Proceedings of the IEEE
International Conference on Information and Automation
Ningbo, China, August 2016

Face Recognition Based on Convolutional Neural Network and Support Vector Machine[*]

Shanshan Guo, Shiyu Chen and Yanjie Li

Department of Mechatronics Engineering and Automation
Harbin Institute of Technology Shenzhen Graduate School
Shenzhen, Guangdong Province, China

Corresponding E-mail: lyj@hitsz.edu.cn

Abstract **- Face recognition is an important embodiment of human-computer interaction, which has been widely used in access control system, monitoring system and identity verification. However, since face images vary with expressions, ages, as well as poses of people and illumination conditions, the face images of the same sample might be different, which makes face recognition difficult. There are two main requirements in face recognition, the high recognition rate and less training time. In this paper, we combine Convolutional Neural Network (CNN) and Support Vector Machine (SVM) to recognize face images. CNN is used as a feature extractor to acquire remarkable features automatically. We first pre-train our CNN by ancillary data to get the updated weights, and then train the CNN by the target dataset to extract more hidden facial features. Finally we use SVM as our classifier instead of CNN to recognize all the classes. With the input of facial features extracted from CNN, SVM will recognize face images more accurately. In our experiments, some face images in the Casia-Webfaces database are used for pre-training, and FERET database is used for training and testing. The results in experiments demonstrate the efficiency with high recognition rate and less training time.**

Index Terms - Convolutional neural network, Support vector machine, Recognition rate, Training time.

I. INTRODUCTION

Face recognition, as an important field of human-computer interaction, has greatly promoted the development of artificial intelligence. Since the concept was proposed in 1960s, there have been about five methods to implement face recognition, including the geometrical characteristic method [2], the subspace analysis method [3], the elastic graph matching method [4], the hidden Markov model method [5], and the neural network method [15]. Generally, the first four methods are classified as shallow learning since they can merely make use of some basic features of images, and they all rely on artificial experience to extract sample features. The methods based on neural network are considered as deep learning since they could extract more complicate features, for example, corner point and plane features.

When referring to the face recognition based on neural network, we may commonly think about the methods such as Convolutional Neural Network (CNN) [14], Deep Belief Network (DBN) [16], and Stacked Denoising Autoencoder (SDAE) [17]. CNN can take the images as the direct input, and robust to rotation, translation and scaling deformation of images. Moreover, manually acquiring some facial features from face images is relatively difficult, while CNN could extract effective facial features automatically. Generally speaking, CNN is a good choice for face recognition. In 1988, LeCun [1] successfully processed the 2-D images with multi-layer CNN. With the development of computer hardware, in 2012, Hinton and Krizhevsky [7] applied the deep CNN to process ImageNet database, and achieved a better consequence than ever. Besides face recognition, CNN is also widely used in face verification that also had remarkable results. In face verification, Sun Y [11, 12, 13] researched and developed DeepId method based on CNN, and they had worked out three generations of DeepId until 2015. With their DeepId methods, they proved that their face verification results were superior to human eyes. CNN is not only used as classifier to carry out two or multi-class classification problems, but also as feature extractor to extract effective features. In this paper, we mainly use CNN to extract the facial features.

In this paper, we put forward a method of face recognition based on CNN and SVM. After finishing the feature extraction, we use SVM [6] as our final classifier to recognize face because of its obvious classification effect on nonlinear data. SVM was proposed firstly by Corinna and Vapnik [8] in 1995, which belongs to the supervised learning method. SVM shows many special advantages in solving small samples, nonlinear and high dimensional pattern recognition. Moreover, SVM can be applied to other machine learning problems, such as function overfitting and curse of dimensionality. In our system, SVM is to realize the further feature extraction and final classification on the basis of the facial features extracted from CNN. In this way, we might extract more features than only CNN itself in a degree. Thus, the recognition result in our system based on CNN and SVM is better than the CNN itself.

II. SYSTEM MODEL DESIGN

A. The Structure of System

[*]This work is partially supported by Shenzhen basic research programs, No. JCYJ20140417172417158, No. JCYJ20150731105106111 and No. JCYJ20140901003938996.

