Design of face recognition system based on CNN

Yong Li 1*, Zhe Wang 2, Yang Li 3, Xu Zhao 4, Hanwen Huang 5
1 Zaozhuang Vocational College, Zaozhuang, Shandong, 277000, China
2 Jilin University, Changchun, Jilin, 130012, China
3 Central South University, Changsha, Hunan, 410000, China
4 Monash University, Melbourne, Victoria, 3163, Australia
5 Xi’an Jiaotong University, Xi’an, Shanxi, 710049, China
*Corresponding author’s e-mail: liyong5100769@126.com

Abstract. This paper studies and implements a face recognition system based on convolutional neural network. The system firstly uses AdaBoost algorithm to detect the face, that is to get the position and size of the face accurately in the image, then uses the deep convolution neural network to extract the face features and classify them. Finally, the hardware application Altera DE1_SOC development board, designs and completes a face recognition system with high recognition rate.

1. Introduction
As a kind of biometric recognition technology, face recognition technology has unique stability, uniqueness and convenience compared with fingerprint recognition, retina recognition, iris recognition and other technologies, so it is widely used in the field of identity recognition. Face recognition technology can be divided into four categories: Geometric based method, algebraic based method, template based method and depth learning based method. Geometric based feature extraction method has good classification and recognition effect for face image with good quality, but it is not ideal for some or poor quality images. Because of its relatively low computational complexity, algebraic feature-based methods have attracted a lot of attention, but their shortcomings are that they are seriously affected by expression changes, illumination angle intensity changes and perspective changes, and their robustness is poor. The advantage of template matching is that it is easy to implement, and the disadvantage is that it is difficult to deal with the pose and scale changes effectively. Face recognition algorithm based on deep learning has better robustness to light, pose and complex background, and can greatly improve the recognition accuracy. However, deep learning also faces some challenges, such as big data training. Nowadays, the amount of training data of deep learning algorithm has reached millions and tens of millions, or even larger scale. Some existing technologies are not suitable for big data processing [1-2].

In this paper, the convolution neural network (CNN) is used to solve the problem of low efficiency of deep learning in big data processing. Based on this point, a face recognition system based on CNN is designed. The system has good robustness and high recognition rate.
2. Core algorithm design of the system
The core algorithm of the face recognition system designed in this paper consists of two parts: AdaBoost based face detection algorithm and CNN based face feature extraction algorithm.

2.1. Research on AdaBoost based face detection algorithm
AdaBoost algorithm is one of the boosting algorithms. AdaBoost algorithm will automatically optimize the false recognition rate of hypothesis according to the result of weak learning. Therefore, it doesn't matter if AdaBoost algorithm doesn't know the lower limit of false recognition rate. Therefore, even if there is no prior knowledge of weak learners, it can keep the efficiency of Boosting algorithm, so it is widely used.

The principle of AdaBoost algorithm is: given training set \((x_1, y_1), \ldots, (x_N, y_N)\) where \(\{x_i \mid x_i \in X\}\) represents the set of training samples. Use category label \(i \in \{1, -1\}\) to indicate whether \(x_i\) is the correct category label, \(i = 1, 2, \ldots, N\). When initializing training samples, the weight of each sample is \(1/N\), and the sample initialization distribution on the training set is
\[D_1(i) = \frac{1}{N}\] (1)

At first, the weak classifier is used to train the sample set and obtain the weak hypothesis sequence \(h_i\). After \(t\) iterations, after each iteration, the weight of training set is allocated again according to whether the training is correct or not. For the training cases of error classification, we give them more weight, and then train them on the new distribution to get the hypothesis sequence \(h_1, h_2, \ldots, h_t\). For \(t = 1, 2, \ldots, T\) , find the weak classifier \(h_i : X \rightarrow \{-1, 1\}\) with the smallest error rate on the distribution \(D_t\) where the error rate of the weak classifier on the distribution \(D_t\) is:
\[\varepsilon_t = P_{D_t}(h_i(x_i) \neq y_i)\] (2)

Calculate the weight of the weak classifier:
\[\alpha_t = \frac{1}{2}\ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)\] (3)

Update the distribution of training samples:
\[D_{t+1}(i) = \frac{D_t(i)e^{-\alpha_t y_i h_i(x_i)}}{Z_t}\] (4)

