. Hinton suggested that the neural network with multiple hidden layers had good ability for learning characteristics. It can improve the accuracy of prediction and classification by obtaining different degrees of abstract representation of the original data. So far, the deep learning algorithm has achieved good performance in speech recognition, collaborative filtering, handwriting recognition, computer vision and many other fields [4].

The concept of Convolutional Neural Network (CNN) was presented by Yann LeCun et al. in [7] in the 1980s, where a neural network architecture was composed of two kinds of basic layers, respectively called convolutional layers (C layers) and subsampling layers (S layers). However, many years after that, there was still not a major breakthrough of CNN. One of the main reasons was that CNN could not get ideal results on large size images. But it was changed when Hinton and his students used a deeper Convolutional Neural Network to reach the optimal results in the world on ImageNet in 2012. Since then, more attention has been paid on CNN based image recognition.

In this paper, we present a method to achieve facial expression recognition based on a deep CNN. Firstly we implement face detection by using Haar-like features and histogram equalization. Then we construct a four-layer CNN architecture, including two convolutional layers and two subsampling layers (C-S-C-S). Finally, a Softmax classifier is used for multi-classification.

The structure of the paper is as follows: Section 2 introduces the whole system based on CNN, including the input module, the image pre-processing module, the recognition algorithm module and the output module. In Section 3, we simulate and evaluate the recognition performance of the proposed system under the influence of different factors such as network structure, learning rate and pre-processing. Finally, a conclusion is drawn..

II. Facial Expression Recognition System based on CNN

A. System Overview

This section starts with the overall introduction of CNN-based facial expression recognition system. System flow is showed in Fig. 1.

[*] The corresponding author

978-1-5090-5756-6/17/$31.00 ©2017 IEEE
SERA 2017, June 7-9, 2017, London, UK

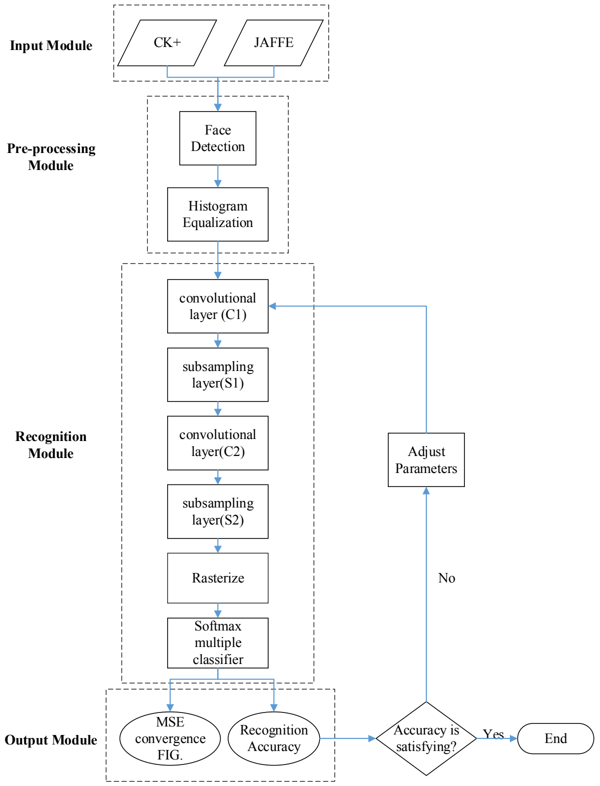


Fig. 1. System Flow Diagram.

We employ the Extended Cohn-Kanade Dataset (CK+) [8] and the Japanese Female Facial Expression Database (JAFFE) [9] for the simulation, which are both standard facial expression database categorized for 7 kinds of expressions. First of all, the Input Module obtains the input image 2D data. The Pre-processing Module includes 2 steps: face detection and histogram equalization. Thus we can get the main part of the human face and minimize the difference of lighting conditions in backgrounds. Recognition Module is based on convolutional neural network (CNN) algorithms and multiple classifiers Softmax. The Output Module shows MSE convergence figure and calculate the recognition accuracy. If the recognition accuracy does not meet the requirement, re-adjust the network parameters and begin a new round of training until the accuracy is satisfying. Details of each module are described as follow.

B. Image Pre-processing

We employ two standard facial expression databases for the simulation, which are both widely acknowledged by academia. JAFFE contains 213 images of 10 Japanese women, while CK+ covers the expression images of all races of people and has 328 pictures totally. Before the recognition, some pre-processing work need to be done firstly. In our image pre-processing procedure, we run a two-step process to reduce the interference in the original images, which are Face Detection and Histogram Equalization.

