A FACE RECOGNITION ALGORITHM BASED ON CONVOLUTIONAL NEURAL NETWORK

Tingting Ding¹, Yaolin Huang² *

Abstract

In recent years, face recognition has emerged as a popular technique for human recognition. The convolutional neural network (CNN) is a desirable tool for face recognition, due to its accuracy in feature extraction and simplicity in recognition. However, the traditional CNN has several defects, such as high computing load and lack of universality. To overcome these defects, this paper puts forward a novel CNN algorithm for face recognition. Unlike the traditional CNN with a fixed structure, our CNN automatically expands the number of initial neurons based on the pre-set training errors. In addition, the branch structure, network expansion interval and spreading factor were all optimized in the proposed CNN. To verify its performance, the proposed CNN algorithm for face recognition was compared with the traditional CNN through experiments on the ORL face database. The results show that our CNN algorithm outperformed the traditional algorithm in recognition accuracy, and strikes a perfect balance between training time and test error. The research results shed important new light on the application of deep learning in face recognition.

Key words: Convolutional Neural Network (CNN), Face Recognition, Training Error, Universality.

INTRODUCTION

The Internet technology has ushered people in a new era of face recognition in biometrics (Phillips, Moon, Rizvi et al., 2000). Face recognition is intuitive and simple for identity authentication by comparing facial features (Zhang & Xie, 2012), so widely used in personal identification, public security systems, government sector and business domains, etc. (Wiskott, Fellous, Kruger et al., 2009; Georgiades, Belhumeur, & Kriegman, 2002), for example, it has been integrated with mobile payment and video surveillance, etc. for human face recognition (Sim, Baker, & Bsat, 2003; Hager & Zhang, 2005). In 1888, Galton, et al. opened the door to studying face recognition. However, it was not until 1965 that the technology aroused people’s wide concern and gained a rapid development (Dong, Liu, & He, Y. 2015). Before the 1990s, face recognition was studied under laboratory conditions since the recognition rate could be easily destroyed by complex backgrounds in outdoor environment, such as illumination variation, expression changes, etc. (Dong, Liu, Xu et al., 2015; Sugiki, Narushima, & Yabe, 2012). As computer technology develops in super express, the face recognition technology has also made a dramatic leap (Zhang & Wang, 2012). Artificial neural network is used to learn and recognize human faces, extract some facial features that are highly complex and even recessive to be uneasily predictable by manual works (Li & Qu, 2014). The CNN developed based on artificial neural networks are trained using counter propagation to reduce the data processing workload by normalizing the original image (Dong, Liu, He et al., 2016). The CNN extracts the facial features and classifies these using a
classifier for recognition and integration, thus making the algorithm simpler (Yu, Zhao, & Wei, 2007). However, due to its strong fitting capacity, the CNN is prone to such phenomenon that the effect can be improved in training but impaired in practice (Liu & Wang, 2011; Cetisli & Edizkan, 2015). Aiming at the structural defects of CNN, we propose a neural network structure that can automatically expand the number of initial neurons based on preset training errors. Compared with the traditional neural network with fixed structure on the ORL face database by the experiment, it seems more superior.

**CNN**

The CNN is an artificial neural network designed based on the working principle of visual nerves, capable of carrying out deep learning. It can effectively blend the facial features extracted with the classification in the recognition process.

**Network structure**

The CNN includes input, output and hidden layers. The hidden layer also consists of single or multiple feature extraction sublayers. The original 2D image is input for feature extraction layer to extract the facial features. After that, the feature extraction and the output layers are linked with the fully connected layer. That output layer outputs the classified results, see Fig. 1 for its typical structure. Among them, the feature extraction layer includes convolution layer used for extracting facial features and sub-sampling layer for reducing the operands.

**Figure 1. A schematic diagram of self convolution neural network structure**

**Initial parameters**

To train the CNN, it is required to set the initial value of the network, and then fine-tune it to obtain the best network structure. The initial value of CNN generally succumbs to the uniformly distributed random number of formula (1). In CNN training, the appropriate parameters are obtained by learning the weights and bias terms between the neurons in the convolutional layer.

\[
W^{(1)} \sim U \left( -\frac{\sqrt{n}}{\sqrt{p^{(l)}+p^{(l-1)}}, \frac{\sqrt{n}}{\sqrt{p^{(l)}+p^{(l-1)}}}} \right) \quad (1)
\]

where, U represents the distribution function; \(p^{(l)}\) represents the number of neurons in the layer \(l\).