Where \(Z_t\) is the normalization constant.
A hypothesis \(H\) can be obtained by voting with weight. The final strong classifier is:
\[H_{\text{final}}(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_i(x)\right)\] (5)

Let \(\varepsilon_t = 1/2 - \gamma_t\), for the reason that the error rate of weak classifier is always lower than that of random guess, so \(\gamma_t > 0\), the training error is:
\[R_{\gamma}(H_{\text{final}}) \leq e^{-\sum_{t=1}^{T} \gamma_t}\] (6)

Let \(\forall t, \gamma \geq \gamma_t > 0\), then formula (6) is established, which indicating that the training error will be reduced in the last session with the increase of training rounds.
AdaBoost is an iterative algorithm. After each iteration, the weight needs to be adjusted. In this way, more attention can be paid to the misclassified samples, and the misclassified samples will be obvious, forming a new data distribution. Under this data distribution, weak classifiers are trained again to improve the performance of weak classifiers. These weak classifiers can be cascaded to form a strong classifier.
During the implementation of AdaBoost algorithm, the rectangular feature of image, namely Haar feature, is selected. AdaBoost algorithm is influenced by feature selection and feature value calculation. The feature information of face can be represented by rectangle feature.

When the data is insufficient, the problem of coding the state of a specific region can be solved by the detection based on the feature, and the rectangular feature has the advantage of high speed. The disadvantage of rectangular features is that they are sensitive to simple graphic structures such as edges and line segments. But it can only describe the structure of specific direction such as vertical, horizontal and diagonal, so the representation of features is relatively rough. When the number of detectors is 24, the number of rectangular features is far more than 160000. Therefore, not all the rectangular features used are useful. It is necessary to select appropriate features and combine these features into a strong classifier. Common rectangular features are shown in Figure 1.

Several white rectangles and several black rectangles are the basic elements of the Harr feature template. The difference between the sum of the pixels in the white rectangle and the sum of the pixels in the black rectangle is the template characteristic value. Haar feature represents the change of gray level in the adjacent rectangular region of the image. Each feature of Haar feature can be expressed as:

\[
\text{feature}_j = \sum_{i=1}^{N} w_i \times \text{rectsum}(r_i)
\]  

(7)

In formula (7), \(w_i\) is the weight of rectangular \(r_i\), \(\text{rectsum}(r_i)\) is the sum of pixels in rectangular \(r_i\), \(N\) is the number of rectangles that make up \(\text{feature}_j\).

After we get the rectangular feature, the purpose is to get the value of the rectangular feature. The integral graph of coordinate \(A(x,y)\) in the integral graph is the sum of all image pixels of the shadow block, which can be defined as

\[
i(x,y) = \sum_{x',y' \leq x,y} i(x',y')
\]  

(8)

Where \(i(x,y)\) represents the integral graph, \(i(x,y)\) represents the color value of the original image; for an image without color, it is its gray value, if the image has color, it is the range of this point from 0 to 255. The integral graph can also be used to calculate the pixels of an area.

In order to realize face detection, we need to design cascaded classifiers. The algorithm principle of cascading several strong classifiers is to arrange several strong classifiers from weak to strong. The false recognition rate can be properly sacrificed, but the detection rate of strong classifier cannot be reduced. We train cascaded classifiers to improve the accuracy of face detection. Here we will talk about the detection system in the Harr classifier. A large picture is given to the Harr classifier for multi-scale detection of multiple regions. In this case, a picture is divided into multiple regions, and each region is detected one by one. The images used in training are relatively small, so it needs multi-scale detection to detect large faces. At present, there are two multi-scale detection mechanisms. One is not to adjust the proportion of the search window, but to adjust the image. The disadvantage of this method is that the adjusted image needs to calculate the region characteristic value, which is
inefficient. Another way is to initialize the size of search window continuously when training samples, and improve the efficiency through continuous window search.

The specific training process is shown in Figure 2.

(1) The Harr like feature is used to represent the face, the Haar feature location parameters are calculated, and the threshold parameters of each weak classifier are calculated according to the feature location parameters;

(2) The AdaBoost algorithm is used to select some rectangular features (weak classifier) that can best represent the face, and the weak classifier is constructed into a strong classifier according to the way of weighted voting;

(3) A cascade classifier is composed of several trained strong classifiers in series. The cascade structure can effectively improve the detection speed of the classifier, and then through a large number of face training to achieve the effect of face detection.

