Journal of Physics: Conference Series

PAPER • **OPEN ACCESS**

A Face Recognition Method Based on CNN

A Face Recognition Method Based on CNN

Zhiming Xie[1, 2, 3], Junjie Li[1, 2, 3] and Hui Shi[1, 2, 3]

[1] Department of Information Engineering, Shanwei Polytechnic, Shanwei 516600, Guangdong, China

[2] The Research Institute of Cloud Computing and Data Center Project Design, Shanwei Innovation Industrial Design & Research Institute, Shanwei 516600, Guangdong, China

[3] The Professional Committee Cloud Computing and Big Data in Higher Vocational College, Higher Education Institute in Guangdong Province, Shanwei 516600, Guangdong, China

106710856@qq.com

Abstract. The traditional face recognition technology is more complicated for the extraction of facial image features and the selection of classifiers, and the recognition rate is not high. With the continuous maturity of the convolutional neural network from handwritten digit recognition to face recognition, A face recognition algorithm that tests CNN using the Python+Keras framework. The method mainly involves two aspects. One is to observe the influence on the network by changing the number of neurons in the hidden layer; the other is to observe the influence on the network by changing the number of feature maps of the convolutional layer 1 and the convolutional layer 2. The best CNN model is 36-76-1024 through multiple sets of experimental tests. The model can automatically extract facial image features and classify them. Using adam optimizer and softmax classifier for face recognition can make training faster convergence and more. Effectively improve accuracy and use the Dropout method to avoid overfitting. The experimental results show that the recognition rate of the CNN model on the olivettifaces face database is 97.5%. When the optimal CNN model is used, the average recognition rate is close to 100%, which verifies the validity and accuracy of the algorithm and model.

1. Introduction

Face recognition refers to a biometric feature that uses a computer to automatically identify and analyze facial visual features. It belongs to the category of computer vision [1] and is the field of artificial intelligence such as deep learning and pattern recognition in recent years. One of the research hot spots. Especially in recent years, with the introduction of deep convolutional neural networks, the accuracy of face recognition has been greatly improved and is widely used in access control security, attendance, candidate identification, character recognition, face payment and other fields [2].

At present, the commonly used face recognition methods [3-4] are mainly based on template matching [5], geometric features based [6], algebraic features based [3] (the typical representative mainly includes HMM, PCA, LDA) Based on methods such as artificial neural network [7] (which is representative of BP algorithm). However, with deep learning, it has become a research hotspot of machine learning, especially the development of the Convolutional Neural Network (CNN) has made the accuracy of face recognition further improved. CNN is a variant of MLP inspired by the biological vision to simplify the preprocessing operation. It belongs to the forward feedback neural network. The feature extraction method is to extract features by layer-by-layer convolution and then multi-layer nonlinearity. The mapping enables the network to automatically learn from the training samples that have not been specially pre-processed to form feature extractors and classifiers suitable for the recognition task. Chen Yaodan et al. had mentioned that the use of CNN for face recognition can effectively reduce the requirements for training samples, and the more network layers are learned the more global the features [8]. Zang Haijuan's method of extracting image feature values by using matrix-like kernels overcomes the problem of large spatial complexity of direct use of face image data, increases the nonlinear structure of features, and improves the expression ability of feature vectors [9]. In the work of Zhang Jiwei et al, they concluded that constrained sparse matching method is used to effectively solve the similarity between the face image sequences and automatically select similar face image sequences, which improves the feature matching accuracy [10]. It can be seen that with the continuous advancement of artificial intelligence, more and more experts and scholars are involved in the research and discussion of the frontier technology of face recognition.

The LeNet-5 [11] model structure was proposed by computer scientist Professor Yann LeCun in 1998. This model is a convolutional neural network used to identify handwritten digits. It was used by most banks in the United States to identify handwritten cheques. The number has reached a commercial level, indicating that the algorithm has a high recognition rate. In this paper, the 6-layer CNN network is designed for the face recognition of the olivettifaces face database based on the model structure. The first 4 layers of the network are alternately cascaded by convolutional layer and pooled layer for image feature extraction, and then the network layer is followed by The full connection layer, the last layer is the Softmax classifier with strong nonlinear classification ability for the output layer, and outputs the probability OUTPUT of the classification result.

2. CNN Modle

2.1. Network Structure

The CNN infrastructure model designed in this experiment is shown in Figure 1. It includes the input layer, convolution layer 1, pooling layer 1, convolution layer 2, pooling layer 2, fully connected layer and Softmax regression classification layer. According to Figure 1, we find that the main difference between CNN and other traditional neural networks is that the processing of convolutional and pooling layers is added to extract image features. This paper introduces these two layers.