978-1-5090-4102-2/16/$31.00 ©2016 IEEE

In our system, all samples that we use are scaled to 32×32 pixels and preprocessed by flipping up. The Fig. 1 and the Fig. 2 respectively shows the training and testing framework. They describe the approximate recognition process of our system. We firstly use part of images of Casia-Webfaces database to train CNN, and get weights that represent facial features. Then we use these weights to initialize the layers of CNN except its last layer. For the last layer, initializing its weights by random initialization. After setting weights of all layers, we train CNN with target training dataset to extract facial features and utilize these features to train SVM. For the target testing dataset, we use the trained CNN extractor to extract features and use these features to recognize all the samples by the trained SVM classifier.

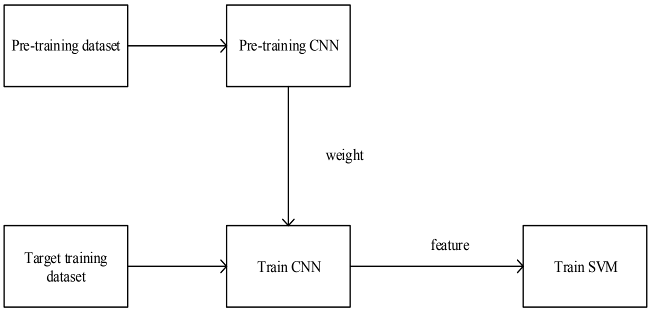


Fig. 1 The whole training system framework.

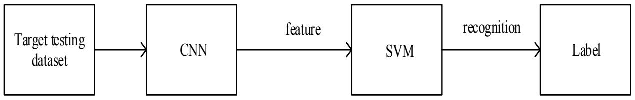


Fig. 2 The testing framework.

B. Facial Features Extraction

1) The Convolution and Pooling: As one of artificial neural network, CNN takes the image as its input, which greatly avoids massive data reconstruction and complex feature extraction in the traditional recognition algorithms. The weight sharing is the main advantage of CNN that makes it more similar to the biological neutral network. In addition, the weight sharing greatly reduces the complexity of the CNN system and decreases the number of parameters to be calculated. Weight sharing is realized through the convolution process, which is convenient for the follow-up treatment to the facial feature map. Convolution is the process that filters the image until it traverses the whole image, whose convolution formula can be described as follows:

$$y_j^l = f\left(\sum_{i \in M} x_i^{l-1} * k_{ij}^l + b_j^l\right) \quad (1)$$

$$f(x) = \max(0, x) \quad (2)$$

where x_i^{l-1} is the i th input feature map at the $l-1$th layer, y_j^l is the j th output feature map at the l layer, M is the set of of feature maps at the $l-1$ th layer. k_{ij}^l is the convolution kernel between the i th input map at the layer $l-1$ and the j th output map at the layer l. b_j^l is the bias of the j th output map at the l th layer. $f(x)$ is Rectified Linear Units function.

Pooling is another key point in CNN. Its theoretical basis is that images have the "stationarity" property, that is to say, the features that are useful in one region are also likely to be useful for other regions. Pooling is a process of subsampling that aggregates statistics of these features at various locations, which could reduce the resolution of image. Pooling enhances the robustness to the variations of images, such as rotation, noise, and distortion. It also reduces the dimensions of the output and reserves the notable features. There are two ways of pooling, the max pooling and the average pooling. In this paper, we adopt the max pooling in CNN. The maximum pool formula that we use is:

$$y_{j(\mathrm{m,n})}^{l+1} = \max_{0 \le r,k \prec} \left\{ x^{l}_{\cdot\, n\cdot s+r, n\cdot s+k)} \right\} \quad (3)$$

where $m \ge 0, n \ge 0, s \ge 0$, and $y_{j(\mathrm{m,n})}^{l+1}$ is the value of the neuron unit (m,n) that in the j th output feature map y_j^{l+1} at the $l+1$ layer, $(m \cdot s + r, n \cdot s + k)$ is neuron unit in the i th input map x_i^l at the $l+1$ layer, whose corresponding value is $x_{i(m \cdot s+r, n \cdot s+k)}^l$, $y_{j(\mathrm{m,n})}^{l+1}$ is obtained by computing the biggest value over an $s \times s$ non-overlapping local region in the input map x_i^l.