1) Face detection based on Haar-like feature

The first step of image pre-processing is face detection. In the face detection part, detection results are based on the Haar-like feature in OpenCV, which is one of the most classic features for face detection. It was originally proposed by Papageorgiou et al. [10] [11] and also known as the rectangular feature. Haar-like feature templates are divided into three categories, namely edge features, linear features and center surround features. On this basis, Haar-like feature templates Viola and Jones [12] used are shown in Fig. 2.

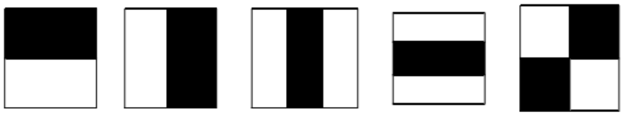
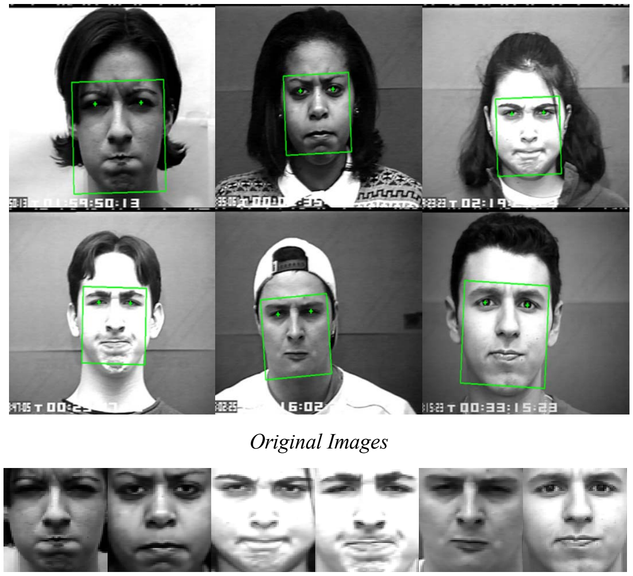


Fig. 2. Demonstration of Haar-like feature templates.

A feature template is composed of black area and white area. After placing it on a certain part of the image, we can get the feature value by the subtraction between all the pixels added within the white rectangle coverage and that within the black rectangle coverage. Accordingly, the goal of these black and white rectangles is to quantify facial features to get the distributing information of the face and finally to distinguish the non-face portion and the face portion.



Original Images

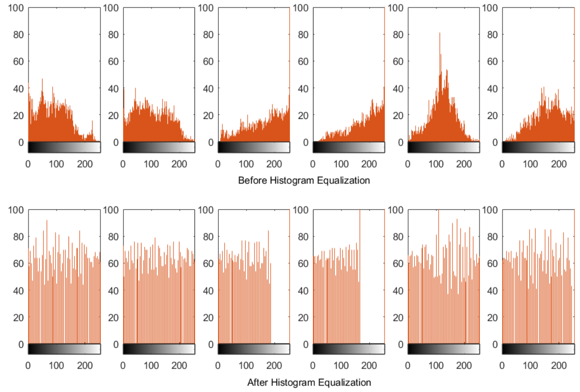
After Detection and Extraction

Fig. 3. Face detection based on Haar-like feature.

A demonstration of our detection results is showed in Fig. 3. We can see that the Haar-like feature is effective in catching the useful portion of facial expression and removing most of the meaningless background information. Therefore, it can reduce the amount of data we need to deal with, as well as effectively avoid the interference of different backgrounds and other objects in the picture on the recognition results.

2) Histogram equalization

After acquiring the very face portion of the image, other troublesome issues should also be considered. Due to the different lighting conditions when taking pictures, the portions of human face will also show in different brightness, which will inevitably cause large interference on recognition results. Thus, we decide to conduct histogram equalization(HE) before recognition. Histogram equalization is a simple but effective algorithm in image processing, which can make the gray values distribution in different images more uniform and reduce interference caused by different lighting conditions.

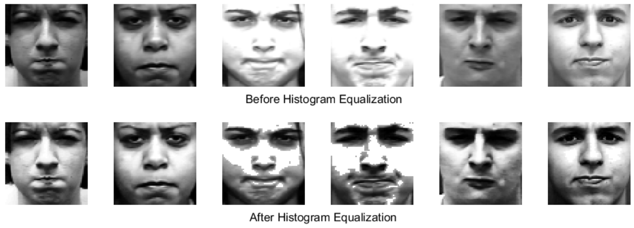


Before Histogram Equalization

After Histogram Equalization

Fig. 4. Histograms contrast before and after histogram equalization.