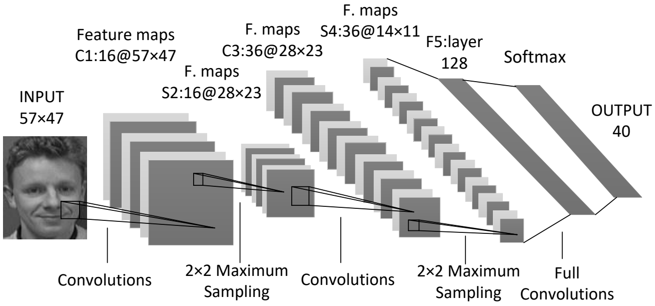


Figure 1. CNN infrastructure model designed for experiment.

2.2. Convolution Layer

The convolutional layer, also known as the feature extraction layer, represents the layers of all the components obtained by filtering the input image. The input of each neuron is connected to the local receptive field of the previous layer and the local features are extracted [12]. The meaning of this layer is that the original image is subjected to convolution operation to generate multiple images. The convolution process according to formula (1) is shown in Figure 2. First, the convolution kernel value is determined in a random manner, and the image to be converted is second from the left. To the right, top to bottom, select the image matrix in order and finally multiply the selected image matrix and the convolution kernel. Each convolution is a feature extraction method, just like a sieve, which filters out the qualified parts of the image. Generally speaking, the larger the activation value, the more the screening conditions are met. Convolution does not change the image size, so the processed image size remains the same, similar to the filter effect, which helps us extract different features of the input, such as edges, lines, and angles [13].

The mathematical expression of the convolution process [14] is:

$$s(i,j) = (X \times W)(i,j) + b = \sum_{k-1}^{n\_in}(X_k \times W_k)(i,j) + b \qquad (1)$$

Where n_in is the number of input matrices or the dimension of the last dimension of the tensor, b is the offset of the neurons on the convolutional layer feature map, X_k represents the kth input matrix, and W_k represents the convolution kernel k sub-convolution kernel matrices, s(i,j) is the value of the corresponding position element of the output matrix corresponding to the convolution kernel W.

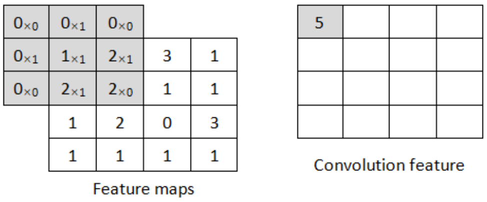


Figure 2. Convolution operation process

2.3. Pooling Layer

The pooling layer, also known as the feature mapping layer, represents the layer obtained by downsampling the input image. Each computing layer of the network consists of multiple feature maps, each of which is mapped to a plane with equal weights for all neurons on the plane. The feature mapping structure uses a small sigmoid function that affects the function kernel as the activation function of the convolutional network so that the feature mapping structure has displacement invariance [12]. Images processed by the pooling layer usually have the following advantages: first, the data points to be processed are reduced, and the time required for subsequent operations is reduced; secondly, because the sampling is reduced, the image position difference becomes smaller; finally, due to the number of parameters And the amount of calculation is reduced, and the over-fitting is also controlled to some extent. In this experiment, the maximum pooling method is used, the size is 2×2, that is, the input feature map is divided into non-overlapping 2×2 rectangles, and the maximum value is taken for each rectangle. The implementation process is shown in Figure 3.

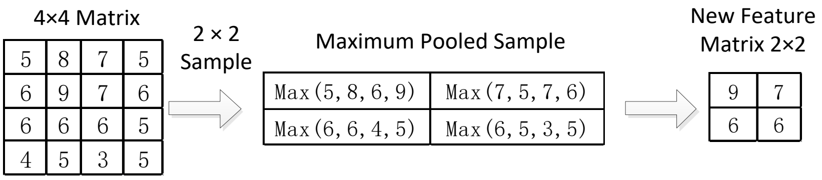


Figure 3. Maximum pooling method.

3. CNN Training Experment

3.1. Dataset Selection

The experimental data used is New York University's olivettifaces face database, which consists of 400 pictures of 40 people, that is, each person's face picture is 10. The gray level of each picture is 8 bits, the gray level of each pixel is between 0 and 255, and the total picture size is 1190×942. There are 20×20 faces in total, so the size of each face is (1190/20) × (942/20) is 57 × 47 = 2679. We divide the raw data into three parts, the training data is 320 samples, the verification data and the test data are all 40 samples, and some human faces are shown in Figure 4.

