Hybrid Face Recognition Method Based on Gabor Wavelet Transform and VGG Convolutional Neural Network with Improved Pooling Strategy

Cheng Xing, Jie-Sheng Wang*, and Bo-wen Zheng

Abstract—**A face recognition algorithm based on VGG convolutional neural network with an improved pooling method is proposed. The recognition effect is not good as the recognized images may be interfered by various factors. For the situation that images are affected by illumination, the number of images are few and the quality of images are not good, based on the image pretreatment with normalization and de-average, the histogram homogenization is adopted to reduce the illumination effect, the randomly cutting out the number of images is to reduce the possibility of the network's over-fitted and the Gabor wavelet transform is adopted to enhance the images. Then the Faster R-CNN network is adopted to carry out the face detection experiments on LFW database. Aiming at the problem that there are three full-connection layers in traditional VGG-16 network, which has a lot of parameters be produced in network training, the traditional VGG-16 network is improved by reducing the number of fully connected layers, replacing the original max-pooling method with a random square pooling method, which changes the last pooling layer to global mean pooling by referring to the GoogLeNet network method. Finally the simulation experiments are carried out on the LFW database and the self-built database. It is eventually found that the improved method effectively reduces the network training parameters, greatly reduces the network training time and obtains a good recognition rate.**

Index Terms—**convolutional neural network, face recognition, pooling method, Gabor wavelet transform algorithm**

I. INTRODUCTION

FACE recognition is a type of biometric technology, which requires only one camera for information collection, so this simple identification method is the safest and most effective way to verify personal identity [1-2]. In the 1950s, scientists conducted preliminary research on face recognition. In the 1960s, face recognition has officially opened up research in various disciplines and fields [3]. Computer scientists, neuroscientists, biologists, psychologists, and other scholars in different disciplines have all carried out the face recognition research [4-9]. Although automatic recognition was not implemented at this stage [10-11], it had made a leap in the face recognition technology. From the 1960s to the 1990s, this was the earliest mechanical recognition stage [12], which was the initial stage of face recognition. At this stage, there are not many important achievements, and there is no breakthrough in practical application. From the early 1990s to the end of the 1990s, this was an outstanding face recognition age [13]. During this period, the face recognition technology developed rapidly, and various excellent algorithms were proposed. Especially in the ideal situation, such as accurate information collection, object coordination, and the size of the face database is small, the face recognition in this case has been well recognized [14-16]. Face recognition technology has also been recognized by commercial companies and has begun to be put into practical commercial applications. From the end of the 20th century to the present, face recognition technology has been applied to all aspects of life. The poorly robust face recognition caused by the fact that object cannot coordinate and non-ideal acquisition conditions (such as attitude problems, illumination problems, and facial occlusion problems), and the face recognition based on huge database are all nowadays main research directions and hot issues [17-19]. Generally speaking, face recognition is mainly divided into four stages: one is the face detection and positioning stage, the other is the face image preprocessing stage, the third is the face image representation stage, and the fourth is the face image matching and classification stage. The face detection stage mainly detects and locates the face information by using a two-dimensional image containing face information, and then extracts the face information from the corresponding scene. In the face image preprocessing stage, since the obtained face information is interfered and limited by many factors, some correction processing must be performed at this stage to obtain easy-to-handle face information. The face representation stage, which is the face feature extraction and the process of constructing the face feature model, uses the two-dimensional face features to represent the face. In the face recognition stage, the extracted

Manuscript received July 12, 2020; revised December 20, 2020. This work was supported by the Basic Scientific Research Project of Institution of Higher Learning of Liaoning Province (Grant No. 2017FWDF10), and the Project by Liaoning Provincial Natural Science Foundation of China (Grant No. 20180550700).

Cheng Xing is a Ph.D candidate in the School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, 114044, PR China (e-mail: xingcheng0811@163.com).

Jie-Sheng Wang is with the School of International Finance and Banking, University of Science and Technology Liaoning, Anshan, 114051, PR China; National Financial Security and System Equipment Engineering Research Center, University of Science and Technology Liaoning. (Corresponding author, phone: 86-0412-2538246; fax: 86-0412-2538244; e-mail: wang_jiesheng@126.com).

Bo-Wen Zheng is with the School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, 114051, PR China (e-mail:18842038281@163.com).

face image information and the face database information are matched one by one to obtain correct identity information, and then to accomplish face recognition. The face recognition stage includes classification and matching.

A face detection and recognition method is proposed based on convolutional neural network. Normalize and de-average methods are applied in the pretreatment on the face images. Image-enhanced random clipping, histogram homogenization and Gabor wavelet denoising processing are adopted for disposing the face images with less data and lower image quality. Then, a traditional face detection method is introduced by combing Harr and Adaboost methods in details. The faster R-CNN network is analyzed, and the face detection experiments are performed on the LFW database and the self-built database. The structure and pooling layer of VGG-16 network are improved and the face recognition experiments and results analysis are performed on the LFW face database and self-built database to shown the effectiveness of the proposed method.

II. Convolutional Neural Networks

A. Basic Structure of CNN

The convolutional neural network originated in the 1960s was proposed by two famous scholars, Hubel and Wiese. By observing the cat's cerebral cortex visual system, they found that each individual visual neuron would process a small piece of sensory area, and then the complexity of the feedback neural network is effectively reduced. CNN has two biggest features: 1) Sparse connectivity. It performs partial connection for the non-fully connected manner between each neuron. 2) Weight sharing. Convolution kernels with the same weights is used to realize the feature extraction for different image regions. Convolutional neural network has a new neural network structure evolved from Multilayer Perceptron (MLP). It is mainly composed of convolutional layer, pooled layer and fully connected layer, where C represents the convolutional layer and S represents the pooled layer. The convolutional layer is for feature extraction, the pooled layer performs dimensionality sampling for the extracted features, and the fully connected layer is mainly for classification. The CNN structure is shown in Fig. 1.

B. Convolutional Layer

Because the convolutional layer is connected with the previous layer of the neural network through the convolution kernel, the input of the convolutional layer is the output of the previous layer. The purpose of the convolutional layer is to extract the feature vector of the image by the operation of the convolution kernel. The lower convolution layer extracts simpler image features, such as lines, corners and borders. A more complex features are realized by combining the low-level rich edge features through the high-level convolutional layer. The convolutional layer operation is actually equivalent to a linear correlation operation, which can be expressed as follows.

$$x_j^l = \sigma\left(\sum_{i\epsilon M_j} x_i^{l-1} * w_{ij}^l + b_j^l\right) \quad (1)$$

where, x is a two-dimensional vector of size $i \times j$, w is the convolution kernel, b is the offset, l is the number of layers, and M_j is the characteristic image of the input. An example of convolutional motion of a convolution kernel on an image is shown in Fig. 2, where the original source pixel is 5*5, the convolution kernel is 3*3, and the step size is 1.

C. Pooling Layer

The pooling layer is also called the downsampling layer, which is usually connected behind the convolutional layer and whose main purpose is to reduce the feature image. Generally, the downsampling layer selects the size of 2*2. The pooling methods in convolutional neural networks mainly include Max-Pooling, Mean-Pooling, and Stochastic Pooling [20]. The mathematical expression for the pooling layer can be described as:

$$x_j^l = f(\beta_j^l down\left(x_j^{l-1}\right) + b_j^l) \quad (2)$$

where, down () represents the function of downsampling.

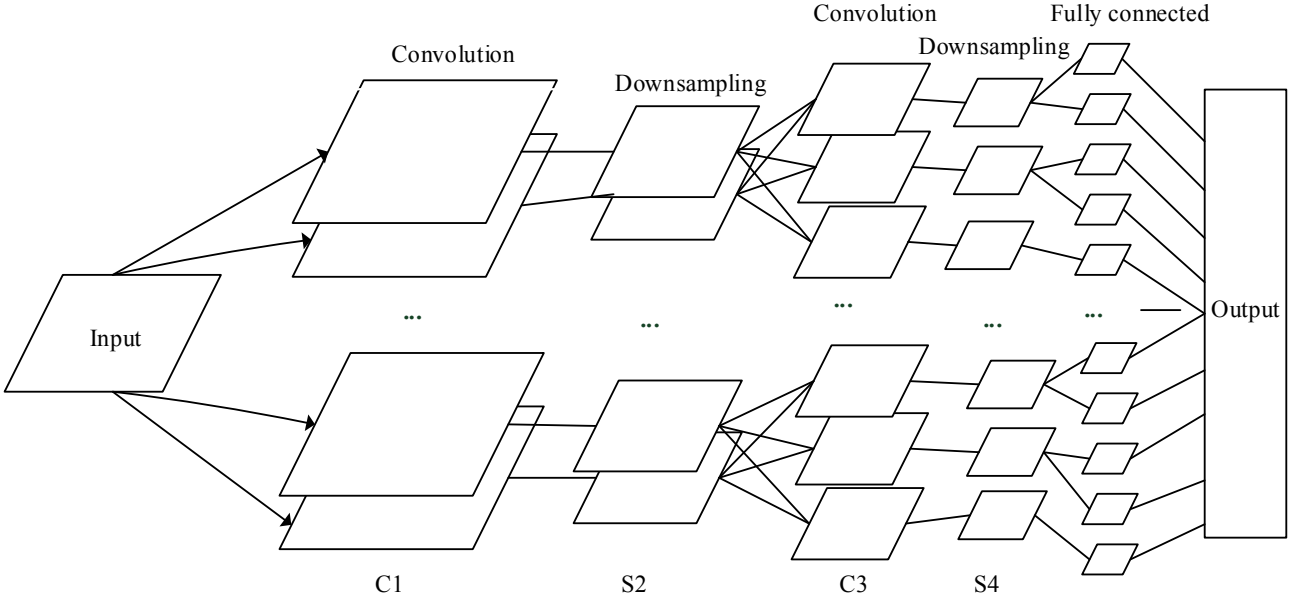

Fig. 1 The structure diagram of the convolutional neural network.