2) The Structure of CNN: In recent years, the rapid improvements of computer hardware performance well satisfy the high demand to deal with large data and deep neural network. Many researchers tend to design deeper convolutional neural networks. For example, Parkhi etc. [10] designed a very deep CNN with 37 layers to manage an especially large database, which took them many days to train. The deeper layers might lead to the lower identification speed. We hope our system can recognize samples accurately and quickly, and it can also quickly recognize a new sample. The CNN that we design has only nine layers, as Fig. 2 shows, including one input layer, three convolution layers, three pool layers, one full connected layers and one output layer. Layer C1, C2 and C3 are convolution layers, and respectively consist of 30, 60 and 80 feature maps that extract and combine some features. In three convolution layers, each neuron in each feature map connects to a 5×5 local receptive field into the previous layer. Layer S1, S2 and S3 are subsampling layers, whose number of feature maps is equal to the maps number of their previous convolution layer. In three subsampling layers, each neuron in each feature map connects to a local receptive field in 2×2 to the previous layer. F1 is the first full connects layer with 512 neurons, each of which connects to all the maps of S3. The output layer is also a full connected layer, each neuron of which connects to the first full connected layer. Output layer will finally recognize all the input images based on the exacted features.

To optimize the convolutional neural network, we use some optimization techniques. First, using the Rectified Linear

Units function $f\left(x\right)$ that describes neural signal activation well to replace the sigmoid function in our convolutional layers. Second, we apply weight penalty based on L2 regularization in our model because the smaller network weights could effectively prevents the over-fitting. Considering the high recognition rate and the less training time, we finally decide to use weight penalty in the third and the fifth layer by experiments. Third but not the last one, we use dropout that randomly selects half of the network units to be zero in our first fully connected layer. Dropout improves performance of neural networks by preventing co-adaptation of feature detectors. By using these optimization techniques, we get a better CNN with stable performance. In addition, in order to improve the generalization ability of the network, we introduce a lot of auxiliary data to pre-train the CNN.

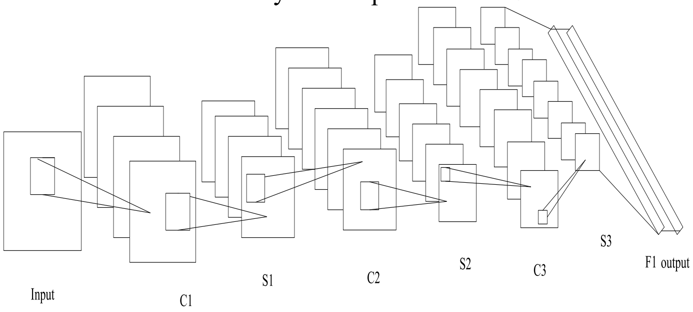


Fig. 3 Convolutional neural network model.

3) The Pre-training of CNN: We introduce the Casia-Webfaces database as our auxiliary dataset in pre-training, which consists of 500,000 images with 10,000 samples in different expression, pose, light, age and other conditions. We select part of samples with 30 images or more as our pre-training dataset. In pre-training, we only use the first 30 images of these samples, which the first 20 images are selected as training dataset and the last 10 images are selected as testing dataset. In total, we only use about 80,000 face images of Casia-Webfaces database. The pre-training dataset is inputted to the convolution neural network that we describe in the subsection B. 2), and network weights will be updated after certain number of iterations. In the process of convolution, the features extracted from the deeper layers are more effective and more complicate, and they are the abstract represent of the features from the lower layers. For example, it might extract some line features from the first convolution layer, and then it might extract some facial contour features from the latter convolution layer. As Fig. 4 shows, the weights after updating represent the facial features, such as brow, nose, eyes and mouth.

We then use our target data FERET dataset to train CNN. Considering the similarity of target data and auxiliary data, we may initialing the weights of CNN except its last layer with the weights obtained by Casia-Webfaces database. The final convolution neural network would have stronger generalization ability and faster convergence rate. In order to better evaluate the CNN after the pre-training, we compare the result of FERET dataset before and after the use of pre-training. As Fig. 5 shows, the result of using pre-training is better than not using pre-training in recognition rate and convergence speed. Pre-training improves recognition rate by 13% and saves the training time of FERET dataset.

