Facial Expression Recognition with Small Samples Under Convolutional Neural Network

Cheng Weiyue[1], Jiahao Geng[2], and Kezheng Lin[2(✉)]

[1] Heilongjiang College of Business and Technology, Harbin, China
cheng_weiyue@sina.cn
[2] School of Computer Science and Technology, Harbin University of Science and Technology, Harbin, China
link@hrbust.edu.cn

Abstract. In order to further improve the accuracy of facial expression recognition in small samples, a small sample expression recognition method based on deep learning and fusion of different models is proposed. In this method, a single CNN model is first compared, and the relatively appropriate convolutional neural network (CNN) is selected by preserving probability of different nodes in the dropout layer. Then, the scale-invariant feature transformation (SIFT) algorithm is used to extract features. The purpose of extracting features with SIFT is to improve the performance of small data. And then, in order to reduce the error, avoid over fitting, all the model to carry on the summary, all the model of the weighted Average CNN-SIFT-AVG (Convolutional Neural Network and Scale Invariant Feature Transformation business) model. Finally, only a few sample data are used to train the model. The model has been tested on FER2013, CK+ and JAFFE datasets. Experimental results show that this model can greatly improve the accuracy of small sample facial expression recognition, and has produced excellent results in FER2013, CK+ and JAFFE dataset, with a maximum improvement of about 6% compared with other facial expression recognition methods.

Keywords: Facial expression recognition · CNN · SIFT · Small sample

1 Introduction

Facial expression recognition has always been a challenging research topic and has been widely used in detecting mental disorders and human-computer interaction [1]. It could help create more intelligent robots capable of recognizing human emotions. Many real-life applications, such as fatigue driving detection and interactive game development, also benefit from the technology. At present, various feature extraction and machine learning algorithms are used in the field of facial expression recognition. After the success of ILSVRC [2] and AlexNet [3] deep convolutional neural network model, deep learning has begun to be widely used in the field of computer vision. The challenge of facial expression recognition [4] may be one of the earliest works that put forward the deep learning method for facial expression recognition. After that, the system with the

S. Shi et al. (Eds.): 6GN 2021, LNICST 439, pp. 383–396, 2022.
https://doi.org/10.1007/978-3-031-04245-4_34

highest score in the FER challenge in 2013 is the deep convolutional neural network [5], while the best model of manual features only ranks fourth [6].With a few exceptions [7, 8], most recent studies on facial expression recognition are based on deep learning [9–14].Recent research [15] proposes to train convolutional neural network cascade to improve performance. Others [16] combine deep features with handmade features in dynamic video expression recognition.

In most cases, the training of the CNN depend on a large amount of data, however, the facial expression recognition, the data set is limited while the scale invariant feature transform [17] and other traditional local feature extraction method of accurate results than CNN, but they don't need a lot of data sets can be obtained as a result, therefore, put forward the deep learning under the fusion of different characteristics of facial expression recognition method of small sample first carries on the comparison to a single CNN model, screening of relatively suitable CNN, then SIFT local feature extracting, finally summary and all model, get a CNN-SIFT-AVG model. This method was evaluated on FER2013, CK+ and JAFFE data sets respectively.

2 Related Work

2.1 CNN

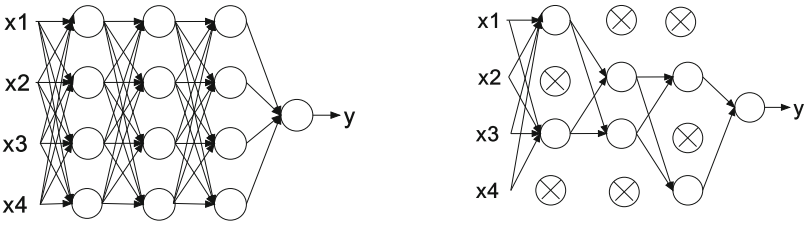
(1) Convolutional layer
Characteristics of the input data are extracted, and its interior contains multiple convolution kernels, composed of convolution kernels of each element corresponding to a weight coefficient and a bias term each neuron in the convolution layer from the previous layer close to the location of multiple neurons connected area, the size of the area depends on the size of the convolution kernels convolution kernels at work, will regularly sweep input characteristics, within the receptive field of input characteristics do matrix elements multiplication summation and superposition deviation value, calculated as shown in formula (1):

$$Z^{l+1}(i,j) = [Z^l \otimes w^i](i,j) + b = \sum_{k=1}^{K_l} \sum_{x=1}^{f} \sum_{y=1}^{f} [Z_k^l(s_0 i + x, s_0 j + y) w_k^{i+1}(x,y)] + b$$
$$(i,j) \in \{0, 1, \ldots L_{l+1}\} L_{l+1} = \frac{L_l + 2p - f}{s_0} + 1 \quad (1)$$