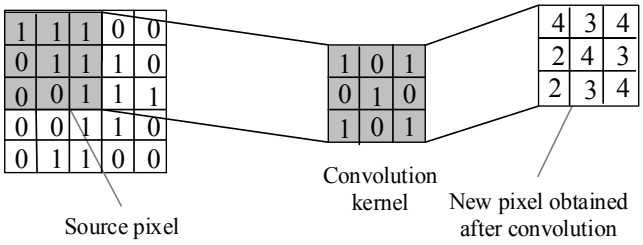

Fig. 2 Operation process of Convolution kernel.

The max-pooling process, the mean pooling process and the stochastic pooling process are shown in Fig. 3-5, respectively. The max-pooling process is to select the maximum value of the region's activation function for an extracted region to represent the characteristics of the region. The mean pooling process is to select the average of this region to describe the characteristics of this region. The stochastic pooling process is normalized according to the feature number of the pooling window, and the sampling probability is assigned according to normalized value, that is to say the probability that the element is large is also large.

D. Improved Pooling Method

In the practical application of face image recognition, the max-pooling method is mainly used because it can better preserve the texture information of the image. The mean pooling method can better preserve the background information of the image and the stochastic pooling method is between mean pooling and max-pooling methods. In the image recognition by using the max-pooling method, the remaining hidden information of the pooling window cannot be obtained and transmitted. Therefore, it is hoped to get the most useful information through the max-pooling, and not to make the network generalization drop caused by the too single and solidified maximum pooled function. So by combining the max-pooling method and the stochastic pooling method, an improved pooling method shown in Fig. 6 is proposed to make each pooled sampling area not only have better ability to store information, but also better improve the generalization of neural network. In order to highlight the information provided with maximum, which is more easily sampled by random sampling, its value can be amplified and each window value is squared. Elements with larger values are assigned to a higher sampling probability. In this way, not only the characteristics of the maximum sampling are well preserved, but also the possibility that the smaller implicit values are transmitted to the previous layer is improved, which effectively enhances the generalization ability and better feature extraction ability of the network. Firstly, the value of each sampling window x_i is squared and then it is normalized so as to get the corresponding sampling probability P_i:

$$P_i = \frac{x_i^2}{\sum_{k\in R} x_k^2} \tag{3}$$

$$\sum_{k\in R} P_i = 1 \tag{4}$$

After the discrete distribution probability set $P_i(i \in R)$ is obtained by calculation, γ is obtained by carrying out the randomly sampling on i according to this probability. So the pooled sample value s belonging to this the window can be defined as:

$$s = a_\gamma, \quad \gamma \sim P\{P_1, P_2, \ldots, P_R\} \tag{5}$$

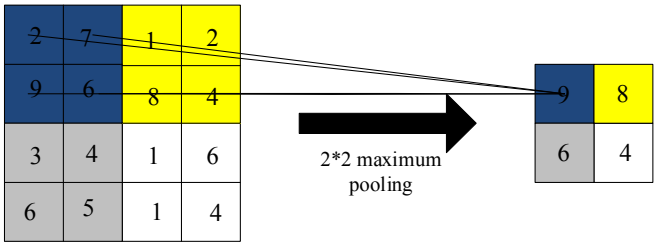

Fig. 3 Max-pooling.

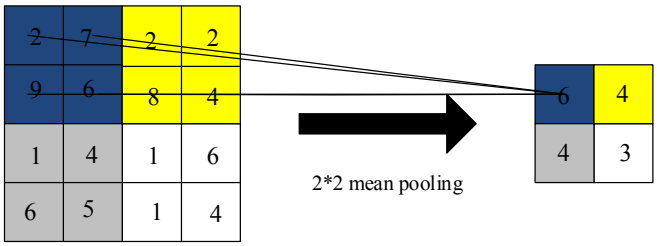

Fig. 4 Mean pooling.

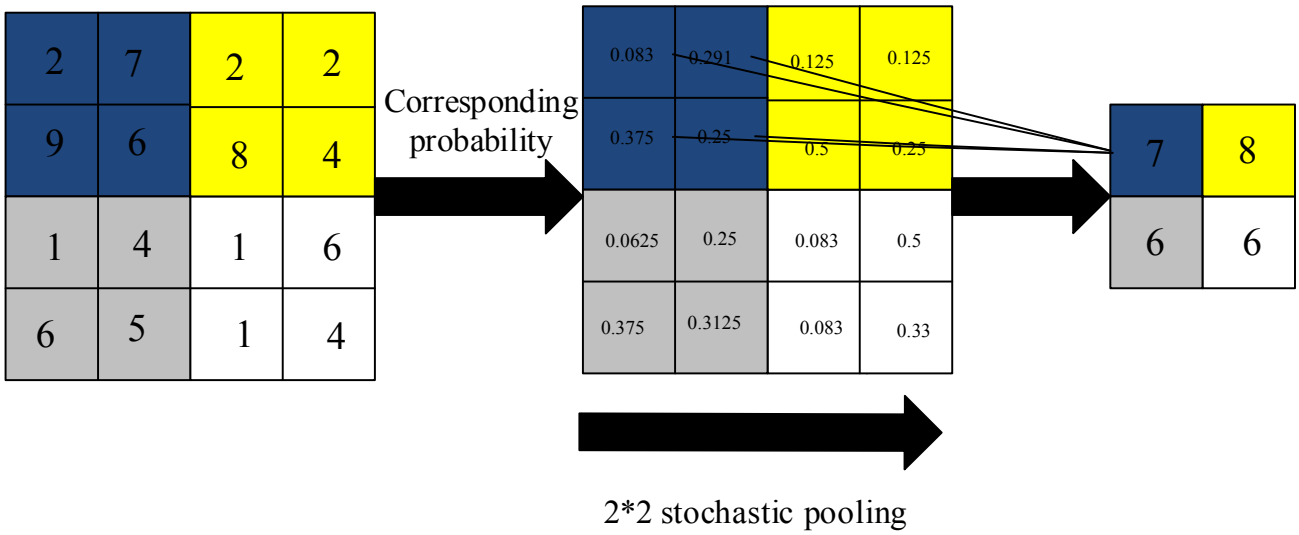

Fig. 5 Stochastic pooling.

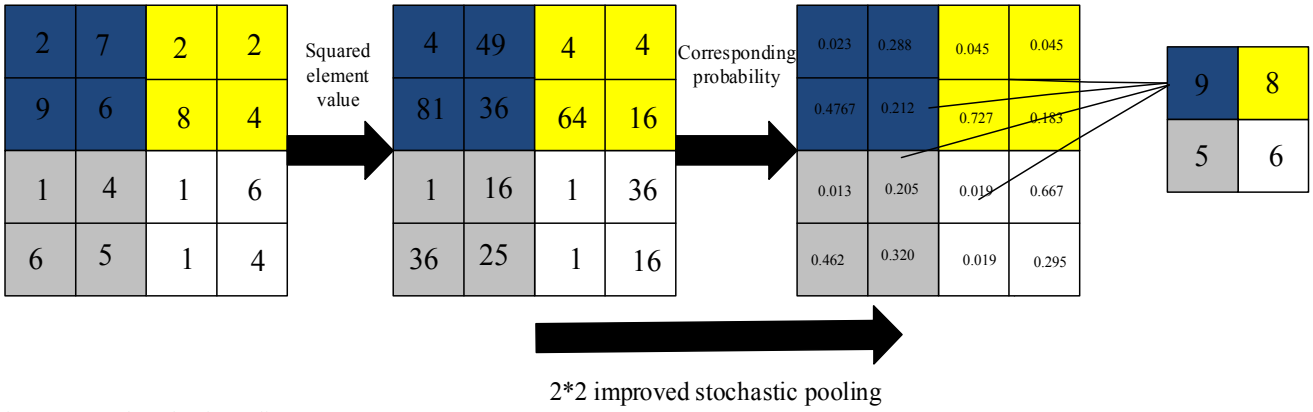

Fig. 6 Improved stochastic pooling.

The pooled structure obtained by this method is closer to the max-pooling method. However, it has stronger network generalization ability than the ordinary max-pooling, and has the ability that largest pooling describes characteristic objects than the ordinary stochastic sampling pooling method. This plays a better role in feature extraction and training of convolutional neural networks. In order to more specifically understand the application of different pooling methods in the sampling process, an example is adopted to illustrate the effects of different pooling methods, which is shown in Fig. 7 and Fig. 8. The above image is the effect of a 5*5 pooled window. The improved stochastic pooling effect and the max-pooling effect shown in Fig. 7 and Fig. 8 are the same, but in practical applications, it can be found that when the pooling window is increased, and on the basis that the improved stochastic pooling method and the max-pooling method are similar, more hidden information is extracted, which can enhance the generalization of the network structure and make the abstraction process of image features closer to reality.