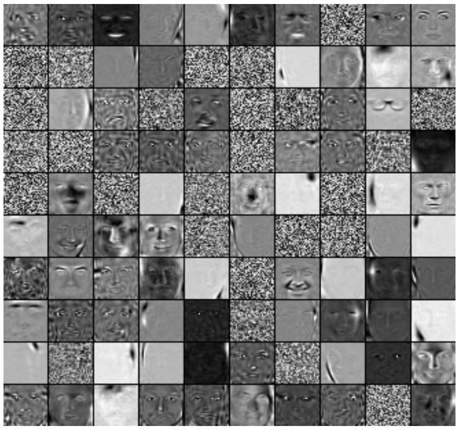


Fig. 4 Facial features extracted from pre training

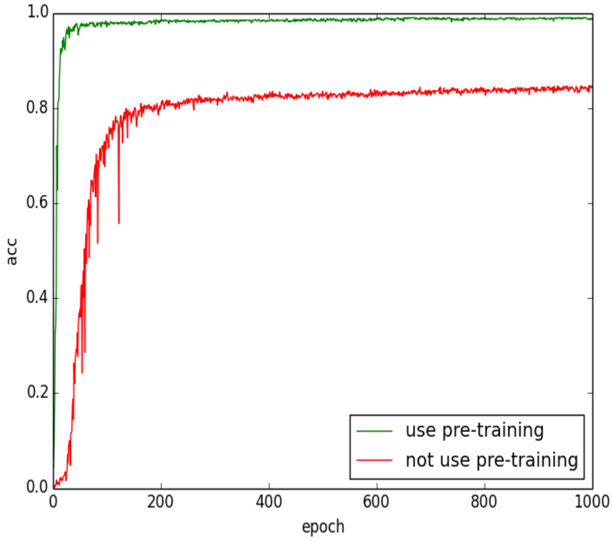


Fig.5 The comparison of using and not using the pre-training.

C. *Face Recognition Based On CNN and SVM*

In this section, we chose SVM to recognize faces as its excellent performance in solving linear inseparable problem. SVM may find an optional separating hyperplane, which makes the distance of the training samples close to it maximization. SVM is aimed to minimize empirical risk and confidence interval to achieve good statistical rules of samples and improve the generalization ability of machine learning. For linear inseparable problem, SVM maps input in low dimensions into a higher dimension feature space that makes separation easier.

In this paper, we use the Support Vector Clustering (SVC) function to find the support vector. There are two significant advantage in SVC, which can generate the cluster boundary of arbitrary shape and analyze noise data points to separate the overlapping clusters. SVC may map the data points to a high dimensional feature space by using the Gauss kernel, which could find a smallest sphere that can surround all the data points. The sphere is mapped back to the data space, which makes a set of contour lines of closed data points. These data points that are closed by the contour line belong to the same cluster.

In our system, the input of SVM is the facial features of the output layer in CNN. The input training dataset and the testing dataset of SVM are the output features of the training dataset and the testing dataset of CNN, respectively. The training label and the testing label of SVM are respectively same to the training label and the testing label of CNN. In SVC function, we adopt Radial Basis Function (RBF) as our kernel function:

$$K\left(x_i,x_j\right)=\exp\left(-\gamma \| \qquad \| \quad ,\gamma \succ \right. \tag{4}$$

where x_i, x_j are samples of input dataset, and γ is kernel parameter. The final optional problem is:

$$\min_{\alpha}\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}y_i y_j \alpha_i \alpha_j K\left(x_i,x_j\right)-\sum_{j=1}^{l}\alpha_j \tag{5}$$