As is showed in Fig. 4, distributions of gray value in different picture of the same expression are very inconsistent before equalization, which causes large interference to recognition algorithm. After histogram equalization, gray value of each image uniformly covers the entire range of gradation, image contrast is improved and gray distribution of different pictures is more unified.



Before Histogram Equalization

After Histogram Equalization

Fig. 5. Contrast between the face portion before and after HE.

Fig. 5 shows more clearly that brighter face portion is optimized by histogram equalization. Thus the important features are better presented and all the images are unified as possible. We can conclude that histogram equalization is effective in reducing interference caused by different lighting conditions. The following experiments also prove it.

C. *Structure of CNN-based Recognition Algorithm*

Convolutional Neural Networks (CNN) is composed of two basic layers, respectively called convolutional layer (C layer) and subsampling layer (S layer). Different from general deep learning models, CNN can directly accept 2D images as the input data, so that it has unique advantage in the field of image recognition.

A classic CNN model is showed as Fig. 6. 2D images are directly inputted into the network, and then convoluted with several adjustable convolutional kernels to generate corresponding feature maps to form layer C1. Feature maps in layer C1 will be subsampled to reduce their size and form layer S1. Normally, the pooling size is 2×2. This procedure also repeats in layer C2 and layer S2. After extracting enough features, the two-dimensional pixels are rasterized into 1D data and inputted to the traditional neural network classifier. In practical applications, we generally use Softmax as the final multiple classifier.

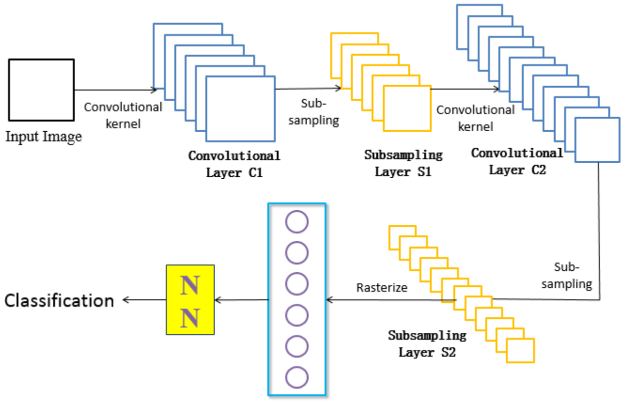


Fig. 6. Structure of CNN.

Entering a convolutional layer, the feature map of upper layer is divided into lots of local areas and convoluted respectively with trainable kernels. After the convolutions are processed by activation function, we will get new output feature maps. Let the l -th layer is a convolutional layer, the j -th output in this layer can be expressed as:

$$X_j^l = f(\sum_{i\in M_j} X_i^{l-1} * K_{ij}^l + b_j^l) \tag{1}$$

Wherein, M_j presents the local area connected by the j -th kernel, K_{ij}^l is a parameter of convolutional kernel, b_j^l is bias, $f(\bullet)$ is the Sigmoid function.

In subsampling layer, the most commonly used method is mean pooling in a 2×2 area. That is, average 4 points in the area as a new pixel value.

Parameter estimation in CNN still uses the gradient algorithm of back propagation. However, according to the characteristics of CNN, we should make some modifications in several particular steps.

Suggest that the residual error vector spread to the raster layer is d_r. Specifically, it can be written as:

$$d_r = [d_{111}, d_{112}, \ldots, d_{jmn}]^T \tag{2}$$

Because the rasterization is a 2D-to-1D transfer, residual error vector reverse pass to the subsampling layer is simply needed to re-organized from 1D to 2D matrix.

When reversing pass from S layer to C layer, different pooling method is corresponding to different process of residual error back propagation. In mean pooling, we just average the residual error in current point to 4 points of upper layer. Suggest that a residual error in a point of S layer is Δ_q . After up-sampling, error transferred to C layer can be presented as:

$$\Delta_p = upsample(\Delta_q) \qquad (3)$$

There are trainable parameters in C layer. Therefore, C layer has two tasks in back propagation: reverse residual error and update its parameters. According to the BP algorithm and consider the convolution operation, we can get the formula that update parameter θ_p in convolutional layer p is:

$$\frac{\partial E}{\partial \theta_p} = rot180((\sum X_{q'}) * rot180(\Delta p)) \qquad (4)$$

where rot180 refers to a 180-degree rotation to a matrix.