The summation part of the equation is equivalent to solving for a cross correlation. b for the deviation value, Z^l and Z^{l+1} represent the convolution input and output of the $l + 1$ layer, also known as the feature graph. L_{l+1} is the size of Z^{l+1}. It is assumed that the feature graph has the same length and width. Z(i, j) corresponds to the pixel of the feature graph, K is the number of channels of the feature graph, f, s_0 and p are the parameters of the convolutional layer, Corresponding to the size of convolution kernel, stride and padding layers.

(2) Dropout regularization method
Dropout layer is by iterating through each layer of the neural network node, and then based on the layer of a neural network is set up the node retention probability p, that

is, the layer of the node has the probability of p is retained, p values range between 0 and 1 by setting the probability of retention of the layer of neural network, the nerve network won't go to a certain node (because the node may have been deleted), so that the weight of each node will not too big, to alleviate neural network fitting standard network as shown in Fig. 1(a), with the comparison of dropout network as shown in Fig. 1(b).



(a) Standard neural network (b)Part of temporarily deleted neurons

Fig. 1. Comparison of standard network and dropout network

In the training model stage, each unit of the training network should add a probability process. Formula (2) without Dropout method was used. When you make a prediction, you're going to pre-multiply each of the parameters of the cell by p.

$$\mathrm{Z}_{\mathrm{i}}^{(l+1)} = w_i^{(l+1)} y^l + b_i^{l+1} \text{ and } y_i^{(l+1)} = f(z_i^{l+1}) \text{ and } \mathrm{Z}_{\mathrm{i}}^{(l+1)} = w_i^{(l+1)} y^l + b_i^{l+1} \tag{2}$$

The formula for using Dropout network is shown in formula (3).

$$\begin{aligned} r_j^{(l)} &\sim Bernoulli(p) \text{ and } \tilde{y}^{(l)} = r^{(l)} * y^{(l)} \\ z_i^{(l+1)} &= w_i^{(l+1)} \tilde{y}^l + b_i^{(l+1)} \text{ and } y_i^{(l+1)} = f(z_i^{l+1}) \end{aligned} \tag{3}$$

(3) Activation function
The ReLU function is shown in Eq. (4). When $x < 0$, as the training progresses, some of the input will fall into the hard saturation region and the corresponding weight cannot be updated. Therefore, for the hard saturation problem with $x < 0$, Leaky ReLU function is adopted, as shown in Eq. (5).

$$\mathrm{Re}LU(x) = \begin{cases} 0, x < 0 \\ x, x \geq 0 \end{cases} \tag{4}$$

$$Leaky\ \mathrm{Re}LU(x) = \begin{cases} ax, x < 0 \\ x, x \geq 0 \end{cases} \tag{5}$$

The advantage of using Leaky ReLU over ReLU is that it sets all negative values to zero, whereas Leaky ReLU gives all negative values a smaller non-zero value. This can solve the problem of neurons not learning after the ReLU function enters the negative range.

2.2 Local Feature Extraction

SIFT characteristics under the condition of rotating scale zoom brightness can keep invariance, so for each image, the SIFT extracted from face image point positioning key points, key to the adjacent pixels is used to calculate the direction and size of the grid in order to identify the main direction, set up the gradient histogram finally, SIFT descriptor by image segmentation into 4 x4 square to determine for each square of the 16 squares, using a vector to represent the length of 8By merging all the vectors, the eigenvectors with the size of 4 * 4 * 8 = 128 for each key point are obtained, and the normalization is finally done.