**Network training**

The CNN analyzes classification input data and obtains the mapping relationship between output and input data, which serves as a basis for determining new data types. The CNN propagation operation process is given as follows: First, the random value generated by formula (1) is used to initialize \(w\) and \(b\); then, the preprocessed images are input and output after the operation; if the output results are in line with expectations, eliminate the operation and reserve the parameters; If not, conduct counter-propagation until the training is satisfactory. The operation process is shown in Fig. 2.

**Figure 2. Operation process of convolution neural network**

![Operation process of convolution neural network](image-url)
EXPERIMENT DESIGN

ORL face database
ORL face database consists of 40 individuals, each of whom has 10 grayscale images with different expressions and postures. See Fig. 3 for partial images from the face database. In the experiment, 8 images for each person are randomly chosen for training.

CNN with fixed structure
The CNN consists of input layer, feature extraction layer, fully connected layer, and output layer. CNN designed herein is shown in Fig. 4, where C is the convolutional layer and S is the subsampling layer.

As the facial image is larger, the input layer is set to 32×32; the convolution layer uses 6 neurons, each gets a 28×28 feature map; after that, zoom in the images by the filter matrix to 14×14, and then 12 convolution kernels are connected with upper output data to obtain 12 12×12 feature maps, and 12 6×6 images from the 2×2 sub-sampling layer. After the image features are extracted, 84 eigenvectors are output from the fully connected layer. The position corresponding to the largest component in the eigenvector is the classification result.

Self-constructed CNN

Network structure
In general, as the number of neurons increases, the neural network can be improved in the term of performance, but blindly expanding the network is highly time-consuming. In order to better balance the training time and improve the recognition rate, an automatically constructed CNN algorithm is designed hereof, see Fig. 5 for its structure map. Among them, the neurons in the convolution layer C1 and the sub-sampling layer S1 are all 1, and the neurons in the convolution layer C2, the sub-sampling layer S2, and the output layer A are all P.

Determine whether the neural network converges by the formula (2).
\[ f(w, b)_{i+G} - f(w, b)_i \leq T \]  

(2)

where, \( f(w, b) \) represents the function at the time of the iteration \((i + G)\); \( f(w, b)_i \) represents the function at the iteration \(i\); \( G \) represents the whether the convergence occurs is judged once at the iteration of \( G \) iterations; \( T \) represents the threshold of the system function, generally set to 0.1. The working principle is such that, after the computation, if the formula (2) is met, the network converges, and no new branch network structure needs to be extended; otherwise, one is needed until the calculation result satisfies the formula (2).

**Algorithm code**

The algorithm of self-constructed CNN is given as follows:

```
Initialize network structure:
for i=1:M
  Train the weight:
  if mod(i,G) = 0 & \(\beta M \leq \beta M & f(w, b)_i - f(w, b)_{i-G} \leq T \)
    Store the network structure and parameters
    Expand one branch network structure unit,
    and initialize the new branch,
  end if
end for
The training winds up.
```

Where, \( i(1 \leq M) \) represents the number of iterations; \( \alpha \) is the target error rate; \( G \) represents the expansion interval, and \( \beta \) is the expansion factor, indicating that the network no longer expands when the number of iterations reaches \( \beta \) times \( M \).

**EXPERIMENTAL RESULTS**

**Validation**

A common CNN structure with two feature extraction layers is initialized. Training starts from the structure where only one neuron is available in each layer, and the new neurons are gradually added by using error back propagation until the convergence conditions of formula (2) is met. The convolutional layers \( C_1 \) and \( C_2 \) take 5x5, 3x3 respectively, and the subsampling layer is 2x2, and the images are normalized and never input until 32x32. Self-constructed CNN is compared with the traditional network structure based on ORL face database.

8 images randomly chosen for each person from the ORL face database are trained. The self-constructed CNN (represented by “SANN”) and traditional neural network with fixed structure are compared by the experiment to obtain the training error, as shown in Fig. 6. Since the initial values of the network parameters are not fixed, the final structure of the CNN is different. Two 6-6-6-6 and 8-8-8-8 structures are available after running several times. These numbers represent the neural networks of the two feature extraction layers.

As shown in Fig. 6, the training error of self-constructed CNN training error is higher than the traditional CNN with fixed structure, while the 8-8-8-8 structure can even reach zero. In addition, self-constructed CNN also have a high number of convergences.