E. Full Connection Layer

The fully connected layer is to connect all the neurons of the previous layer to the neurons of the next layer one by one. In general, the fully connected layer will appear after the convolutional and pooling layers and before its output layer. Its main function is to arrange the two-dimensional output images of the previous layer into a one-dimensional feature vector according to a certain rule. The aligned one-dimensional feature vector obtains the output for the fully connected layer l through the activation function:

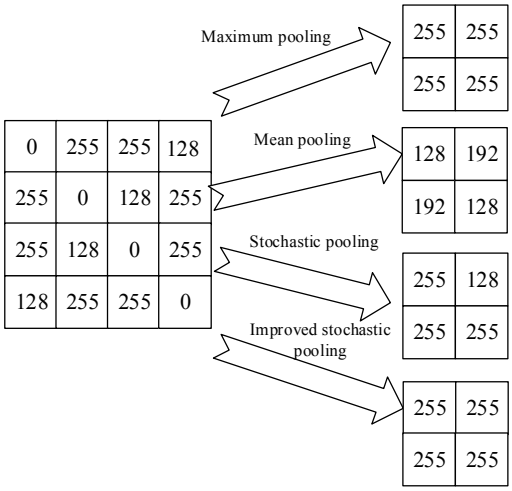

Fig. 7 Effect comparison of different pooling methods (a).

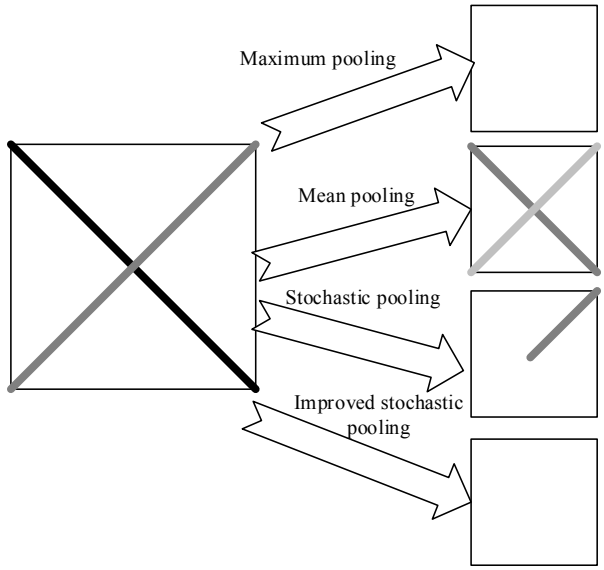

Fig. 8 Effect comparison of different pooling methods (b).

$$x^l = f(u^l) \tag{6}$$

$$u^l = w^l x^l + b^l \tag{7}$$

where, u^l is the net activation which is obtained by weighting and offsetting the output of the previous layer, w^l is the weight coefficient, and bl is the offset.

F. Softmax Classification

In the structure of a convolutional neural network, the Softmax classifier is usually used at the output layer to classify images, which is a multi-classifier that can perform classification tasks of type greater than 2. It is based on the derivation and evolution of the logistic regression model. If the Softmax classifier needs to classify features of a k (where k>2) classes, the set of quantities used for classification is m, and each sample can be thought of as an n-dimensional vector, then all training sets can be expressed by:

$$T = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})\} \tag{8}$$

where, $y^{(i)} \in \{1,2,\ldots,k\}$ is the label of the different samples, and $x^{(i)} \in R^n$ is the sample i of each type. Each sample is changed to the corresponding sample probability by Softmax:

$$P(y = j|x), (j = 1,2,\ldots,k) \tag{9}$$

By converting the sample into an k-dimensional probability vector, calculation of the Softmax classifier function is realized by:

$$f(x^{(i)}|\theta)=\begin{bmatrix}p(y^{(i)}=1|x^{(i)},\theta)\\p(y^{(i)}=2|x^{(i)},\theta)\\\cdots\\p(y^{(i)}=k|x^{(i)},\theta)\end{bmatrix}=\frac{1}{\sum_{j=1}^{k}e^{\theta_j^T x^{(i)}}}\begin{bmatrix}e^{\theta_1^T x^{(i)}}\\e^{\theta_2^T x^{(i)}}\\\cdots\\e^{\theta_k^T x^{(i)}}\end{bmatrix} \quad (10)$$

where $\theta=[\theta_1^T,\theta_2^T,...,\theta_k^T]$ is the learning parameter of the Softmax classifier. The optimal parameter θ is obtained by learning on the training set sample T and continuously iterative fitting the data set T. The minimum loss value of the loss function of the Softmax classifier can be obtained by:

$$J(\theta)=-\frac{1}{m}[\sum_{i=1}^{m}\sum_{j=1}^{k}1\{y^{(i)}=j\}log\frac{e^{\theta_j^T x^{(i)}}}{\sum_{j=1}^{k}e^{\theta_j^T x^{(i)}}}] \quad (11)$$

where, $1\{y=j\}$ indicates that when $y\neq j$, the probability value is taken as 0, otherwise, the value is 1. It can be seen from Eq. (11) that the network is converging when the loss value is smaller, that is to say, Softmax has a good classification ability.

III. Training Process of Convolutional Neural Networks

The training process of convolutional neural networks can be seen as a mapping from input to output, which does not need to be described by detailed mathematical functions between input and output, and only requires a large amount of data for network training. The training process consists of two phases: the forward propagation training phase and the back propagation training phase. (1) Forward stage: input the a part of image information in the training set into the network, and identify the image through feature extraction of the network. (2) Reverse phase: Since the actual output of the network will definitely be inaccurate with the target value, it needs to re-adjust the partial derivative and weights of the parameters by the gradient descent method.

A. Forward Propagation Training Process

(1) Extract a sample X from the sample set, input it into the training network, and set the ideal output value as Y_P. (2) Calculate the actual output value Y_O. In this process, the input samples need to undergo multiple layers of transformation to reach the output layer. This process is mainly to convolve and pool the weight matrix of input data. It can be expressed as:

$$Y_O=F_n(...(F_2(F_1(X_1W^{(1)})W^{(2)})...)W^{(n)}) \quad (12)$$

The forward process shown in Fig. 9 can be expressed as:

$$picQ=conv(picP,filter,valid) \quad (13)$$

B. Back Propagation Training Process

(1) The error between the actual output value Y_O and the ideal output value Y_P is calculated. If the actual output value has a large error with the ideal output value, the reverse error adjustment process needs to be carried out. The error propagation process is shown in Fig. 10.

$$epicP=conv(epicQ,filterRot180,full) \quad (14)$$

(2) The weights of the matrix are adjusted by minimizing the error method, and then the adjusted information is used to perform the forward propagation process. The weights of the matrix are adjusted by error propagation, which is shown in Eq. (15) and Fig. 11.

$$filterD=conv(picP,Rot180,epicQ,valid) \quad (15)$$

C. Network Training Procedure

Before the network training, the input images need to be preprocessed so that the convolutional neural network can extract the feature information better. Then all the weights are initialized. CNN may cause the network to enter the saturation state in advance because the weights are too large. Therefore, for the initialization of weights, the random number method is adopted to perform initialization operations to ensure that the network can operate normally. Then the training samples pass through multiple convolutional layers and pooling layers to extract valid feature information, then all the feature information enter the output layer through the full connection to enhance the data, and the Softmax classifier is used to perform the classification operation. Finally, it is compared with the set output. If it is met, the output result is obtained, otherwise a back-propagation process is performed, that is to say the error and the weight are propagated layer by layer, and the corresponding weight parameters are modified and a new round of network training is performed again until the correct output value is obtained. Its flowchart is shown in Fig. 12.

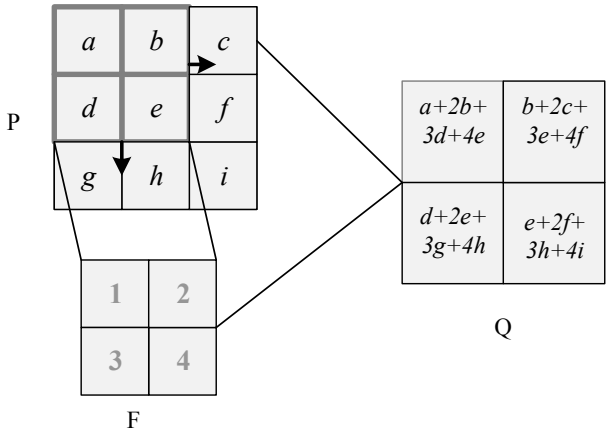

Fig. 9 Forward propagation.

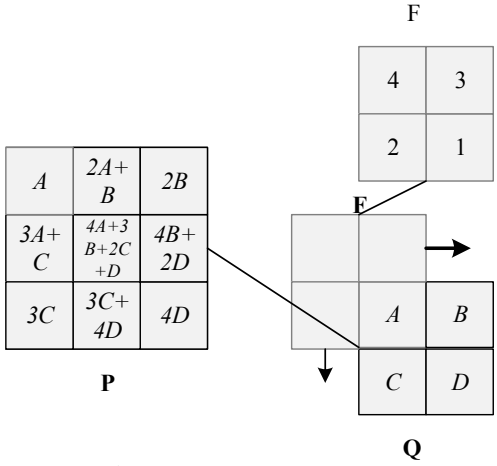

Fig. 10 Error propagation.