3 Proposed Method

3.1 CNN Network Model Structure

In this paper, a network model of the structure of the input layer is for 48 * 48 pixels gray image is the one of single channel image, so the dimension of the input image by 48 * 48 * 1 convolution filter layer is a 3 * 3, in order to keep the input and output specifications of the size is changeless, added zero padding around the border padding filter processing matrix depth and the depth of the current neural network node matrix is consistent, so it is 1 through a filter and the convolution of the input image can get a 48 characteristic figure, with 32 filter got 32 consecutive feature maps. After the convolution, the pixel position of the input image is then sliding, and the stride length is equal to 1.

Then the convolution result is integrated with the maximum pool, which is a kind of nonlinear down sampling. A 2 * 2 filter is used here for each 2 * 2 region element. At the same time, each time the maximum pooling layer is added, the number of the next convolution kernel will be doubled. So the number of convolution filters is 64,128 and 256, respectively.

After the convolutional layer output, flatten was completed and then input to the entire connection layer, which was composed of 2048 neurons. Dropout layer is introduced after each maximum pooling layer to reduce the risk of network over fitting. Finally, in the last phase of the network, you place a softmax layer with seven outputs.

According to the analysis in the previous section, in order to introduce non-linearity to CNN, Leaky ReLU was used as the activation function, and the function image was shown in Fig. 2 Specific parameters are shown in formula (6).

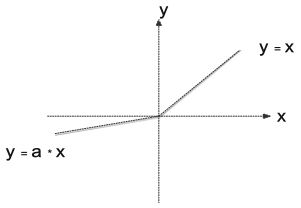


Fig. 2. Leaky ReLU functional image

$$f(x) = \begin{cases} \frac{x}{20}, x < 0 \\ x, x \geq 0 \end{cases} \tag{6}$$

Finally, the loss function is optimized by the method of classification cross entropy and the adaptive gradient optimization method Adam.

In order to achieve better classification performance, multiple CNN models were used to establish three different dropout probability models, namely, dropout1 = 0.25, dropout2 = 0, dropout2 = 0.5c2, dropout1 = 0.1, dropout2 = 0.1, dropout3 = 0.4, dropout1 = 0.1, dropout2 = 0.1, dropout3 = 0.4, dropout1 = 0.1, dropout2 = 0.1, dropout3 = 0.4, dropout1 = 0.1, dropout2 = 0.1 and dropout3 = 0.5c2 at C1, C2 and C3. The dropout probability is 0.5 to increase the diversity of the model. The overall CNN model is shown in Fig. 3.

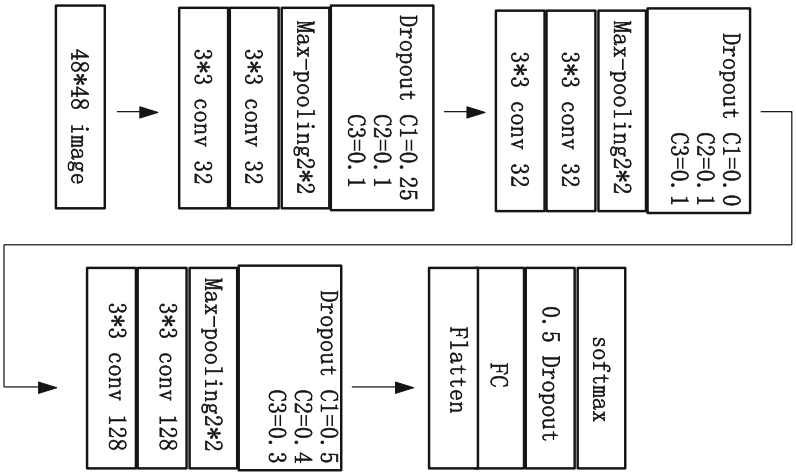


Fig. 3. CNN model adopted

3.2 Different Feature Fusion Models

In order to use the key point descriptor in SIFT in the classification, a fixed size vector is required. For this purpose, k-means is used to group descriptors into clusters. Then by calculating the number of descriptors contained in each cluster to form a bag of key points, the size of the eigenvector obtained is K.

K vector is adopted by the full connection layer of 4096 and the dropout layer. The weight of the fully connected layer is regularized with L2 norm and the value is 0.01. Three different modes S1, S2 and S3 have been tested, and the K values of each model are K1 = 256, K2 = 512 and K3 = 1024, respectively. Finally, it is merged with the C2 schema, as shown in Fig. 4.