**Figure 6. Contrast diagram of training error on face Library**

However, by combining the training time factors (see Table 1), the CNN with 8-8-8-8 structure has the highest training time, twice as high as the self-convolutional CNN, the test error is relatively high. Judging from this, the training time and test error reflect the fact that self-constructed CNN still has a high applicability.

**Table 1. Performance comparison between adaptive neural network and fixed structure neural network in ORL face library**

| Network structure | Training time(s) | Training error (%) | Testing error (%) |
|-------------------|-----------------|--------------------|------------------|
| Self-adaption     | 341             | 1.57               | 10.1             |
| 6-6-6-6          | 401             | 1.03               | 12.7             |
| 8-8-8-8          | 658             | 0                  | 12.3             |

**Self construction rate of self-constructed CNN**

8 images are randomly chosen from the ORL
face database to investigate how the branch network structure unit, the extension interval $G$, and the network expansion factor $\beta$ affect the performance of the self-constructed CNN. To enhance the confidence level of results, each experiment is simulated 25 times, and a mean value is taken.

**Effect of branch network structure**

In order to study what’s the effect that the branch network structure has on the self-constructed CNN, suppose $P$ is 1, 2, 4 to construct a self-constructed CNN structure. The training error is shown in Fig. 7.

As shown in Fig. 7, the training error is minimized when $P=2$. Therefore, the self-constructed CNN can be optimized by setting the branch structure.

**Figure 7. Comparison of training errors of different branch network structures**

![Training error comparison of different branch network structures](image)

**Table 2. Test error comparison of different branch network structure units**

| Branch network | Training time(s) | Training error (%) | Testing error (%) |
|----------------|------------------|--------------------|-------------------|
| 1-1-1-1        | 338              | 4.52               | 15.01             |
| 1-1-2-2        | 437              | 1.09               | 8.89              |
| 1-1-4-4        | 887              | 2.13               | 10.31             |

As shown in Table 2, the training time increases as the neurons in the second feature extraction layer heap up. When $P=2$, the training time, error, and test error are the lowest. In the ORL face database, therefore, the self-constructed CNN has optimal branch structure unit when $P=2$, i.e. the 1-1-2-2 structure.

**Effect of network expansion interval on performance of self-constructed CNN**

If each network iteration needs to calculate the training error difference, it will lead to a huge burden on the calculation. Therefore, when the network expansion interval $G$ is used to express the error difference is calculated once after the $C$ iterations. Set $G$ to 5, 10, 15, and 20, respectively, for training. The training error curve and performance are shown in Fig. 8.

As shown above, the network training error is the minimum when the convergence is judged once every 5 iterations, so that the network expansion interval will affect the convergence of the self-constructed CNN.

When $G=5$, the training and test errors are the minimum, but the training time is the longest. Therefore, it is required to balance the training time and test error to select an appropriate network expansion interval.

**Figure 8. Comparison of training errors of different branch network structures**

![Training error comparison between different network expansion intervals](image)

**Table 3. G test error comparison between different network expansion intervals**

| $G$  | Training time(s) | Training error (%) | Testing error (%) |
|------|------------------|--------------------|-------------------|
| 5    | 578              | 2.39               | 8.95              |
| 10   | 339              | 4.28               | 14.32             |
| 15   | 209              | 27.98              | 30.41             |
| 20   | 136              | 14.91              | 28.32             |

**Effect of network expansion factor**

The network expansion factor $\beta$ is to prevent the network from being converged and increasing the branch network structure when iterations maximize, because this will lead to the
network not being fully trained. $\beta$ is set to 0.2, 0.4, 0.6, and 0.8, respectively, to find out what’s the effect that the network expansion factor plays on self-constructed CNN. G takes 5, and the training result is shown in Fig. 9 below.

As can be seen from Fig. 9, when $\beta = 0.2$, the iterations exceed 20 and the network no longer adds the branches. Therefore, improper $\beta$ can lead to non-convergence of self-constructed CNN training.

When $\beta$ takes different values, the comparison table (Table 4) between training and test errors as a function of training time can be available: as $\beta$ increases, the training time extends, and the training and test errors are the minimum when $\beta=0.8$, but the training time is the longest.