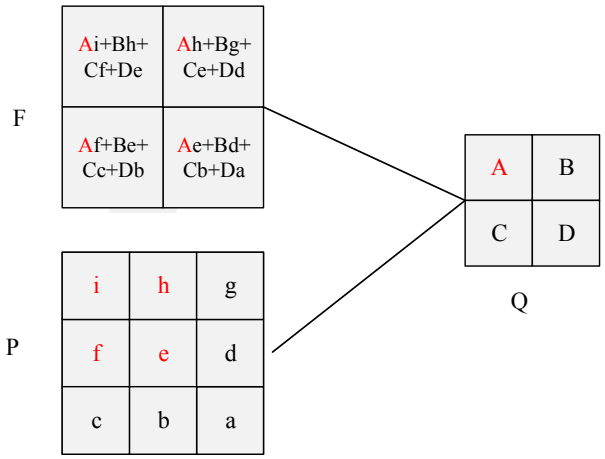

Fig. 11 Weight modification.

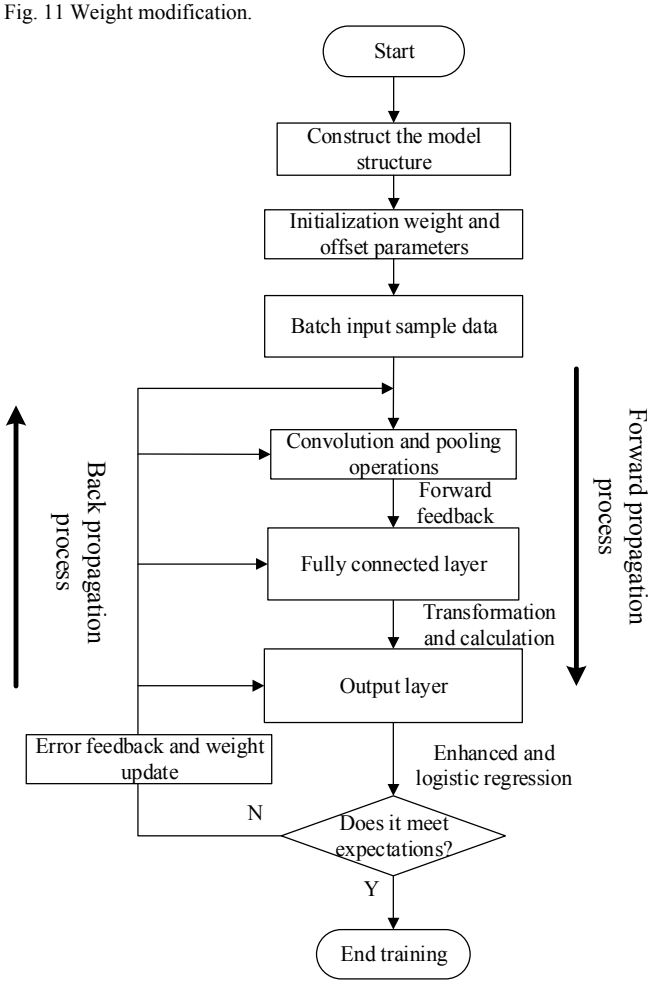

Fig. 12 Training flowchart of convolutional neural network.

IV. Face Detection and Recognition Method Based on Convolutional Neural Network with Improved Pooling Layer

A. Data Preprocessing

CNN is a layer-by-layer learning process that input data directly into an output in an end-to-end manner. Moreover, it is a learning process that requires a large amount of data to be trained and self-learned through the network to achieve the desired result. Therefore, when network training experiments are performed on a standard database, the experiments are carried out with the data set as input by performing a simple image prepossessing operation on the standard database, such as image normalization operation, image de-average operation.

(1) Image Size Normalization

CNN has a limited input on the image size when it trains the network, so the size of the images need to be normalized according to different CNN. Image normalization refers to the process that the image changes to a standard form as input through some operations. This standard form images have constant characteristics for pseudo-radiation changes such as rotation, scaling, and translation. The pseudo-radiation transformation can be understood as multiplying the original coordinates by different matrices to achieve different linear transformations in a linear space to achieve some typical pseudo-radiation transformation of the image. Assuming that the pseudo-radiation transformation matrix is a 3*3 matrix, the original coordinates (x,y) of transformation matrix are transformed into new coordinates $(x^{'},y^{'})$, which is shown as:

$$\begin{bmatrix} x^{'} \\ y^{'} \\ 1 \end{bmatrix} = \begin{bmatrix} m_{00} & m_{01} & m_{02} \\ m_{10} & m_{21} & m_{22} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{16}$$

The principle of matrix-based image size normalization is to change the function parameters by the non-changed characteristics of image pseudo-radiation. Then the original image is tackled by the changed parameters function to obtain a standard image. Image size normalization can improve training speed and efficiency.

(2) Image De-average Operator

During the training of a convolutional neural network, one of the operations that is often performed is to de-average the image, which is to calculate the data of each dimension of all the images first, and then center them to a value of 0, that is to say, an operation of calculating the average value of the image information, and then subtracting the mean value from all the samples. The operation of the image de-average is mainly to standardize all the image information, that is to standardize the data characteristics. Data feature standardization is to make each dimension of each data have unit variance and zero mean. This operation is to perform the mean calculation on each dimension for all data on all data sets, and then divide standard deviation of of each dimension of all data of the data set on each dimension, whose function is mainly to remove the average brightness of the image information because the brightness of each image is not needed when performing feature recognition. On the other hand, the de-average operation of the image will affect the back-propagation efficiency of the network to a certain extent, which is shown in Eq. (17).

$$\frac{\partial E}{\partial w_{11}^{(2)}} = x_1 \delta_1^{(2)} \tag{17}$$

B. Image Data Enhancement Based on Gabor Wavelet Transform

The quality of data sets training CNN directly affects the results of network training. In some cases, the training results may be affected due to fewer training sets or low quality training images. For example, fewer data volumes in the

training set can cause the network to become over-fitting and result in poor network generalization capabilities. The low quality of the training set will result in poor network training results. Therefore, in the training process of the network, the number of data samples should be increased and the quality of the trained data samples should be improved to achieve better network performance and avoid over-fitting. This approach can also increase the generalization capabilities of the network. So some pre-processing operations need be performed on the sample images to improve the quality of the training set according to different situations, such as complex illumination, posture, blurred biological information, and small amount of data. The Gabor wavelet transform is actually a variant of the wavelet transform. In fact, the basis function of the wavelet transform is replaced by the Gabor function. Because the Gabor function has the characteristics of frequency, it can perform more detailed frequency analysis on the image. It is also because of this characteristic that different scale facial information of the image can be obtained through Gabor wavelet transform. By filtering the noises of pixel set by using the Gabor wavelet transform, the underlying data distribution can be better obtained. The feature information $J_2(W_i)$ of the output image obtained after wavelet denoising can be described as follows.

$$J_2(W_i) = \sum_{r=1}^{t} \sum_{q=1}^{k_2} \left\| W_i^T x_{ir} - W_i^T x_{irq} \right\|^2 B_{irq} \quad (18)$$

$$= tr(W_i^T [\sum_{r=1}^{t} \sum_{q=1}^{k_2} (x_{ir} - x_{irq})(x_{ir} - x_{irq})^T B_{irq}] W_i) \quad (19)$$

$$= tr(W_i^T H_2 W_i) \quad (20)$$

where:

$$H_2 = \sum_{r=1}^{t} \sum_{q=1}^{k_2} (x_{ir} - x_{irq})(x_{ir} - x_{irq})^T B_{irq} \quad (21)$$

Eq. (21) represents the Gabor wavelet transform scale information. After deriving H_1 and H_2, the geometric distribution can be projected through the information of the face, and W_i is obtained shown in Eq. (22).

$$W_i = (H_1 - H_2)\omega = \gamma\omega \quad (22)$$

Suppose $\{w_1, w_2, ..., w_{d_i}\}$ is d_i eigenvalue in a linear subspace, which is distributed according to the density $\{\lambda_1 \geq \lambda_2 \geq ,..., \geq \lambda_{d_i}\}$ of the feature distribution. The feature vector corresponding to $\{\lambda_j | j = 1,2,...,d\}$ is a feature of the diversity of the human body posture, and the value of d_i is determined by different distance values $\{\lambda_1 \geq \lambda_2 \geq ,..., \geq \lambda_{d_i} > 0 \geq \lambda_{d_{i+1}} \geq \lambda_{d_{i+2}} \geq ,..., \geq \lambda_d\}$ of the feature vectors of the images. $W = \{w_1, w_2, ..., w_{d_i}\}$ is a pseudo-radiation invariant matrix. The weight vector $W_1, W_2, ..., W_N$ of N projections can be obtained by wavelet denoising so that a better quality face image can be obtained.

Five different frequency filters are selected for processing, namely: 1, $\frac{1}{\sqrt{2}}$, $\frac{1}{2}$, $\frac{1}{2\sqrt{2}}$, $\frac{1}{4}$. The eight directions selected are: $\frac{\pi}{4}$, $\frac{3\pi}{8}$, $\frac{\pi}{2}$, $\frac{5\pi}{8}$, $\frac{3\pi}{4}$, $\frac{7\pi}{8}$, 0, $\frac{\pi}{8}$. The simulation results are shown in Fig. 13-18.