In order to improve the accuracy of the model, the average and the CNN-ONLY,CNN-SIFT,CNN-SIFT-AVG outputs are respectively summarized, as shown in Fig. 5, where CNN-ONLY is the weighted average of C1, C2 and C3, as shown in formula (7), CNN-SIFT is the weighted average of S1, S2 and S3, as shown in formula (8). Finally, the weighted average of the six models is added to the CNN-SIFT-AVG model, as shown in formula (9).

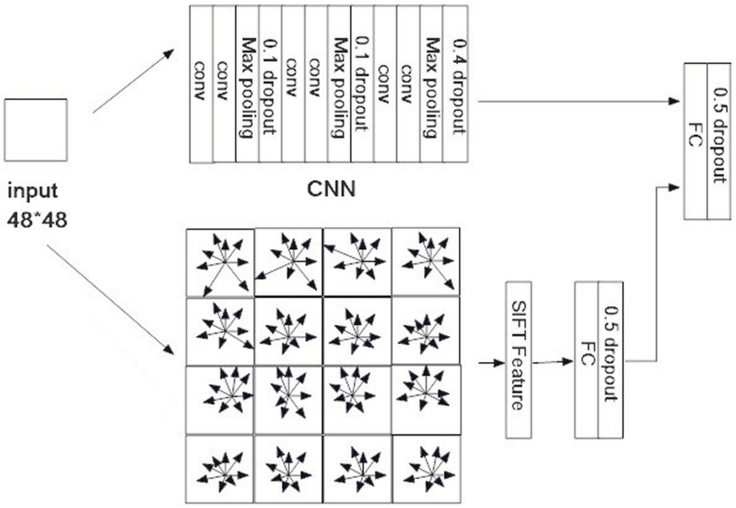


Fig. 4. CNN and SIFT model fusion

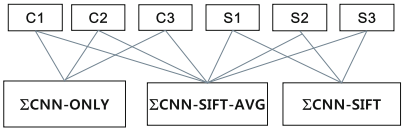


Fig. 5. Aggregating all models into CNN-ONLY, CNN-SIFT and CNN-SIFT-AVG

$$P_1(e|x) = \frac{C(e|x_{C1}) + C(e|x_{C2}) + C(e|x_{C3})}{3} \tag{7}$$

$$P_2(e|x) = \frac{S(e|x_{S1}) + S(e|x_{S2}) + S(e|x_{S3})}{3} \tag{8}$$

$$P_3(e|x) = \frac{C(e|x_{C1}) + C(e|x_{C2}) + C(e|x_{C3}) + S(e|x_{S1}) + S(e|x_{S2}) + S(e|x_{S3})}{6} \tag{9}$$

x_{Ci} represents the input dropout probability under Ci (i = 1, 2, 3) model, and $C(e|x_{\mathrm{Ci}})$ represents the probability to be judged as a certain expression under Ci mode. x_{Si} represents the input in Si mode. As can be seen above, in S1, K = 256, in S2, K = 512,

and in S3, K = 1024. $S(e|x_{Si})$ represents the probability of judging an expression in Si mode. $P_1(e|x)$ represents the probability that is determined as a certain expression under the CNN-ONLY model. $P_2(e|x)$ represents the probability of judging as a certain expression in the CNN-SIFT model. $P_3(e|x)$ represents the probability of judging as a certain expression in the CNN-SIFT-AVG model. Because each model has a softmax layer as the last layer, the output is limited to between 0 and 1.

3.3 Algorithm Steps

Based on the previous analysis, this paper takes the model as an example to elaborate the specific operation steps of this method.