Figure 4. Olivettifaces face library sample

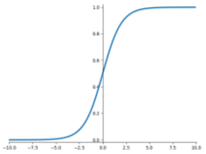
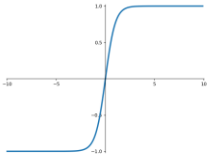
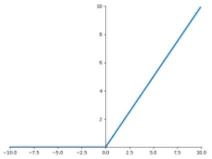
3.2. Data Preprocessing

The experimental environment of this experiment is Windows7+Python3.6+ Keras, the CPU is 3.40GHz Intel i7-6700, the memory is DDR4 16G, and the graphics card is NVIDIA GT-720 2G. First, the image is read by loading the PIL module. Secondly, numpy.asarray is used to normalize the data, that is, each image is drawn into a one-dimensional vector 2679 and a label is added; finally, the pickle module is loaded to save these normalized values and labeled vector.

3.3. Activation Function Selection

If a linear function is used as the activation function in the neural network layer, the connection relationship between the networks of the layers is also linear, which is meaningless to the deep neural network. Therefore, this experiment chooses to modify the linear unit ReLU as the activation function of neurons. The reason why ReLU is chosen instead of the traditional sigmoid or Tanh as the experimental activation function is because sigmoid and other functions have calculations when performing inverse propagation to obtain error gradients. The large amount and easy to lead to the disappearance of the gradient, so that the deep network training cannot be completed; and ReLU will make the output of a part of the neurons 0 to form network sparsity, while reducing the interdependence of parameters, alleviating the over-fitting problem, and secondly Compared with sigmoid and other functions, ReLU is unilateral and more in line with the characteristics of biological neurons. The specific function equations and curves are shown in Table 1.

Table 1. Some commonly used activation functions.

Name	Equation	Plot
sigmoid	f(x)=$\frac{1}{1+e^{-x}}$	
Tanh	tanh(x)=$\frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$	
ReLU	f(x)=max(0,x)	

3.4. Establish CNN Model

The operation of constructing this experimental model is as follows:

- Establish a Sequential linear stacking model of keras to facilitate the subsequent addition of various neural network layers;
- Create convolution layer 1, each face image size is 57 × 47, the number of construct filters is 16, each filter size is 3 × 3, and by adding the same parameter to ensure that the convolution operation will be generated The convolution image size remains the same, and ReLu [15-16] is selected as the activation function of the layer, so after the first convolution operation, 16 images are generated, and the image size is still 57×47;
- The pooling layer 1 is created, and the first downsampling is performed, and 16 57×47 images are reduced to half, and the number remains unchanged;
- Create convolution layer 2 and construct a filter number of 36, which converts the original 16 images into 36 images;

- Create pooling layer 2, perform the second downsampling, reduce the image of pooled layer 1 by half again, and add Dropout to this layer. The advantage is that it will randomly be in the neural network every iteration training. Partially abandon neurons to avoid overfitting [17];
- Establish a flat layer to convert the image of the pooled layer 2 into a one-dimensional vector;
- Create a hidden layer, select ReLu as the activation function for the layer, and add the Dropout layer to the model again. The CNN model summary information is listed in Table 2.
- Establish the output layer and use the softmax activation function to convert and predict the probability.

Table 2. CNN model 16-36-128 information summary.

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 57, 47, 16)	160
max_pooling2d_1	(MaxPooling2 (None, 28, 23, 16)	0
conv2d_2 (Conv2D)	(None, 28, 23, 36)	5220
max_pooling2d_2	(MaxPooling2 (None, 14, 11, 36)	0
flatten_1 (Flatten)	(None, 5544)	0
dropout_1 (Dropout)	(None, 5544)	0
dense_1 (Dense)	(None, 128)	709760
dropout_2 (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 40)	5160

3.5. Experimental Results and Analysis

According to the model summary information in Table 1 above, we find that the number of hidden layer neurons and the number of two convolutional layer feature maps have a great influence on the network accuracy. In this design model C1-C2-H [18], respectively, Where C1 represents the number of feature maps of convolutional layer 1, C2 represents the number of feature maps of convolutional layer 2, and H represents the number of hidden layer neurons.