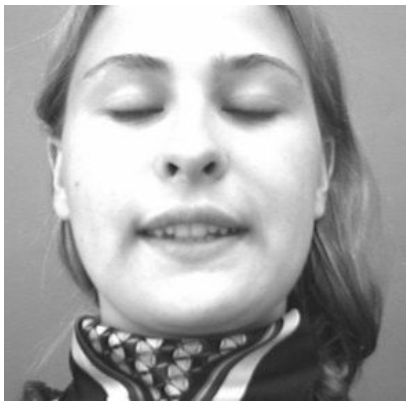

Fig. 13 Face images.

Forty face images with feature information can be obtained by the above method. This method is used to enhance the image quality information, and the number of trained samples is increased, the data set is enhanced, and the network training is better completed. According to the category of the network input data, if the input data is three channels, the grayscale image needs to be changed into an RGB image as input by adopting the software OpenCV.

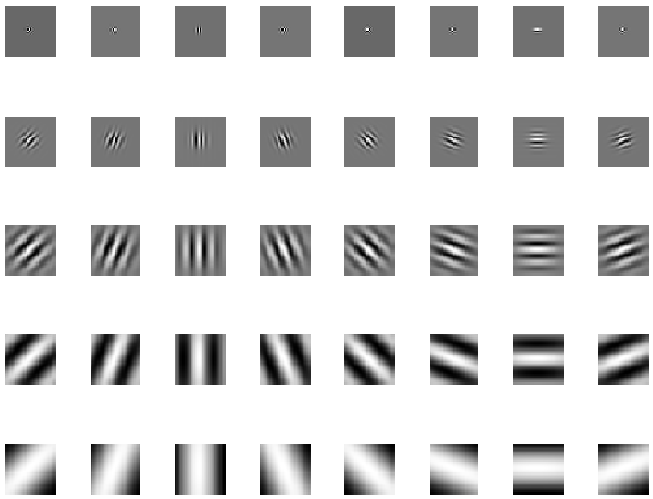

Fig. 14 The real part of Gabor wavelet filter.

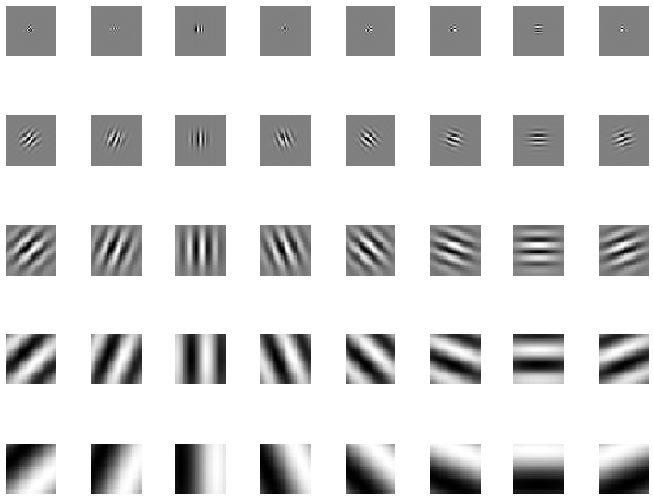

Fig. 15 The imaginary part of Gabor wavelet filter.

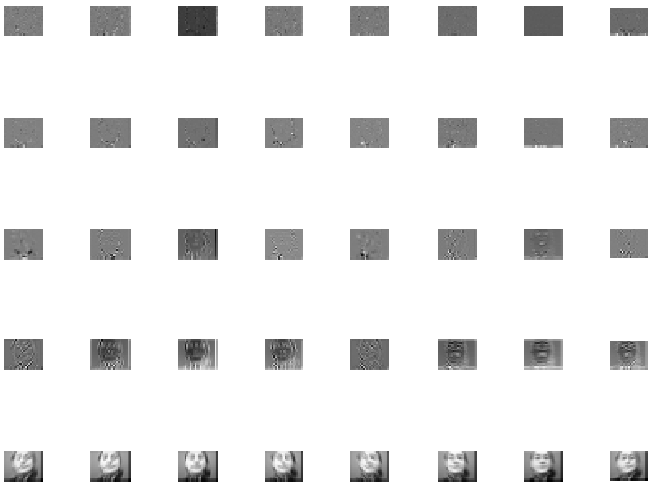

Fig. 16 Real part feature face.

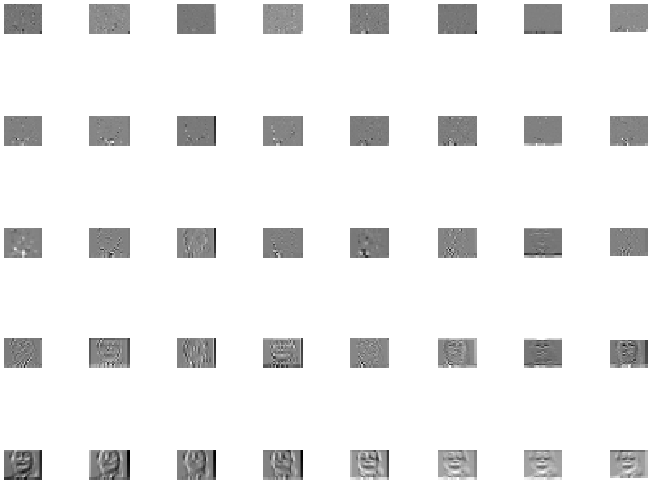

Fig. 17 Imaginary feature face.

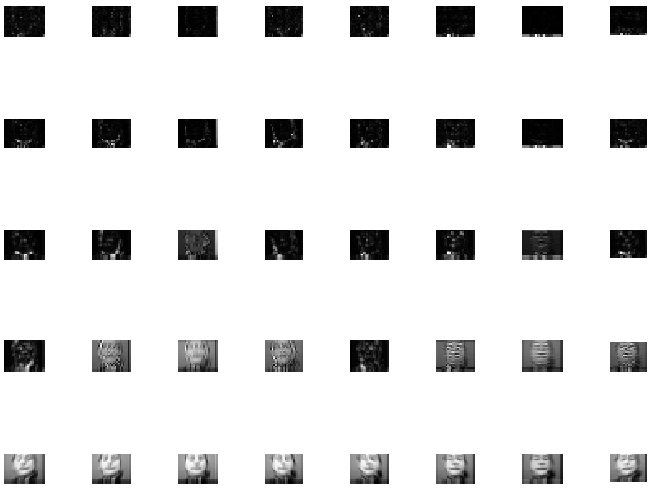

Fig. 18 Combination of real part and virtual part.

C. Standard Face Database

For the face recognition based on CNN, the simulation experiment need the authentication database. Here a standard database is adopted to carry out the related experiments. When performing face recognition by using a convolutional network, a large amount of face information is required for network training. Therefore, it is often necessary to provide face information through the face database for operation. There are many common face databases in the face recognition field, such as ORL face database, YALE face database, MIT face database, AR face database, FERET face database, CMU-PIE face database, LFW face database, Casia-Web face database, Face-Scrub database, etc. Taking into account the experimental situation and hardware equipment and other factors, the LFW face database is adopted for carrying out the experiments. Labeled faces in the wild (LFW) is a commonly used deep learning database, which is mainly a face database that is photographed in an unconstrained natural scene. The LFW database is mainly collected from more than 13,000 photos of more than 5,000 world-renowned people collected through the Internet. Their photo information is collected in different scenes, different illuminations, and different postures. Among them, 1680 people have 2 or more different photos and each face photo has a specific person name tag information.

D. Faster R-CNN Network

Faster RCNN adopts the region proposal network (RPN) as a frame recommendation, gives a frame that may be a face, and combines the CNN network to achieve target detection. It is also the application of the RPN on the candidate frame that solves the problem that the detection network is inefficient due to the time-consuming selection of the regional candidate frame. RPN is also a kind of convolutional neural network structure, which can be regarded as a full convolutional neural network. The input of each layer of RPN is an arbitrary size feature image of the previous layer network and the output is a candidate frame for multiple target areas of different sizes. The generated candidate regions can be moved through the sliding window on the feature image output by the shared convolutional network. Its main structure is shown in Fig. 19.

As shown in Fig. 19, convolutional layer 1 to ReLU5 adopt general structure of the Zeilerhe and Fergus model, which contains 5 convolutional layers, 2 pooling layers, and 2 normalized layers, and adopts the function ReLU as the activation function. The normalization layer is local response normalization (LRN), which divides each input value by $(k + a/n \sum_i x_i^2)^\beta$, where a is the scaling factor and the default value is generally set to 1, β is the number phase and the default value is set to 5, n is the local size. The next structure is the recommendation layer with an n*n convolutional layer and the candidate frame regression layer with a 1*1 convolutional layer, which is mainly for calculating the deviation of the selected frame from the real frame, and the other 1*1 convolution layer is for scoring the selected frame. The candidate frame probability layer is a Softmax regression layer, which is mainly to calculate the probability of the target frame.

After the candidate region is obtained through the RPN operation, the obtained region is subjected to pyramid pooling operation, the candidate region is mapped to the next convolution model, the deep features of the corresponding region are extracted to avoid repeated input of different regions, and target category prediction and boundary frame selection are performed on the target feature area. The structure of the Faster R-CNN network is shown in Fig. 20.