Step 1: The size of facial expression images in the training samples and test samples is unified as 48 * 48, and all images are normalized into vectors with zero mean and unit variance.
Step 2: Construct a CNN network model and input training data $\{x_i\}$, respectively using C1, C2 and C3 as dropout probability values p, and the number of nodes in the hidden layer is $y_1, y_2, \ldots, y_i$. During dropout, p* i of the two nodes are set to 0.
Step 3 SIFT feature extraction is carried out for the samples in the database. According to formula (6), each key point descriptor is obtained, and vector $S = \{s_1, s_2, \ldots s_{128}\}$ with size of 128 is obtained. Then k-means algorithm is adopted to divide the cluster into $C = \{C_1, C_2, \ldots, C_j\}$. By calculating the number of descriptors contained in each cluster to form a bag of key points, the size of the feature vector obtained is K.K has three numerical modes of K1, K2 and K3 respectively.
Step 4: merge the three modes obtained in step 3 with the dropout = C2 model of CNN network, and cascade the extracted features to the full connection layer.
Step 5: the model obtained in step 1 is weighted average of the model results obtained in step 4.
Step 6: for the test sample, dropout layer could screen out some neurons, and the vector should be scaled, i.e., multiplied by $1/(1 - p)$. In other parts, steps 1–5 are successively adopted to obtain the corresponding classification and recognition accuracy of facial expressions.

4 Experiments

4.1 Experimental Environment and Data Preprocessing

In order to verify the effectiveness of the proposed method in this paper, three data sets of fer-2013, CK+ and JAFFE were used in the experiment to evaluate the performance of the proposed method. Table 1 shows the quantity distribution of each expression in FER201, CK+ and JAFFE dataset. Before the experiment, the size of the image was unified as 48 × 48, and all images were normalized into vectors with zero mean and unit variance.

Based on the tensorflow deep learning framework, this paper conducts relevant experiments on the windows10 operating system using Python3.6 programming language. Hardware platform: 7th-generation Intel core i5, Nvidia Geforce GTX 1070Ti GPU, graphics memory 8 GB.

Table 1. Expression quantity distribution of FER201, CK+ and JAFFE dataset

Experimental	FER2013	CK+	JAFFE
Angry	4953	45	30
Disgust	547	59	29
Fear	5121	25	32
Happy	8989	69	31
Normal	6198	0	31
Sad	6077	28	30
Surprise	4002	83	30

4.2 Data Set Introduction

FER2013 image size is 48 * 48 pixels, 7 expressions in the data set are marked with 0–6 Numbers, respectively, angry, disgust, fear, happy, sad, surprise. The data set contains training set and test set, in which the training set contains a total of 28,709 images and the test set contains 3,589 images. Figure 6 shows the sample image CK+ data set of 7 kinds of expressions in FER2013 database, which contains 327 expressions.

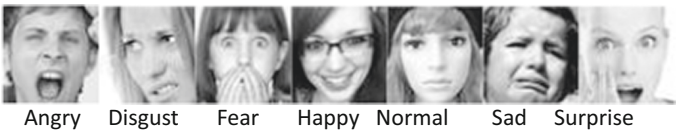


Fig. 6. Sample images of 7 expressions in FER2013 database

In order to make the experiment compatible with other experiments and FER2013 data, the contempt expression was deleted, so 309 pictures of the remaining 6 expressions were used to train the model. Figure 7 shows that all the sample networks in CK+ database only trained for 20 cycles to prevent data overfitting.

Fig. 7. Sample images of 7 expressions in CK+ database

JAFFE data set is a basic expression database specially used for expression recognition research by Japanese ATR. The database contains 213 Japanese women's face expression database, including 10 people, each woman has 3 or 4 of each expression,

and each person has 7 kinds of expressions (including Angry Normal Disgust, Fear, Happy Sad and Surprise).The JAFFE database is all positive faces, and the original image is adjusted and pruned to make the position of eyes in the database image roughly the same, the face size is basically the same, and the illumination is all positive light source, but the illumination intensity is different. Figure 8 shows the sample in JAFFE database.

Fig. 8. Sample image of seven expressions in the JAFFE database

4.3 Experimental Results and Analysis

For FER2013 dataset, all models have been trained with 28,709 samples. Of these 28,709 were used as training sets and 3,589 as test sets. Each network trains 300 epochs with a batch size of 128.

CK+ data set to use all of the 309 images for training, according to one thousand one hundred percent of cross validation method to every experiment repeat 10 times in order to prevent a fitting, all network training only 20 epoch. FER2013 CK+ and JAFFE expression distribution of the data set as shown in Table 1. JAFFE dataset using cross validation method, the images of the data set can be divided into five copies, each with one of the four as a training set, the remaining one as a test set.