3.5.1. View the impact on the network by changing the number of hidden layer neurons. The 6-layer CNN model selected for testing was: 16-36-128, 16-36-256, 16-36-512, 16-36-1024. The two convolutional parameters of this experiment remain unchanged, and only the number of hidden layer neurons in the fully connected layer is adjusted. The results after the operation are shown in Figure 5, Figure 6 and Table 3.

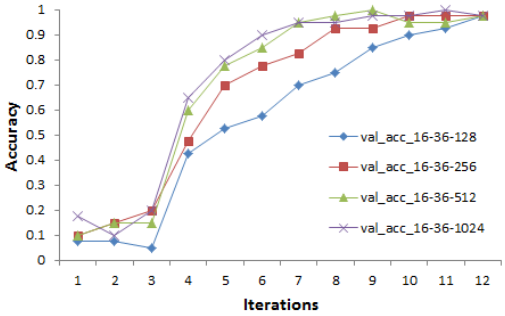


Figure 5. The effect of the number of hidden layer neurons on the test results

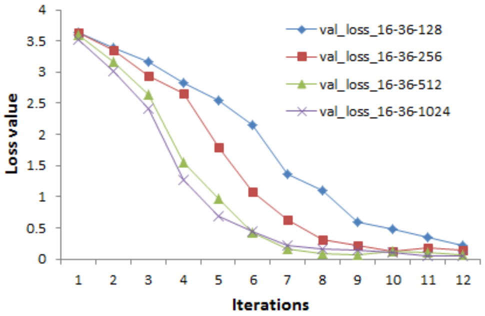


Figure 6. The effect of the number of hidden layer neurons on cross-entropy

Table 3. The number of hidden layer neurons affects the correct rate of face recognition.

Model	Number of faces	16-36-128	16-36-256	16-36-512	16-36-1024
Time/s		14	25	26	50
Training-set	320	87.19	93.44	98.75	99.69

Verification-set	40	97.50	97.50	97.50	97.50
Test-set	40	97.50	97.50	97.50	97.50

By observing the experimental results, we found that as the number of neurons increases, the cross-entropy decreases continuously, that is, the error becomes lower and lower, and the accuracy becomes higher and higher, which has a greater impact on the training set, and the recognition rate is increased from 87.19 to 99.69. However, the recognition rate of the verification set and the test set has stabilized to 97.50, so it can be temporarily determined that the optimal CNN model of this experiment is 16-36-1024.

3.5.2. View the impact on the network by changing the number of feature maps of convolutional layer 1 and convolutional layer 2. The CNN models designed according to the optimal number of neurons in 1024 determined by the above experiment are 16-36-1024, 26-56-1024, 36-76-1024, 76-156-1024. In this experiment, the number of hidden layer neurons is kept unchanged. By changing the number of image features of two convolutional layers, the best CNN faces recognition model for this experiment is found. The results after the operation are shown in Figure 7, Figure 8 and Table 4.

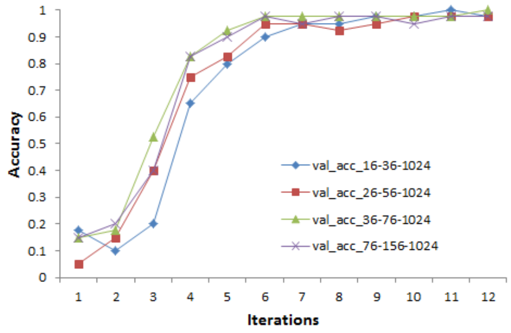


Figure 7. Effect of the number of image features of the convolutional layer on the test results

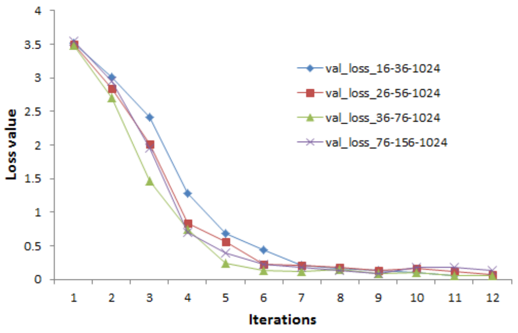


Figure 8. Effect of the number of features of convolutional layer images on cross-entropy

Table 4. The number of convolutional layer image features affects the correct rate of face recognition.