If a feature map q' in the former S layer is connected to a set C in convolutional layer p, the residual error spread to q' is:

$$\Delta_{q'} = (\sum_{p\in C} \Delta_p * rot180(\theta_p)) X_{q'} \qquad (5)$$

After all the parameters are updated, the network completes a round of training. This process should be carried out for all training samples until the whole network meet the training requirements.

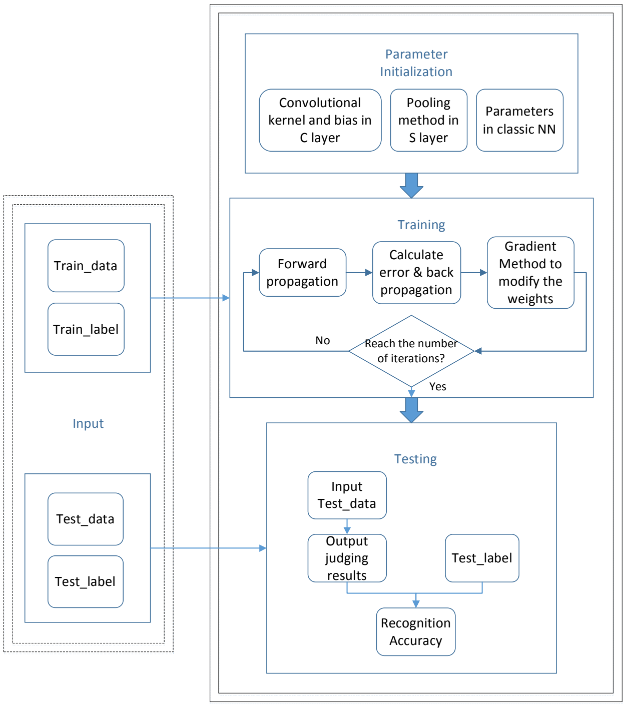


Fig. 7.Schematic of recognition algorithm.

In this paper, we employ a C-S-C-S constructed CNN. The size of convolutional kernel in C layer is 5 × 5, and the initial values are random numbers between -1 and 1. S layer do the mean pooling within the 2 × 2 area of each feature map. The logical relationship of the recognition algorithm between the various parts and functions is shown in Fig.7.

The algorithm consists of several basic parts, respectively achieve the function of data input, parameter initialization, network training and testing.

First of all, the network infrastructure is completely configured and parameters of layers are initialized by the initialization module. When initialization is finished, input the training data and training labels into the network. After the training, testing data together with testing labels are inputted into Testing Module. By comparing output judging results of testing data with the testing labels, we can finally get the recognition accuracy.

III. Performance Evaluation

A. *Performance Evaluation vs. CNN Structures*

Differences of network structures could cause great impact on the recognition performance. Generally, we need to rely on experience and continuous testing to get the best network structure for a particular classification task. For feasibility, we fix the four-layer structure as C-S-C-S and make the number of feature maps of every convolutional layer changeable. In order to better control the variables, the following results is in the case that learning rate η (0 <η≤1) equals 0.5.

TABLE I Recognition accuracy of different CNN structures in JAFFE (%)

C2 **C1**	10	12	14
4	13.9535	65.1163	69.7674
6	13.9535	**76.7442**	69.7674
8	58.1395	58.1395	67.4419

As it can be seen from Table Ⅰ, although more feature maps can extract more types of expression features theoretically, too many features will cause unnecessary interference and decrease the recognition ability of the network. Thus, proper structure of convolution layer is important for obtaining good recognition results. By the experiments, JAFFE get the highest accuracy when C1 has 6 and C2 has 12 feature maps.

TABLE II Recognition accuracy of different CNN structures in CK+ (%)

C2 **C1**	10	12	14
4	21.2121	77.2727	77.2727
6	77.2727	**78.7879**	**77.7879**
8	71.2121	72.7273	21.2121

Similarly, in CK +, we can get the best results when C1 has 6 and C2 has 12 or 14 feature maps showed in the Table Ⅱ. Based on the two databases, we finally selected 6-12 structure for network training. All of the recognition results showed below are based on this structure.

B. *Performance Evaluation vs. Learning Rates*

Learning rate η is the measure to variation of parameter updating, thus the value is controlled between 0 and 1(0 <η≤1). If η is too large, the variation of network parameters on every updating will be too sharp, which will affect the stability of the updated parameters. Even worse, it can eventually lead to nonconvergence error with the increase of training times. Meanwhile, if η is too small, the process of convergence will take a long time and consume too much calculating resources. Therefore, the value of η needs to be selected appropriately according to the actual training environment.