In FER2013 database, in this paper, three models of the highest recognition rate has been a Happy, also known from the analysis of the above data sets, Happy features than other expressions more apparent by the experimental result shows that the integrated model has the significant improvement effect on the individual, CNN-SIFT and CNN-SIFT-AVG model is superior to CNN-ONLY model, especially the CNN-SIFT-AVG model than the other two model to improve the accurate rate of about 1%, and the use of two methods are significantly improves performance. Compared with other advanced models, the accuracy of facial expression recognition in this paper is compared, as shown in Fig. 9. It can be seen that the model in this paper is relatively stable. Although the recognition rate of some expressions is slightly lower, the overall recognition accuracy is slightly higher than that of other models, as shown in Table 2.

In the CK+ data set experiment, it can be found that, with the decrease of the number of expression samples, the performance of CNN-SIFT and CNN-SIFT-AVG is improved compared with the accuracy and performance of CNN-ONLY model, and the performance of CNN-SIFT-AVG model is relatively better. Figure 10(a) shows the confusion matrix of CK+ data set in CNN-SIFT-AVG model. It can be seen from the figure that Angry, Fear and Sad will be slightly confused and lead to errors in the recognition. Compared with other expressions, Happy is easier to be recognized.

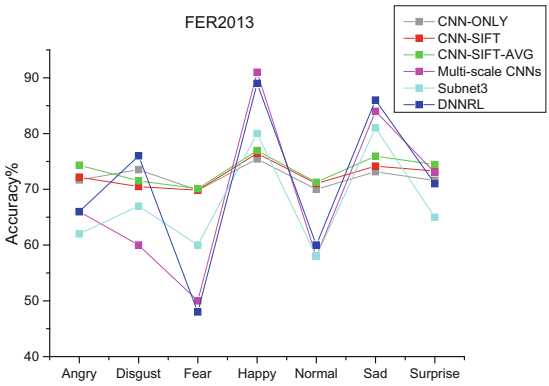


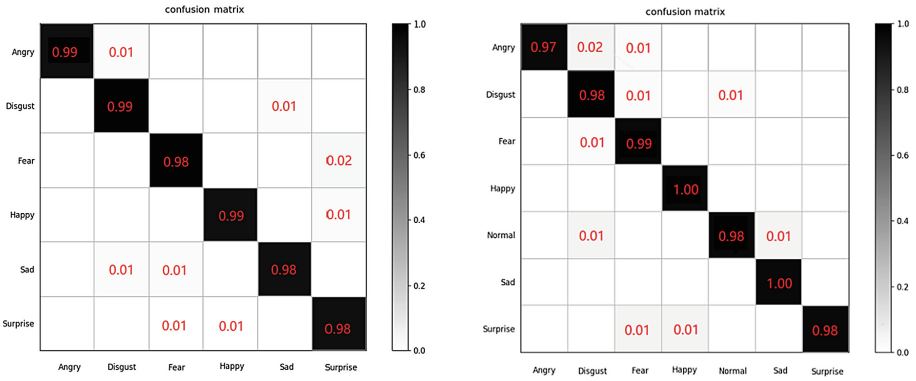
Fig. 9. Comparison diagram about the accuracy on FER2013 (left)

Table 2. Comparison of overall accuracy of FER2013 data set with other methods (%)

Model	Accuracy %
CNN-ONLY	72.17
CNN-SIFT	72.49
CNN-SIFT-AVG	73.51
Multi-scale CNNs [18]	71.80
Subnet3 [19]	62.44
DNNRL [20]	70.86

However, with the reduction of data sets, the advantages of SIFT show obvious effects. Compared with FER2013 database, in CK+ database, the recognition accuracy is greatly improved. Figure 11 shows the comparison of the accuracy of different model recognition under CK+ data set. Table 3 compares the overall accuracy of CK + data set with other methods. It can be seen from the experimental results that the model in this paper is superior to other models, whether it is the recognition accuracy of each expression or the overall recognition rate. The recognition accuracy of this paper is at least 3 percentage points higher than other methods, and the CNN-SIFT-AVG model is relatively better.

In the JAFFE data set, the model in this paper achieved a good accuracy rate, in which the Happy and Sad facial expression recognition achieved the result of no error. Figure 10(b) shows the confusion matrix of JAFFE data set on the CNN-SIFT-AVG model. Each row represents the actual category, and each column corresponds to the probability of the predicted category.