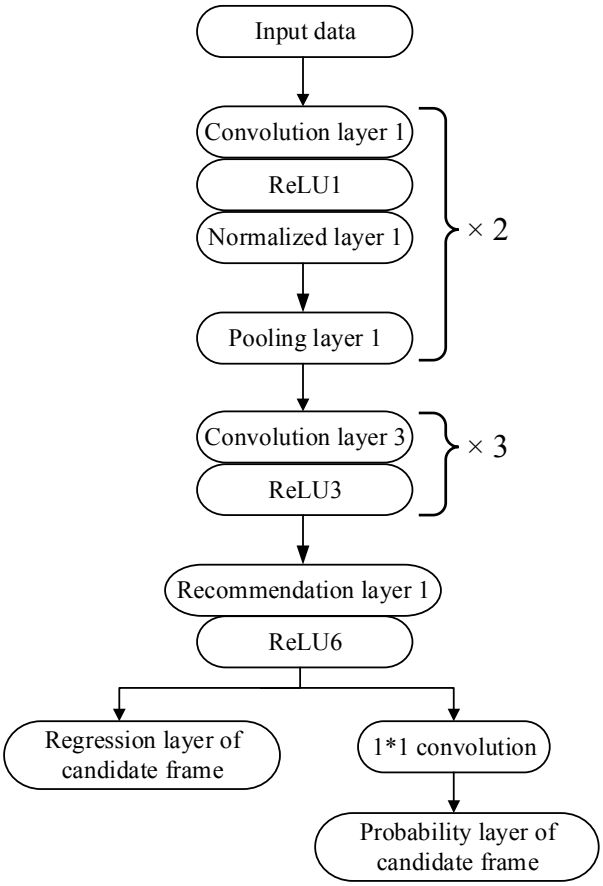

Fig. 19 RPN structure diagram.

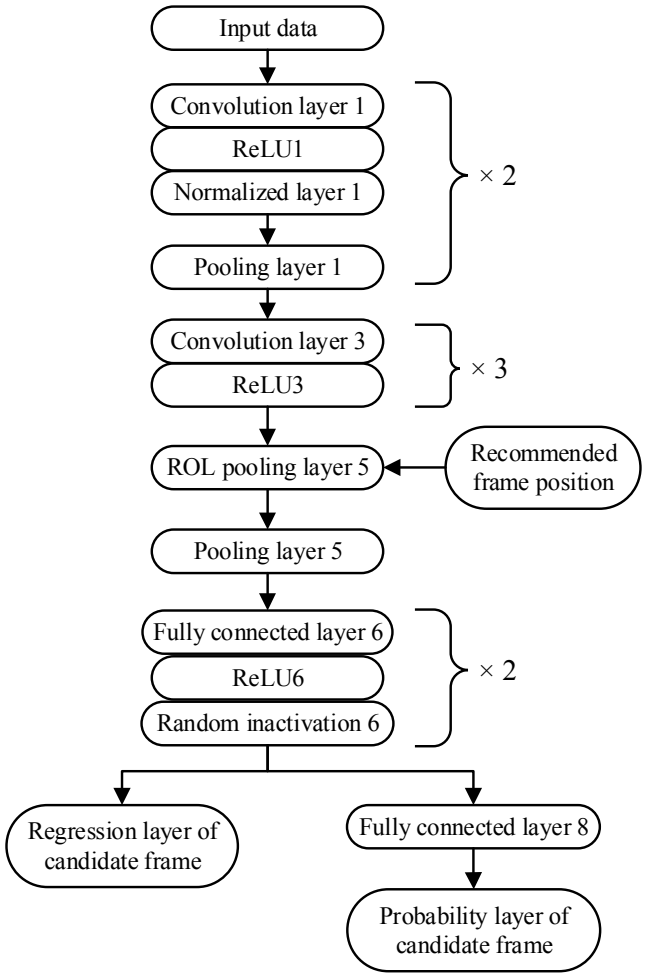

Fig. 20 Faster R-CNN structure diagram.

As shown in Fig. 20, the face images enter the network as an input, feature images are generated after going through from the convolution layer 1 to ReLU3. The recommended frame of the target prediction is obtained through the operation of the RPN network, and the feature images are selected as the input according to the location of the recommended frame to enter the ROI pooling layer. The ROI pooling layer changes the feature image area of the possible target into a fixed-size pooled feature image by the max-pooling operation. However, there is a big difference between the ROI pooling and the max-pooling. The max-pooling uses a fixed pooling core to perform the pooling operation and obtains a feature image with a fixed size, but the ROI pooling uses a pooled kernel of a non-fixed size to extract feature images of different sizes, and then uses max-pooling to fix the output size of the feature images. After going through the ROI pooling layer, it enters the portion of the fully connected layer 6.

In this part, the random inactivation operation is applied, whose principle is that one half of the neurons are randomly deactivated on the basis of the fully connected layer, that is to say that this part does not perform weight calculation and weight update. Since the network of random inactivation for each training is not fixed, the structure of each trained network is different.

The benefits of this operation not only reduce the amount of computation by half, but also increase the generalization of the network, reduce the possibility of over-fitting and enhance the robustness of the network. After going through the fully connected layer 7, the candidate frame output layer and the fully connected layer 8 of the two sub-networks are entered. The difference between the candidate frame output layer and the output candidate frame is compared and the score is calculated to increase the accuracy of the candidate frame. Then,the candidate frame probability layer is entered, and the probability of using the candidate frame as the target frame is calculated by performing a Softmax regression operation. After passing through the network, the position and category of the obtained output candidate area are compared and calculated with the position and category of the actual area. The loss function is obtained by the chain derivative calculation. The weight function is improved on each layer of the network by the cost function, and then the next iteration is proceeded. The loss function is a joint representation of the regression error and the classification error, which is described as follows:

$$\mathrm{L}(\{p_i\}\{t_i\}) = \frac{1}{N_{cls}}\sum_i L_{cls}\left(p_i,p_i^*\right) + \lambda\frac{1}{N_{reg}}\sum_i p_i^* L_{reg}\left(t_i,t_i^*\right) \quad (23)$$

where, i represents the i-th anchor point, $\mathrm{t_i^*}$ is the deviation between the real border and the candidate border, and $\mathrm{p_i^*}$ represents the positive sample of the i-th anchor point.

When using the network for face detection training, if the overlap ratio of the candidate area border to the real border is less than 0.3, the candidate frame is a negative sample. If the overlap ratio of the candidate area border to the real border is greater than 0.7, the candidate frame is defined as a positive sample. If it does not belong to the above two overlapping ratios, the candidate frame is no longer trained in the network.

$L_{cls}(p_i,p_i^*)$ is the network calculation classification confidence of the real border and the selected area. The probability calculated by the real classification is defined as:

$$L_{cls}(p_i,p_i^*) = -\log(p_u) \tag{24}$$

$L_{reg}(t_i,t_i^*)$ indicates the detection error obtained by detecting in the candidate region. By comparing the translational scaling error of the candidate region t^u with the actual region v, Eq. (25) can be obtained.

$$L_{reg}(t_i,t_i^*) = \sum_{i=1}^{4} g(t_i^u - v_i) \tag{25}$$

It can be concluded from Eq.(25) that g is the loss function of L1 which is not sensitive to outlines, which is shown as:

$$g(x) = \begin{cases} 0.5x^2, |x| < 1 \\ |x| - 0.5, |x| \geq 1 \end{cases} \tag{26}$$

E. Face Detection Experiments Based on Convolutional Neural Network

The faster R-CNN network is applied on the LFW face database and the face detection image through experiments are shown in Fig. 21. Through experiments, three kinds of experimental results may be got, such as a image with only one face image, one image which contains two or more than two face images, and result without detecting the face. The specific experimental results are described as follows. (1) Face detection of LFW face database. The data set image only detects a face image, as shown in Fig. 22. (2) The data set image detects two or more face images, as shown in Fig. 23. (3) The face image is not detected in the dataset image, and the experimental result is shown in Fig. 24. Experiments with the LFW face database are carried out and the following experimental results shown in Fig. 25 are obtained.

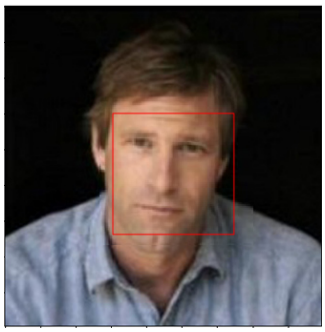

Fig. 21 Face detection experiment in LFW face database.

Fig. 22 Experimental results (1).

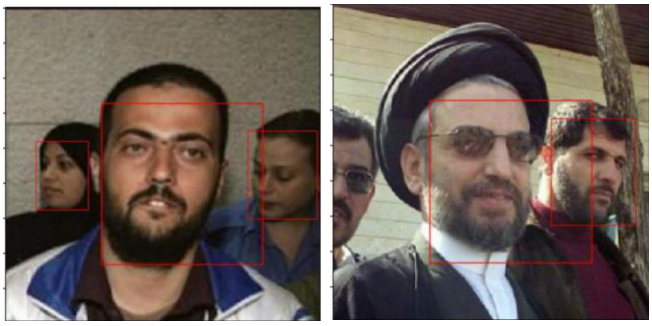

Fig. 23 Experimental results (2).

Fig. 24 Experimental results (3).

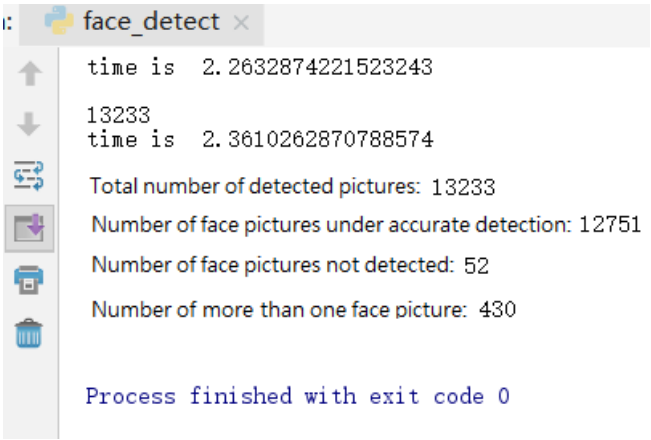

Fig. 25 LFW face database experimental data.