For better recognition performance, we run the tests on different values of η. We selected five discrete values between 0-1 and get the recognition results in both JAFFE and CK+, which are respectively displayed in the Table Ⅲ and the Table Ⅳ as below.

TABLE III Recognition accuracy on different learning rates in JAFFE (%)

η	Accuracy
0.1	69.7674
0.3	72.093
0.5	**76.7442**
0.7	74.4186
0.9	72.093

TABLE IV Recognition accuracy on different learning rates in CK+ (%)

η	Accuracy
0.1	75.7576
0.3	75.7576
0.5	78.7879
0.7	**80.303**
0.9	77.2727

The best recognition result is obtained when η is 0.5 in JAFFE, and 0.7 in CK+. Higher or lower learning rate will decrease the recognition performance of CNN.

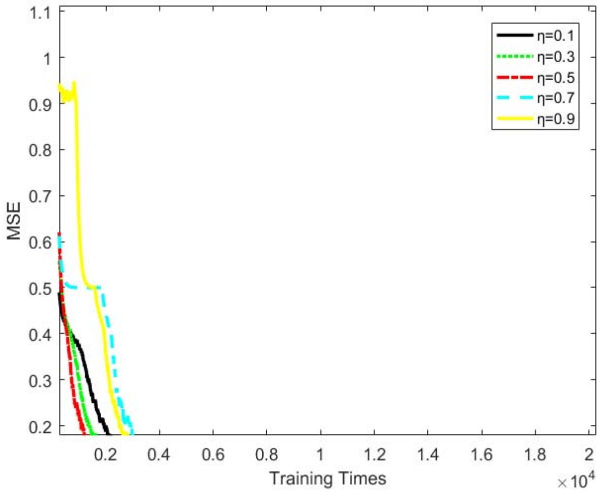


Fig. 8. Partial magnification of MSE convergence curve in CK+.

In order to further analyze the impact of different learning rates on the recognition performance, we draw the partial magnification of MSE (Mean Square Error) convergence curve of CK+ as an example.

As we can see more intuitively from Fig. 8, when η is too large (η=0.9,0.7), the sharp change of weights results in the oscillation of MSE at the beginning of training. As η decreases, the error convergence tends to be stable. The smaller η leads to the slower MSE convergence.

C. *Performance Evaluation vs. Image Pre-processing*

To reflect the effect of pre-processing, we discuss the recognition performance before and after the histogram equalization (HE). The result in JAFFE is showed as Table Ⅴ.

TABLE V Recognition accuracy before and after HE in JAFFE (%)

η HE	0.1	0.3	0.5	0.7	0.9
After	69.7674	72.093	**76.7442**	74.4186	72.093
Before	69.7674	65.1163	67.4419	51.1628	58.1395

The same evaluation in CK+ is displayed in Table Ⅵ as well:

TABLE VI Recognition accuracy before and after HE in CK+ (%)

η HE	0.1	0.3	0.5	0.7	0.9
After	75.7576	75.7576	78.7879	**80.303**	77.2727
Before	71.2121	72.7272	75.7576	77.2727	75.7576

The results show that the recognition accuracy without histogram equalization is reduced due to brightness interference, compared to the accuracy after equalization on every value of learning rate. It proves that histogram equalization does indeed improve recognition performance of the network.

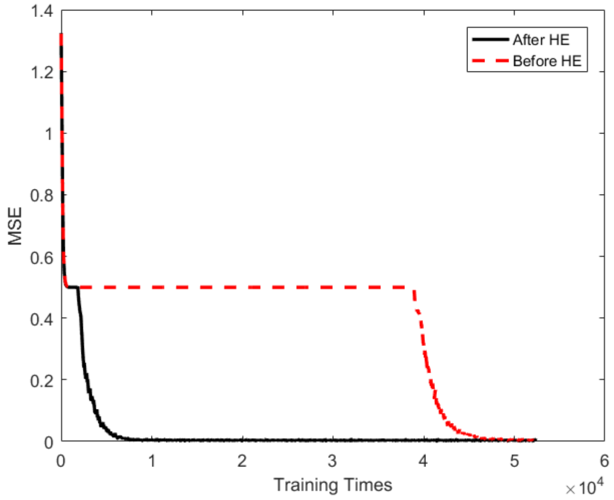


Fig. 9. MSE convergence curve before and after HE in CK+.