$$s.t. \qquad \sum_{i=1}^{l} y_i \alpha_i = 0,$$

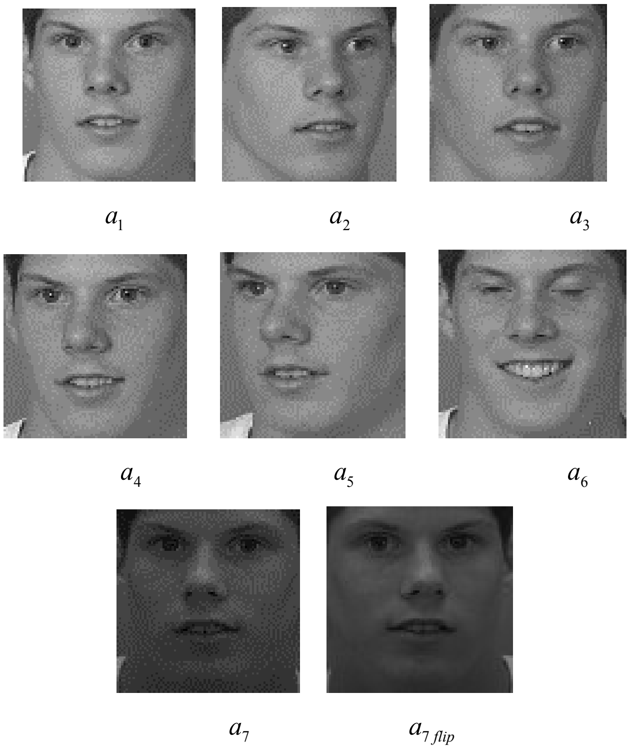
$$0 \le \alpha_i \le C, i=1,...,l$$

where y_i is the category of x_i , α_i is the coefficient of Lagrangian, C is the penalty coefficient. For the final recognition, we use the one-versus-one method in SVM. The one-versus-one is to design a SVM between any two types of samples in training dataset and adopt voting mechanism to identify which classes a new sample belongs to. We need to design $\frac{n(n-1)}{2}$ SVMs if there are n classes. The voting mechanism will test an unknown sample in the $\frac{n(n-1)}{2}$ SVMs, respectively, and determine which class the sample is more likely belong to in each SVM. Then we add one vote to the winning class and count the total number of votes for each class. When an unknown sample is classified, the final class that obtains the most votes is the class of the sample. The method one-versus-one would complete the recognition to all the classes. The fusion of CNN that can extract invariant features and SVM that can learn a better interface by the function SVC will able to improve the recognition performance.

III. Datasets

FERET dataset is used to evaluate our method, which is composed of 1400 pictures, including 200 individuals, each of which has 7 pictures under the different expression, light, and posture conditions. We mask the seven images as a_1 , a_2 , a_3 , a_4 , a_5 , a_6 and a_7 respectively, as the Fig. 6 shows. The image a_1 is frontal face image in normal condition, while the image a_6 is frontal face image with smile, the image a_7 is frontal face image in lower light and others are images with invariant pose.

In order to extract more facial features, we enlarge dataset with flipping up the original images. As Fig. 6 shows, the image $a_{7\,flip}$ is the image obtained by flipping up the image a_7 . In the original dataset division, the training dataset consists of 800 images, while the testing dataset consists of 600 images because four images of each sample are appended to training dataset and the remaining three pictures are added to testing dataset. After flipping up, the final training dataset consists of 1600 images and the testing dataset consists of 600 images since we don't process images in testing dataset.



a_1 a_2 a_3

a_4 a_5 a_6

a_7 $a_{7\,flip}$

Fig. 6 The seven masked images of a sample a_1 , a_2 , a_3 , a_4 , a_5 , a_6 and a_7 , and the filling up image $a_{7\,flip}$ of a_7 .

IV. Experiments analysis

In order to fully verify our network in more aspects, we give some definitions, such as "test samples123" and "train samples 4567". The "test samples123" means the images a_1 , a_2 , a_3 of each sample in FERET dataset that are selected as the testing dataset, while the "train samples 4567" means the images a_4 , a_5 , a_6 , a_7 of each sample in FERET dataset are selected as the training dataset. In our system, "test samples123" is equivalent to "train samples 4567", that is to say, in this two case we will finally get the same training dataset and the testing dataset. There are main three experiments in our system, including recognition performance experiment of CNN, recognition performance experiment based on CNN and SVM and the training time experiment. We mainly evaluate and analyze our system based on these experiments.