![Figure 2. Training process of AdaBoost algorithm](image)

### 2.2. Research on design of face feature extraction algorithm

In this paper we use CNN based feature extraction method. CNN has powerful feature extraction ability, which can extract complex data layer by layer, and finally form ideal features suitable for pattern classification.

CNN is a kind of hierarchical model, whose input raw data can be image data, audio data, etc. After convolution, pooling, activation function mapping and other operations, CNN inputs the semantic information in the original data input layer, layer by layer, which is called feedforward operation. Finally, the last layer of CNN transforms target tasks such as classification and regression into objective functions. Then calculate the difference between the real value and the predicted value, use the back propagation algorithm to feed the loss forward from the last layer, update the parameters of each layer, and then feed forward again. Finally, the neural network model converges to achieve the goal of training by repeatedly repeating [3-4]. CNN structure is shown in Figure 3.

![Figure 3. CNN structure](image)

CNN adds the operations of the convolution layer and the pooling layer to the original data one by one as the basic unit, and finally calculates them through the loss function. The data in this process is a
kind of three-dimensional tensor. In the field of computer vision, the original data of convolutional neural network is usually RGB picture, which is recorded as x1. x2 is obtained after the first layer operation, and the parameters in the first layer operation are recorded as w1. At this time, the previous x2 can be used as the input of the second layer. In the whole process, each operation layer can be convolution, pooling, nonlinear mapping \([5-7]\).

\[
x^1 \rightarrow w^1 \rightarrow x^2 \rightarrow \cdots \rightarrow x^L \rightarrow w^L \rightarrow z
\]

(9)

Then, the network ends with the calculation of loss function. If y corresponds to the true flag of input x1, then the loss function is

\[
z = L(x^L, y)
\]

(10)

The input layer of the convolutional neural network used in this paper is: 68 × 68 size picture, single channel. The first layer convolution: 32 convolution kernels of 2×2 size. The first layer is a 2×2 kernel. The second layer convolution: 64 2×2 convolution kernels. The second layer is a 2×2 kernel. The third layer convolution: it is fully connected with the previous layer, with 96 convolution kernels of 2×2. Full connection input layer: the input is to connect the output of the third layer Max pooling into a 7776 (9×9 ×96) dimensional one-dimensional vector, as the input of the layer, the output is 1024 dimensional. Fully connected softmax output layer: input 1024 dimensions, output 67, each dimension of output is the probability that the picture belongs to this category.

We use Tensorflow, Google's second generation machine learning system, to train CNN. Tensorflow has high flexibility, real portability, automatic differentiation, multi-language support and other features \([8-7]\). The face database used in the training is CMU pie face database, which includes 40000 photos from 68 people, including 13 posture conditions, 43 lighting conditions and 4 facial expressions of each person. We select a model with good training effect, and save the structure and parameters of the facial features extracted in CNN to design the feature extraction module.

The loss function in the training process is the cross entropy loss function, and the optimizer is Adam optimizer in Tensorflow, The learning speed is 0.0001, the training method is batch gradient descent method, the training times are 4000, each time 2000 random pictures are input into the training data, and the current error is printed every 50 times in the training; we use Tensorboard tool in Tensorflow to visualize the network, and get that the recognition accuracy increases with the increase of training times, and the training times tend to be stable around 4K The final training accuracy is 99.06%.

### 3. Implementation of face recognition system

The implementation of face recognition system mainly includes two parts: software implementation and hardware implementation, which are described below.

#### 3.1. System software implementation

The realization of system software can be divided into two parts. The first part is face detection. The face detection development environment adopts the open source computer vision library (openCV), which is supported by willow garage. In open CV, the model of face detection has been established as an XML file, which contains the training results of the classifier with Harr features. By loading this file, we omit the process of building our own cascading table. With cascading tables, we only need to pass the simultaneous interpreting images and cascading tables to the target detection algorithm of open CV, and then we can get a set of detected faces to achieve the purpose of detecting faces. The second part is face feature extraction and recognition. Firstly, based on Tensorflow, we use CMU PIE face database to train CNN, select the optimal training model and build feature extractor by writing python, then input standard face to form face feature database; when the face image to be tested enters the system, we extract the face feature to be tested by feature extractor, and then compare it to be tested The Euclidean distance between the face feature and the standard face feature is measured to complete the recognition. Finally, the python script is invoked in the C++ program, and the recognition result is displayed in the Qt interface, so that the whole face recognition process can be completed.
3.2. System hardware implementation
We mainly use Altera DE1 SoC development board; the main frequency of ARM core in the development board is 1G; the running memory is 1G DDR3 SDRAM; the storage space is 8G of microSD card.