![Figure 9. Different network expansion factor beta training error contrast diagram](image)

*Figure 9. Different network expansion factor beta training error contrast diagram*

| $\beta$ | Training time(s) | Training error (%) | Testing error (%) |
|---------|------------------|--------------------|-------------------|
| 0.2     | 79               | 33.1               | 40.71             |
| 0.4     | 118              | 7.80               | 15.79             |
| 0.6     | 197              | 5.97               | 21.38             |
| 0.8     | 234              | 3.41               | 12.56             |

**CONCLUSIONS**

CNN can effectively extract facial features of humans. As the traditional CNN has some gaps such as heavy workload, etc., we propose the self-constructed CNN algorithm which, in conjunction with the fixed neural network structure, is applied in ORL face database for comparison and recognition. In parallel time, it is investigated that neural network structure, network expansion interval, and spreading factor all have an effect on self-constructed CNN. We bear out the following conclusions hereof:

1. The self-constructed CNN can achieve a better equilibrium between training time and test error than the fixed structure neural network.
2. When the number of neurons in the convolutional layer, subsampling layer, and output layer, P, equals to 2, the training error is minimized. The self-constructed CNN can optimize the algorithm by setting the branch structure.
3. When the extension interval $G=5$, the training and test errors reach the minimum, but the training time is the longest. The training time and test error must be weighed to select the appropriate network extension interval.
4. As the network expansion factor $\beta$ increases, the training time extends. When $\beta = 0.8$, the training and test errors are the minimum, but the training time is the longest.

**Acknowledgements**

This paper was supported by the “Research on the High Integration Of Artificial Intelligence Technology And Innovation And Entrepreneurship Education of college students” (No. GH180067) in Jilin province under the “13th five-year plan”.

**REFERENCES**

Cetişli, B., & Edizkan, R. (2015). Use of wavelet-based two-dimensional scaling moments and structural features in cascade neuro-fuzzy classifiers for handwritten digit recognition. *Neural Computing and Applications*, 26(3), 613-624.

Dong, X., Liu, H., & He, Y. (2015). A self-adjusting conjugate gradient method with sufficient descent condition and conjugacy condition. *Journal of Optimization Theory and Applications*, 165(1), 225-241.

Dong, X. L., Liu, H., Xu, Y. L., & Yang, X. M. (2015). Some nonlinear conjugate gradient methods with sufficient descent condition and global convergence. *Optimization Letters*, 9(7), 1421-1432.

Dong, X., Liu, H., He, Y., Babaie-Kafaki, S., & Ghanbari,
R. (2016). A New Three-Term Conjugate Gradient Method with Descent Direction for Unconstrained Optimization. *Mathematical Modelling and Analysis*, 21(3), 399-411.

Georghiades, A. S., Belhumeur, P. N., & Kriegman, D. J. (2001). From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (6), 643-660.

Hager, W. W., & Zhang, H. (2005). A new conjugate gradient method with guaranteed descent and an efficient line search. *SIAM Journal on Optimization*, 16(1), 170-192.

Li, M., & Qu, A. (2014). Some sufficient descent conjugate gradient methods and their global convergence. *Computational and Applied Mathematics*, 33(2), 333-347.

Liu, J., & Wang, S. (2011). Modified nonlinear conjugate gradient method with sufficient descent condition for unconstrained optimization. *Journal of Inequalities and Applications*, 2011(1), 57-68.

Phillips, P. J., Moon, H., Rizvi, S. A., & Rauss, P. J. (2000). The FERET evaluation methodology for face-recognition algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10), 1090-1104.

Sim, T., Baker, S., & Bsat, M. (2003). The CMU pose, illumination, and expression database. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 25(12), 1615-1618.

Sugiki, K., Narushima, Y., & Yabe, H. (2012). Globally convergent three-term conjugate gradient methods that use secant conditions and generate descent search directions for unconstrained optimization. *Journal of Optimization Theory and Applications*, 153(3), 733-757.

Wiskott, L., Fellous, J. M., Kruger, N., & von der Malsburg, C. (1999). Face recognition by elastic bunch graph matching. *Intelligent Biometric Techniques in Fingerprint and Face Recognition*, 11(5), 355-396.

Yu, G., Zhao, Y., & Wei, Z. (2007). A descent nonlinear conjugate gradient method for large-scale unconstrained optimization. *Applied Mathematics and Computation*, 187(2), 636-643.

Zhang, J., & Xie, X. (2012). A study on the effective approach to illumination-invariant face recognition based on a single image. In *Chinese Conference on Biometric Recognition*, 33-41.

Zhang, Y., & Wang, K. (2012). A new general form of conjugate gradient methods with guaranteed descent and strong global convergence properties. *Numerical Algorithms*, 60(1), 135-152.