A. Recognition Performance of CNN

We randomly select six different testing datasets and their corresponding training datasets to adequately assess the performance of recognition rate. The six different testing datasets respectively are “test samples123”, “test samples127”, “test samples145”, “test samples235”, “test samples246” and “test samples135”. The final experiment results are all converged to a certain value quickly with the high recognition rate due to the pre-training. As the Fig. 7 shows, the recognition rate starts to converge from the 30th epoch, and the error rate begins to converge from the 50th epoch. The recognition rate respectively are 87.29%, 94.16%, 97.59%, 97.59%, 98.25% and 99.66% according to the order from low level to high in the six different dataset selections. The recognition error rate of most selections finally can be reduced to 0.03%, as the Fig.8 shows. The characteristic of image a_6 under dark illumination in each sample is not obvious. In experiments, the recognition rate is relatively low and the recognition error rate is relatively high if selecting the image a_6 of each sample as the testing dataset.

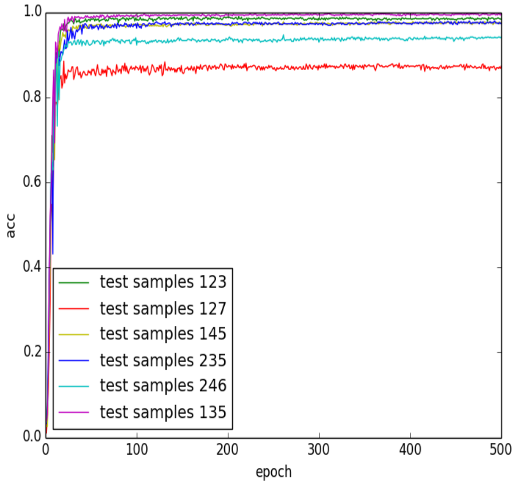


Fig. 7 The recognition rate in six selections based on CNN

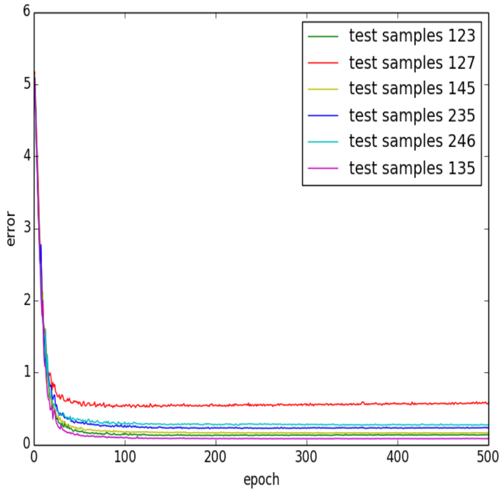


Fig. 8 The recognition error rate in six selections based on CNN

B. Recognition Performance of CNN and SVM

We use the facial features extracted from CNN to train the support vector machine, which equivalents to second extract features, hence, we might extract more facial features. It needs to adjust some parameters when the CNN and SVM are trained. In CNN, we select 128 as the final batch size and 0.0005 as the decay coefficient. In SVM, the most important parameter γ . We also select different six training datasets and testing datasets to compare the performance of CNN and CNN+SVM. Every selection can achieve the optional result by adjusting the parameter γ . We can find that the final classification of CNN+SVM is more accurate, as the Table I shows, the result of CNN+SVM is higher than that CNN is used alone.

TABLE I
THE RECOGNITION RATE BASED CNN + SVM

Dataset selection	CNN	CNN + SVM
Test Samples 126	94.50%	95.36%
Test Samples 123	98.25%	98.63%
Test Samples 135	99.66%	99.83%
Test Samples 145	97.59%	98.45%
Test Samples 235	97.20%	97.59%
Test Samples 246	94.16%	95.19%

C. Training Time

Training time is of great importance in face recognition, however, the high recognition rate is often accompanied with the more training time. In this paper, we use a large number of auxiliary data to train the network to speed up the convergence speed. To compare and analyze the performance of our algorithms in detail, we list the training time and the testing recognition rate in Table II. To compare the training time with the method [9], we also select the ORL dataset as our goal dataset. We select the first seven images of each sample as the training dataset and the remaining images as the testing dataset. The recognition rate of CNN+SVM can reach 97.5% at the 28th second while method [9] reaches 93.30% at the 343th seconds. When CNN+SVM is used in the new samples, it is also quickly to get the recognition result. Therefore, the performance of the training time of CNN+SVM is relatively excellent.