![Diagram of system hardware](image)

The hardware implementation block diagram of the system is shown in Figure 4. Firstly, USB camera collects images through UVC driver and v4l2 interface. In the arm core, the image is input to the face recognition program embedded in the arm core. The display screen connected by VGA interface displays the human-computer interface and related function buttons on the display screen, and realizes the related functions of the face recognition system by clicking the buttons.

4. System test
The test results are shown in Figure 5. In the face recognition operation interface shown in Figure 5, first click the input button, the system will display the face image taken and entered, and prompt to input the name to complete the whole input process, then click to start recognition, the system will collect the current camera image and recognize with the entered face image, and finally display the recognition results in the display interface. Taking the existing laboratory staff as the test object, through a preliminary test, it is found that the system fully realizes the function of face recognition and has a better recognition accuracy. It has good robustness to face pose and illumination. New people can be added to the system without training. In addition, the human-computer interface is relatively complete.

![Test results](image)

5. Summary
In this paper, face location and detection algorithm is used to extract face images from complex scenes. By using convolution neural network algorithm in depth learning, the recognition rate of face
recognition system is improved when face pose and illumination change. Then the convolution feature extraction layer is trained by high-performance machine, and the convolution feature extraction layer is loaded by low-performance machine, which avoids the retraining process of adding new face. Finally, the embedded technology is used to load the whole face recognition system into SoC system, which improves the practicability of the system. Of course, there are also some problems in the system, for example, our convolution neural network needs a lot of convolution operations, which takes a long time, resulting in a slow recognition rate when the hardware performance of the system is not enough. In the future work, we need to constantly improve the neural network algorithm, so as to improve the recognition rate.

References
[1] Alham, N.K., Li, M., Liu, Y., et al. (2013) A MapReduce-based distributed SVM ensemble for scalable image classification and annotation[J]. Computers & Mathematics with Applications, 66:1920-1934.

[2] Moeskops, P., Viergever, M.A., Mendrik, A.M., et al. (2016) Automatic Segmentation of MR Brain Images With a Convolutional Neural Network. IEEE Transactions on Medical Imaging, 35:1252-1261.

[3] Lu, J.W., Liong, V.E, Wang, G., et al. (2015) Joint Feature Learning for Face Recognition. IEEE Transactions on Information Forensics and Security., 10: 1317–1383.

[4] Gao, S.H., Zhang, Y.T., Jia, K., et al. (2015) Single Sample Face Recognition via Learning Deep Supervised Autoencoders. IEEE Transactions on Information Forensics and Security., 10: 2108–2118.

[5] Lu, J.W., Wang, G., Zhou, J. (2017) Simultaneous Feature and Dictionary Learning for Image Set Based Face Recognition. IEEE Transactions on Image Processing., 26: 4042–4054.

[6] Wang, X.M., Gu, T.L., Luo, X.N., et al. (2019) A User Study on the Capability of Three Geo-Based Features in Analyzing and Locating Trajectories. IEEE Transactions on Intelligent Transportation Systems., 20: 3375–3385.

[7] Zhu, M.F., Chen, W., Xia, J.Z., et al. (2019) Location2vec: A Situation-Aware Representation for Visual Exploration of Urban Locations. IEEE Transactions on Intelligent Transportation Systems., 20: 3981–3990.

[8] Peng, F., Zhou, D.L., Long, M. (2017) Discriminating of Natual Images and Computer Generated Graphics Basing on Multi-Fracal and Regression Analysis. AEU-International Journal of Electronics and Communications., 71: 72–81.

[9] Tokuda, E., Pedrini, H., Rocha, A. (2013) Computer Generated Images vs. Digital Photographs: A Synergetic Feature and Classifier Combination Approach. Journal of Visual Communion and the Image Representation., 24: 1276–1292.

[10] Zhang, L., Liu, J., Zhang, B., et al. (2019) Deep Cascade Model-Based Face Recognition: When Deep-Layered Learning Meets Small Data. IEEE Transactions on Image Processing., 29: 1016–1029.