(a) Confusion matrix of CK+ data set **(b)** Confusion matrix of JAFFE data set

Fig. 10. The confusion matrix of CK+ and JAFFE data set on CNN-SIFT-AVG model

Table 3. Comparison of overall recognition accuracy of CK+ data set with other methods (%)

Model	Accuracy %
CNN-ONLY	98.33
CNN-SIFT	98.47
CNN-SIFT-AVG	99.11
Xu Linlin [21]	94.03
CNN	81.67
ZHANG Z [22]	96.30

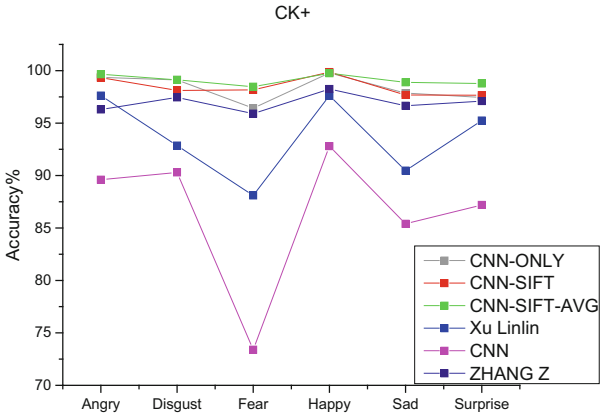


Fig. 11. Comparison of the accuracy of model recognition under CK+ dataset

Figure 12 shows the accuracy comparison of different model recognition in JAFFE data set. Table 4 compares the overall accuracy of JAFFE data set with other methods. The results show that both CNN-SIFT model and CNN-SIFT-AVG model are superior to the existing models. In particular, the recognition rate of CNN-SIFT-AVG model in JAFFE small sample database reaches 98.96%, which is 2 to 6 percentage points higher than other methods.

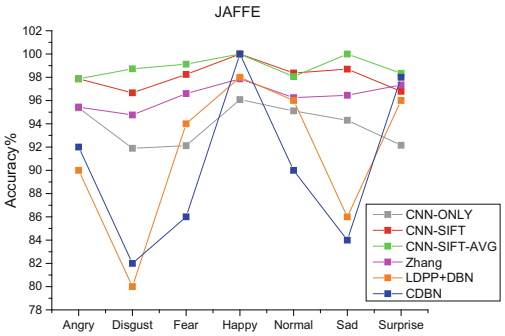


Fig. 12. Comparison of the accuracy of model recognition in JAFFE dataset (right)

Table 4. Comparison of overall recognition accuracy of JAFFE data set with other methods (%)

Model	Accuracy %
CNN-ONLY	93.86
CNN-SIFT	98.76
CNN-SIFT-AVG	98.96
ZHANG Z [22]	96.70
LDPP + DBN [23]	94.28
CDBN [24]	92.85

5 Conclusion

In this paper, an expression recognition method based on deep learning with different feature models is proposed to solve the problem of expression recognition rate in small sample data sets. This paper has shown how SIFT features and convolutional neural networks work together, and this hybrid method combines the advantages of the two methods. On the one hand, it makes full use of the advantages of SIFT that it does not need a large amount of data to extract features and improve the performance of small data. On the other hand, relatively suitable CNN is selected through comparison, and then features of SIFT are extracted and fused to summarize the model, which solves

the problem that CNN needs a lot of data training and improves the accuracy of facial expression recognition under small samples. According to the experimental results, the CNN-SIFT-AVG model in this paper has obvious advantages and plays the role of SIFT in small samples to a large extent. The smaller the data is, the more obvious the improvement effect is.

Acknowledgment. This paper was supported in part by National Natural Science Foundation of China (62071157), Natural Science Foundation of Heilongjiang Province of China (No. F2015040), the Technology Research Project of Education Center in Heilongjiang Province (11551087).