Table II
COMPARISON OF TRAINING TIME BETWEEN ACNN AND CNN+SVM

Algorithm	Training time(s)	Test recognition rate
Global expansion ACNN [9]	275	91.67%
Global + local Expansion ACNN [9]	343	93.30%
CNN + SVM	28	97.50%

V. Conclusion

In this paper, we propose an effective face recognition system based on CNN and SVM. In our system, CNN is used as a feature extractor and SVM is used as a classifier. In order to improve the performance of CNN, we use some optimization techniques to train CNN. Pre-training CNN with some ancillary data to improve the generalization ability of network, which takes much less time to extract facial features of target dataset. Taking the features of output layer as the input of SVM, which gets more accurate recognition result with its advantages in classification. The model that CNN combined with SVM spends less training time and obtains high recognition rate. The experiments based on FERET and ORL dataset verify the advantage of our system. In the future, we will try to find a balance point of recognition rate and training time based on a deeper CNN with more optimization techniques and a larger dataset.

REFERENCES

[1] Y. Lecun B. Boser, J. S. Denker, etc, “Backpropagation applied to handwritten zip code recognition,” *Neural Computation*, vol. 1, no. 4, pp. 541-551, Dec. 1989.

[2] B. Olstad and A. H. Torp, “Encoding of a priori information in active contour models,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 9, pp. 863-872, Sep. 1996.

[3] M. Turk and A. Pentland, “Eigenfaces for recognition,” *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71-86, Jan. 1991.

[4] L. Wiskott, J. M. Fellous, N. Kuiger and C. von der Malsburg, “Face recognition by elastic bunch graph matching,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 775-779, Jul. 1997.

[5] H. Othman and T. Aboulnasr, “A separable low complexity 2D HMM with application to face recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 10, pp. 1229-1238, Oct. 2003.

[6] W. Wu, “A novel solution to test face recognition methods on the training data set,” *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 8, no. 9, pp. 21-30, Sep. 2015.

[7] A. Krizhevsky, I. Sutskever, G. Hinton, “Imagenet classification with deep convolutional neural networks, “ *Proceedings of the Advances in Neural Information Processing Systems 25*, Lake Tahoe, Nevada, USA, pp.1106-1114, 2012.

[8] E. Kremic, A. Subasi, “Performance of random forest and SVM in face recognition,” *The International Arab Journal of Information Technology*, vol. 13, pp. 287-293, no. 2, Mar. 2015.

[9] Y. Zhang, D. Zhao, and J. Sun, “Adaptive convolutional neural network and it’s application in face recognition," *Neural Processing Letters*, vol. 43, pp. 389-399, Apr. 2016.

[10] O. M. Parkhi, A. Vedaldi, and A. Zisserman, “Deep face recognition,” *Proceedings of the British Machine Vision*, vol. 1, no. 3, pp. 6, 2015

[11] Y. Sun, D. Liang, and X. Wang, “Deepid3: Face recognition with very deep neural networks,” http://arxiv.org/abs/1502.00873, 2015.

[12] Y. Sun, X. Wang and X. Tang, “Deep learning face representation from predicting 10,000 Classes,” *2014 IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH, pp. 1891-1898, 2014.

[13] Y. Sun, X. Wang and X. Tang, “Deeply learned face representations are sparse, selective, and robust,” *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, 2015, pp. 2892-2900, 2015.

[14] S. Lawrence, C. L. Giles, Ah Chung Tsoi and A. D. Back, “Face recognition: a convolutional neural-network approach,” *IEEE Transactions on Neural Networks*, vol. 8, no. 1, pp. 98-113, Jan. 1997.

[15] J. E. Meng, S. W, J. L and L. T. Hock, “Face recognition with radial basis function (RBF) neural networks,” *IEEE Transactions on Neural Networks*, vol. 13, no. 3, pp. 697-710, May. 2002.

[16] K. E. Ko and K. B. Sim, “Development of a facial emotion recognition method based on combining AAM with DBN,” *2010 International Conference on*, Singapore Cyberworlds (CW), pp. 87-91, 2010.

[17] R. Xia, J. Deng, B. Schuller and Y. Liu, “Modeling gender information for emotion recognition using denoising autoencoder,” *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Florence, pp. 990-994, 2014.