As shown in Fig. 25, experiment is carried out on a total of 13233 face images, and obtained 13989 face images. Among them, the number of images of only one face detected is 12,751, the number of images without detecting faces is 52, and the number of faces detected with more than one face is 430. All the detected face images are marked out by experiment, the background information which will interfere with the recognition is removed by the clipping, the face image containing only the face information is obtained, and a

total of 13989 face images are obtained.

F. Convolutional Neural Network for Face Recognition

(1) VGG Convolutional Neural Network

The VGG network is a new structure of the convolutional neural network proposed by K. Simonyan and A. Zisserman of Oxford university, which is a deeper network structure based on the LeNet and AlexNet network structures. VGG network has a good application of feature extraction, classification and recognition in image processing. The VGG convolutional neural network adopted in this paper has 11-19 layers different structures. The specific structure of VGG network is shown in Fig. 26 and its parameter configuration is shown in Table 1. The main components of the VGG-16 network model are described as follows. The convolutional layer of the network is a convolution kernel of size 3*3 with a step size of 1 and a same way filled with 1. Each convolution module consists of 2-3 convolutional layers. A total of five convolutional modules exists, each of which is connected to a pooled layer of size 2*2 pooled core with a step size of 1and max-pooling layer filled with 2. The number of convolution kernels of the five convolution modules is the same, but as the number of layers deepens, the number of convolution kernels also increases. The number of convolution kernels corresponding to each module is 64, 128, 256, 512, and 512, respectively. The max-pooling layer of the network structure is connected to a module of three layers of fully connected layers, and the number of neurons in the fully connected layer is 4096, 4096, and 1000, respectively. The Softmax function, the multi-classifier, is then used as the output layer for the final classification. The VGG-16 network uses a continuous stack of convolutional layers to form a convolutional module. This stacking method can effectively enhance the extraction capability. A 3*3 sized convolution kernel is the smallest convolution size that a convolution operation can operate on, but the convolution module can be equated to a larger convolution kernel by a continuous stack operation. The convolution calculation of the stack of two 3*3 convolution kernels is equivalent to a 5*5 convolution kernel operation. The convolution calculation of the stack of three consecutive 3*3 size convolution kernels is equivalent to a 7*7 convolution kernel with convolution operations. The convolution operation by this stacking method is not only more nonlinear than the large-scale convolution, but also uses less operations and has better feature extraction capability. Adding the Dropout method after the first and second fully connected layers not only reduces the possibility of over-fitting of the network, but also reduces the amount of computation and improves the learning efficiency of the network through the method of randomizing random neurons.

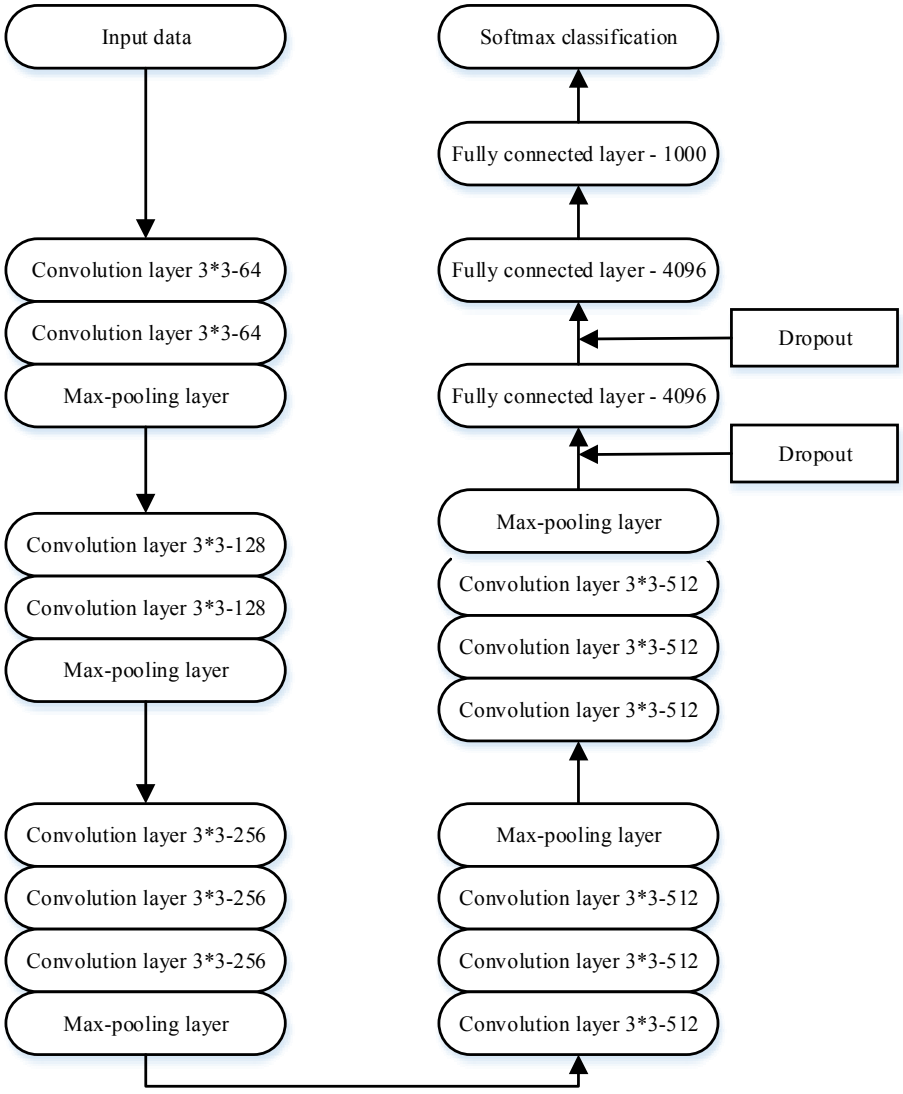

Fig. 26 VGG-16 network structure diagram.

TABLE. 1 SPECIFIC PARAMETERS OF VGG NETWORKS

Module	Layer name	Output value	Parameter
Input layer	Input	224*224 image	0
Convolution module 1	Convolution kernel 3*3 is 64	224*224*64 feature	（3*3*3+1）*64
	Convolution kernel 3*3 is 64	224*224*64 feature	（3*3*64+1）*64
Pooling layer	Max-pooling core 2*2	112*112*64 feature	0
Convolution module 2	Convolution kernel 3*3 is 128	112*112*128 feature	（3*3*64+1）*128
	Convolution kernel 3*3 is 128	112*112*128 feature	（3*3*128+1）*128
Pooling layer	Max-pooling core 2*2	56*56*128 feature	0
Convolution module 3	Convolution kernel 3*3 is 256	56*56*256 feature	（3*3*128+1）*256
	Convolution kernel 3*3 is 256	56*56*256 feature	（3*3*256+1）*256
	Convolution kernel 3*3 is 256	56*56*256 feature	（3*3*256+1）*256
Pooling layer	Max-pooling core 2*2	28*28*256 feature	0
Convolution module 4	Convolution kernel 3*3 is 512	28*28*512 feature	（3*3*256+1）*512
	Convolution kernel 3*3 is 512	28*28*512 feature	（3*3*512+1）*512
	Convolution kernel 3*3 is 512	28*28*512 feature	（3*3*512+1）*512
Pooling layer	Max-pooling core 2*2	14*14*512 feature	0
Convolution module 5	Convolution kernel 3*3 is 512	14*14*512 feature	（3*3*512+1）*512
	Convolution kernel 3*3 is 512	14*14*512 feature	（3*3*512+1）*512
	Convolution kernel 3*3 is 512	14*14*512 feature	（3*3*512+1）*512
Pooling layer	Max-pooling core 2*2	7*7*512 feature	0
Fully connected layer	Fully connected layer is 4096 dimension	1*1*4096 feature	（7*7*512+1）*4096
	Fully connected layer is 4096 dimension	1*1*4096 feature	4096*4096
	Fully connected layer is 1000 dimension	1*1*Number of output categories	1000*4096
Output layer	Softmax Classifier		

According to Tab. 1, the total training parameters can be calculated to be 138 million, of which the training parameters of the three-layer fully connected layer are 123 million, accounting for 89.6% of the total calculation. It can be seen that the fully connected layer is the most computationally intensive part of network training. For each image, there is such a large amount of calculation, so the VGG network structure need to be improved so as to reduce the amount of calculation and shorten the recognition time without greatly affecting the recognition efficiency.

(2) Improved VGG Convolutional Neural Network

When the convolutional neural network recognizes an image, feature extraction is performed layer by layer. Starting from the input layer of the image, pixel-level features such as dots and lines are extracted layer by layer, and the closer to the output layer, the more abstract the image features are extracted. It can be seen that the VGG network is very critical for the underlying feature extraction of the image. So the structure of the fully connected layer which is closed to the output layer can modified and the squared value stochastic pooling method which combines the max-pooling with the random value pooling is adopted to improve the structure of the VGG convolutional neural network.