For deeper analysis, we draw the MSE convergence curve in the case of training process with and without histogram equalization. As a representative, Fig.9 shows the MSE convergence curve before and after histogram equalization in CK+, where η = 0.7.

Obviously, MSE converges more slowly when training the network with original images, which means that there is a lot of interference before the histogram is equalized. Much computational resources and training times are consumed to correct the interference caused by brightness differences. By the appropriate pre-process like histogram equalization, we can render the MSE converge faster and finally get the better training results.

D. Performance Comparison

For a more comprehensive display of CNN's performance on facial expression recognition task, we introduce a traditional classification method KNN (K-Nearest Neighbor) to make a comparison.

For equality, the pre-processing of images for KNN is the same as CNN's. We get the recognition results in different K values as below.

TABLE VII Recognition accuracy of KNN on different K values in JAFFE (%)

k	Accuracy
3	58.1395
5	62.7907
7	**65.1163**
9	62.7907
11	48.8372
13	53.4884

TABLE VIII Recognition accuracy of KNN on different K values in CK+ (%)

k	Accuracy
10	74.2424
15	**77.2727**
20	74.2424
25	72.7273
30	72.7273
35	71.2121

Thus, we can compare the two best recognition results of CNN and KNN, which is shown in Table Ⅸ.

TABLE IX The comparison of best recognition accuracy of CNN and KNN (%)

	CNN	KNN
JAFFE	76.7442	65.1163
CK+	80.303	77.2727

As is shown, CNN's performance is significantly better than KNN's in the task of facial expression recognition. KNN is simply based on the spatial location of known data to judge the test data, while CNN can learn deeper features of data and get more reliable recognition results. In a conclusion, the CNN is obviously more suitable for facial expression recognition.

IV. Conclusions

In this paper, we have proposed a system based on a CNN algorithm to achieve human facial expression recognition. The whole system is composed of Input Module, Pre-processing Module, Recognition Module and Output Module. First of all, we build a theoretical model of the system, and describe the details of every module, especially the CNN algorithm module. Then we introduce two classic facial expression databases JAFFE and CK+ to simulate the recognition process on MATLAB, and analyze recognition performance in different situations. A KNN algorithm is also employed to make comparison with CNN, which demonstrates that the CNN algorithm is more suitable for facial expression recognition.

References

[1] Picard R. W.. Affective computing. MIT Press.

[2] Ekman P, Friesen WV. Constants across cultures in the face and emotion[J]. Journal of personality and social psychology, 1971,17(2): 124.

[3] Mase K. Recognition of facial expression from optical flow. IEICE Trans E[J]. Ieice Transactions on Information & Systems, 1991, 74(10).

[4] X. W. Chen and X. Lin, "Big Data Deep Learning: Challenges and Perspectives," in IEEE Access, vol. 2, no. , pp. 514-525, 2014.doi: 10.1109/ACCESS.2014.2325029.

[5] Hinton G E; Salakhutdinov R R. Reducing the dimensionality of data with neural networks [J]. Science, 2006, 313:504-507. DOI: 10.1126/science.1127647.

[6] Hubel DH, Wiesel TN. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. The Journal of Physiology. 1962;160(1):106-154.2.

[7] Lecun, Y. "Generalization and Network Design Strategies." Connectionism in Perspective 1989.

[8] Lucey, P., Cohn, J. F., Kanade, T., Saragih, J., Ambadar, Z., & Matthews, I. (2010). The Extended Cohn-Kanade Dataset (CK+): A complete expression dataset for action unit and emotion-specified expression. Proceedings of the Third International Workshop on CVPR for Human Communicative Behavior Analysis (CVPR4HB 2010), San Francisco, USA, 94-101.

[9] Michael J. Lyons, Shigeru Akemastu, Miyuki Kamachi, Jiro Gyoba. Coding Facial Expressions with Gabor Wavelets, 3rd IEEE International Conference on Automatic Face and Gesture Recognition, pp. 200-205 (1998).

[10] Pageorgiou C., Oren M., Poggio T.. A general framework for object detection. International Conference on Computer Vision. 1998. 555-562.

[11] Oren M., Pageorgiou C., Ppggio T.. Example 一 based object detection in images by components. IEEE Transaction on Pattern Analysis and Machine Intelligence.2001.23(4): 349-361.

[12] Viola P., Jones M.. Rapid object detection using a boosted cascade of simple features. IEEE Conference on Computer Vision and Pattern Recognition. 2001:511-518.