References

1. Logie, R.H., Baddeley, A.D., Woodhead, M.M.: Face recognition, pose and ecological validity. Appl. Cogn. Psychol. **1**(1), 53–69 (2015)
2. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: In Annual Conference on Neural Information Processing Systems 2012, United states, 3–6 December 2012, pp. 1106–1114. Neural Information Processing Systems Foundation (2012)
3. Russakovsky, O., Deng, J., Su, H., et al.: ImageNet large scale visual recognition challenge. Int. J. Comput. Vision **115**(3), 211–252 (2015)
4. Goodfellow, I.J., Erhan, D., Carrier, P.L., Courville, A., et al.: Challenges in representation learning: a report on three machine learning contests. Neural Netw. **64**, 59–63 (2015)
5. Mengyu, X., Zhenmin, T., Yazhou, Y., et al.: Deep learning for person reidentification using support vector machines. Adv. Multimedia **2017**, 11–18 (2017)
6. Wang, Y., Su, W.J., Liu, H.L.: Facial expression recognition based on linear discriminant locality preserving analysis algorithm. J. Inf. Comput. Sci. **9**(11), 4281–4289 (2013)
7. Owusu, E., Zhang, Y.Z.: An SVM-AdaBoost facial expression recognition system. Appl. Intell.**40**(3), 536–545 (2014)
8. Lekdioui, K., Messoussi, R.: Facial decomposition for expression recognition using texture/shape descriptors and SVM classifier. Sig. Process. Image Commun. **58**, 300–312 (2017)
9. Zhao, X.M., Shi, X.G., Zhang, S.Q.: Facial expression recognition via deep learning. IETE Tech. Rev. **32**(5), 347–355 (2014)
10. Wu, B.F., Lin, C.H.: Adaptive feature mapping for customizing deep learning based facial expression recognition model. IEEE Access **6**, 12451–12461 (2018)
11. Zeng, N.Y., Zhang, H., Song, B., et al.: Facial expression recognition via learning deep sparse autoencoders. Neurocomputing **273**, 643–649 (2018)
12. Zhang, T., Zheng, W.M., Cui, Z., et al.: A deep neural network-driven feature learning method for multi-view facial expression recognition. IEEE Trans. Multimedia **18**(12), 2528–2536 (2018)
13. Sun, X., Pan, T.: Static facial expression recognition system using ROI deep neural networks. Tien Tzu Hsueh Pao/Acta Electronica Sinica **45**(5), 1189–1197 (2017)
14. Yan, G.L., Deng, X.J., Liu, C.: Facial expression recognition model based on deep spatiotemporal convolutional neural networks. J. Central South Univ. (Sci. Technol.) **47**(7), 2311–2319 (2016)
15. Wen, G., Hou, Z., Li, H., et al.: Ensemble of deep neural networks with probability-based fusion for facial expression recognition. Cogn. Comput. **9**(5), 597–610 (2017)

16. Kaya, H., Gürpınar, F., Salah, A.A.: Video-based emotion recognition in the wild using deep transfer learning and score fusion. Image Vision Comput. **65**, 66–75 (2017)
17. Leng, X., Yang, J.H.: Research on improved SIFT algorithm. J. Chem. Pharm. Res. **6**(7), 2589–2595 (2014)
18. Zhou, S., Liang, Y., Wanf, J., et al.: Facial expression recognition based on multi-scale CNNs. In: You, Z., et al. (eds.) CCBR 2016. LNCS, vol. 9967, pp. 503–510. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46654-5_55
19. Liu, K., et al.: Facial expression recognition with CNN ensemble. In: Proceedings - 2016 International Conference on Cyberworlds, Chongqing, 28–30 September 2016, pp. 163–166. IEEE (2016)
20. Guo, Y., Tao, D., Yu, J.: Deep neural networks with relativity learning for facial expression recognition. In: 2016 IEEE International Conference on Multimedia and Expo Workshop, United states, 11–15 July 2016, pp. 166–170. IEEE (2016)
21. Xu, L.L., Zhang, S.M., Zhao, J.L.: Expression recognition algorithm for constructing parallel convolutional neural network. Chin. J. Image Graph.
22. Zhang, Z., Wang, R, Wei, M., et al.: Stacked hybrid auto-encoder facial expression recognition method. Comput. Eng. Appl. (2019)
23. Lee, H., Yan, L., Pham, P., et al.: Unsupervised feature learning for audio classification using convolutional deep belief networks. In: International Conference on Neural Information Processing Systems, pp. 1096–1104 (2009)