Model	Number of faces	16-36-1024	26-56-1024	36-76-1024	76-156-1024
Time/s		50	86	110	219
Training-set	320	99.69	99.37	100	99.06
Verification-set	40	97.50	97.50	100	97.50
Test-set	40	97.50	97.50	100	97.50

By observing the experimental results, we found that the more the number of features added to the convolutional layer image, the longer the running time and the stability of the training set recognition rate and the verification set and the test set has little effect, and basically remain stable. It can be seen from Figure 8 that although the cross-entropy changes, it also has little effect. When the CNN model is 36-76-1024, the face recognition rate reaches 100%, so it can be determined that it is the best CNN model for this experiment.

4. Conclusion

Since the face recognition process is more complicated than handwritten digit recognition, the experiment is based on the LeNet-5 handwritten digit recognition algorithm to obtain higher accuracy. By changing the experimental parameters, ie the number of hidden layer neurons and the number of two convolutional layer features, the current best CNN model is obtained and a high recognition rate is achieved, but the CNN model also has some shortcomings, such as the network. The structure is susceptible to the image database, and the network model is capable of classifying and identifying natural images. Further experiments and research are needed.

5. References

[1] YANG Jucheng, LIU Na, FANG Shanshan,et al. Review of face recognition methods based on deep learning[J]. Journal of Tianjin University of Science and Technology, 2016, 31(06): 1-10.

[2] JING Chenkai, SONG Tao, ZHUANG Lei, et al. A survey of face recognition technology based on deep convolutional neural networks [J]. Computer Applications and Software, 2018, 35(01): 223-231.

[3] ZHENG Jian, WANG Zhiming, ZHANG Ning. Improved face recognition method based on deep Learning [J]. Computer and Modernization, 2018(12): 90-95.

[4] YU Daling, HUANG Hongqiong. A face recognition method based on CNN[J].Microcomputer and Applications,2017, 36(14):49-51.

[5] Brunelli R, Poggio T.Face recognition:Features versus templates[J].IEEE transactions on pattern analysis and machine intelligence,1993,15(10): 1042-1052.

[6] LI Jianzhong, LEI Lili, Li Wei, et al. Dynamic face recognition based on geometric features [J].Science Technology and Engineering, 2010,10(28):6894-6898.

[7] Rumelhart D E,Hinton G E,Williams R J.Learning representations by back-propagating errors [J].Cognitive modeling,1988,5(3):1.

[8] CHEN Yaodan, WANG Lianming.Convolutional neural network for face recognition [J]. Journal of Northeast Normal University(Natural Science Edition), 2016, 48(02): 70-76.

[9] ZANG Haijuan. Face recognition method based on feature fusion of quasi- matrix neural kernel[J]. Computer Engineering and Applications, 2018,54(24):116-122.

[10] ZHANG Jiwei, NIU Shaozhang, CAO Zhiyi, et al. Face recognition via deep learning and constraint sparse representation [J]. Transactions of Beijing Institute of Technology, 2019,39 (03):255-261.

[11] LeCun Y,Bottou L,Bengio Y, et al.Gradient-based learning applied to document recognition [J]. Proceedings of the IEEE,1998, 86, (11): 2278- 2324.

[12] WANG Wanliang. Artificial intelligence and application [M] (third edition). Beijing: Higher Education Press, 2016.300-308.

[13] LIN Dagui. TensorFlow+Keras deep learning artificial intelligence practice application [M]. Beijing: Tsinghua University Press, 2018.91-104.

[14] Syafeeza A R, Khalil-Hani M, Liew S S, et al.Convolutional neural network for face recognition with pose and Illumination Variation [J]. International Journal of Engineering&Technology, 2014, 6 (1): 44-57.

[15] Nair V, Hinton G E.Rectified linear units improve restricted Boltzmann machines [C] Proceedings of the 27th international conference on machine learning (ICML-10).2010:807-814.

[16] Glorot X, Bordes A, Bengio Y. Deep Sparse Rectifier Neural Networks [C] International Conference on Artificial Intelligence and Statistics, 2012.905-912.

[17] Hinton G E,Srivastava N,Krizhevsky A,et al.Improving neural networks by preventing co-adaptation of feature detectors[J].Computer Science, 2012, 3(4):212-223.

[18] DONG Luozhen, CHEN Zhangping.Face recognition based on improved convolution neural network[J].Journal of Hangzhou Dianzi University:Natural Science,2018,38(05): 55-59.

Acknowledgments

The research of this project was supported by the fund of the Higher Education Institute of Cloud Computing and Big Data of Guangdong Higher Education Institute (No. GDYJSKT19-07). It has also received strong support from the Research Department of Shanwei Polytechnic and Shanwei Innovation Industrial Design & Research Institute.