Nowadays, in the study of deep learning, in order to reduce the amount of computation, changing the fully connected layer of the network structure has become the most important method to solve this problem. Both GoogLeNet and ResNet have abandoned the fully connected layer in the network structure. Through the calculation of the parameters of the previous section, it can be seen that the full connection layer in the VGG network causes a huge amount of network learning calculation. So the original VGG network connection layer is changed to reduce the amount of computation. The GoogLeNet network uses the global mean pooling method in the pooling layer before the fully connected layer, so this method is adopted to the VGG network, which not only reduces the training parameters of the network,bu t also does not have much impact on the integrity of the network. However, due to the reduction of the fully connected layer, in order to extract the image features better, the method of stochastic pooling of square values is proposed to get more rich image features. The specific structure of improved VGG network is shown in Fig. 27.

The specific improvement operation is described as follows. The first two full connection layers in the original three-layer fully connected layer are removed, and only the fully connected layer seen as the classification type in the original network structure is retained. The pooling method of the pooling layer after the convolution module 1-4 is changed from the max-pooling method to the square value stochastic pooling method, and the pooling kernel size is still 2*2. In the fifth convolution module, the 512 convolution kernels are increased to 600 convolution kernels, and the max-pooling method of the pooling layer after the fifth convolution module is changed to the global mean pooling method, and the pooled core size is 7*7 and the corresponding output value is 1*1*7. The specific parameters of the improved VGG network are shown in Table 2.

According to the parameter calculation of Table 2, it can be concluded that the total parameter which the improved VGG network need to calculate for a image is 0.168 billion, and the parameter is more than 8 times smaller than the 138 million parameter of the original VGG network. Although the number of convolution kernels of the last convolutional module is increased, the pooled kernel size of 2*2 is increased to 7*7 by using the global mean pooling method instead of the previous fully connected layer. By reducing the two fully connected layers and calculating the required parameters, it was found that the number of parameters was less than 8 times compared to the original VGG network structure. This improved method can effectively reduce the time of network training and increase the efficiency of network training without affecting the recognition accuracy.

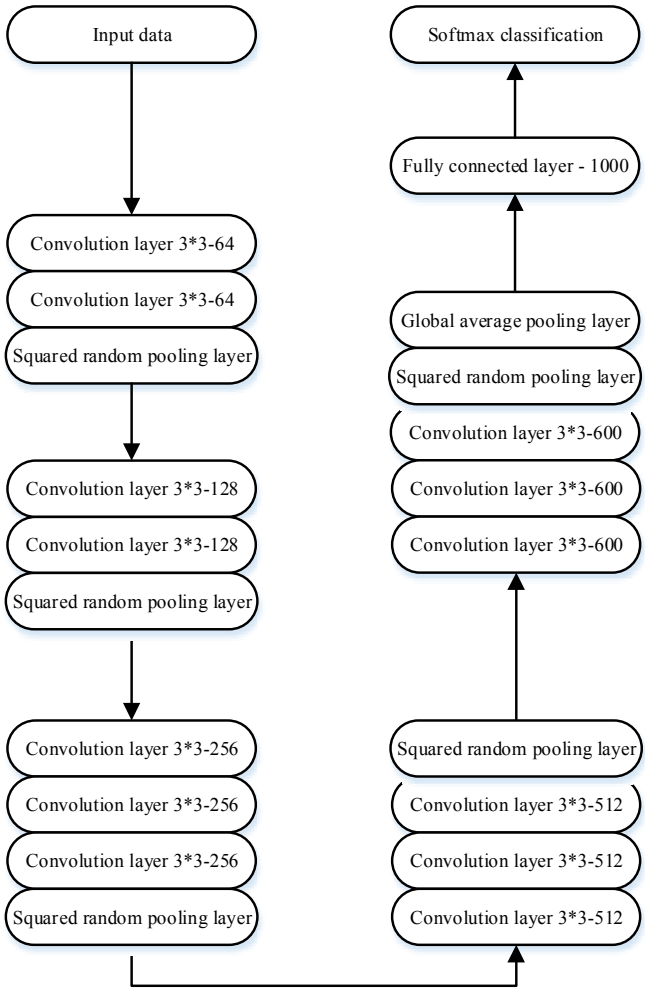

Fig. 27 Improved VGG network structure diagram.

TABLE. 2 SPECIFIC PARAMETERS OF IMPROVED VGG NETWORKS

Module	Layer name	Output value	Parameter
Input layer	Input	224*224 image	0
Convolution module 1	Convolution kernel 3*3 is 64	224*224*64 feature	（3*3*3+1）*64
	Convolution kernel 3*3 is 64	224*224*64 feature	（3*3*64+1）*64
Pooling layer	Squared random pooling kernel 2*2	112*112*64 feature	0
Convolution module 2	Convolution kernel 3*3 is 128	112*112*128 feature	（3*3*64+1）*128
	Convolution kernel 3*3 is 128	112*112*128 feature	（3*3*128+1）*128
Pooling layer	Squared random pooling kernel 2*2	56*56*128 feature	0
Convolution module 3	Convolution kernel 3*3 is 256	56*56*256 feature	（3*3*128+1）*256
	Convolution kernel 3*3 is 256	56*56*256 feature	（3*3*256+1）*256
	Convolution kernel 3*3 is 256	56*56*256 feature	（3*3*256+1）*256
Pooling layer	Squared random pooling kernel 2*2	28*28*256 feature	0
Convolution module 4	Convolution kernel 3*3 is 512	28*28*512 feature	（3*3*256+1）*512
	Convolution kernel 3*3 is 512	28*28*512 feature	（3*3*512+1）*512
	Convolution kernel 3*3 is 512	28*28*512 feature	（3*3*512+1）*512
Pooling layer	Squared random pooling kernel 2*2	14*14*512 feature	0
Convolution module5	Convolution kernel 3*3 is 600	14*14*600 feature	（3*3*512+1）*600
	Convolution kernel 3*3 is 600	14*14*600 feature	（3*3*600+1）*600
	Convolution kernel 3*3 is 600	14*14*600 feature	（3*3*600+1）*600
Pooling layer	Squared random pooling kernel 2*2	7*7*600 feature	0
Pooling layer	Global average pooling kernel 7*7	1*1*600 feature	0
Fully connected layer	Fully connected layer	1*1*Number of output categories	(2*2*600+1)*n
Output layer	Softmax Classifier		

G. Face Recognition Experiments Based on Convolutional Neural Networks

The experiment was carried out on the LFW face database. The face image recognition experiment will be carried out through the improved VGG convolutional neural network. The improved VGG convolutional neural network will be visualized and the feature effect of the random squared pooling layer of each module will be outputted. The comparison of some pooled layer renderings is shown in Fig. 28-32. It can be seen from the pooled layer feature image of visualization operation of the VGG network face image that the pooled layer of the improved network retains the characteristics of the original image well in feature extraction, and feature details that the max-pooling can't keep are increased. Through the 7*7 pooling checks global pooling instead of the full connection layer operation, the original feature size of 7*7 is changed to 1*1, which effectively reduces the amount of calculation for the output.

Fig. 28 The feature images output by the first pooling layer.

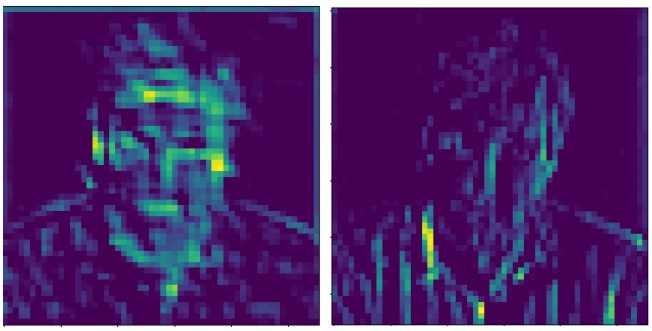

Fig. 29 The feature images output by the second pooling layer.

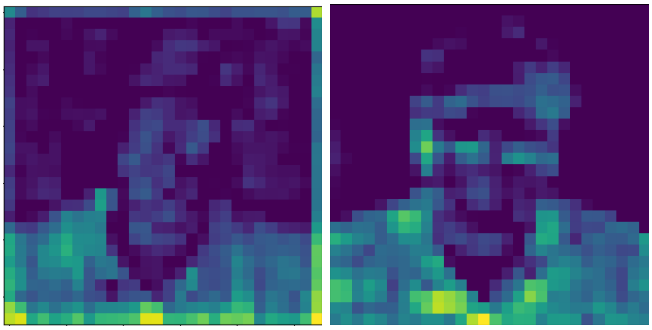

Fig. 30 The feature images output by the third pooling layer.

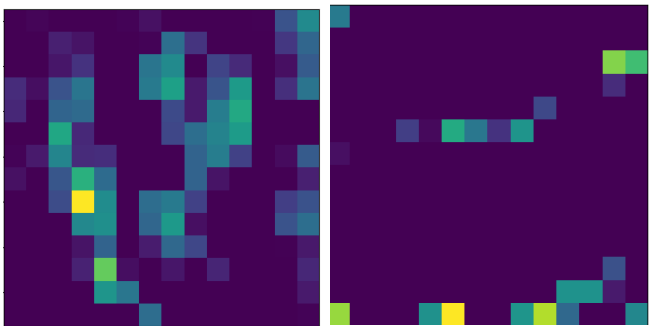

Fig. 31 The feature images output by the fourth pooling layer.

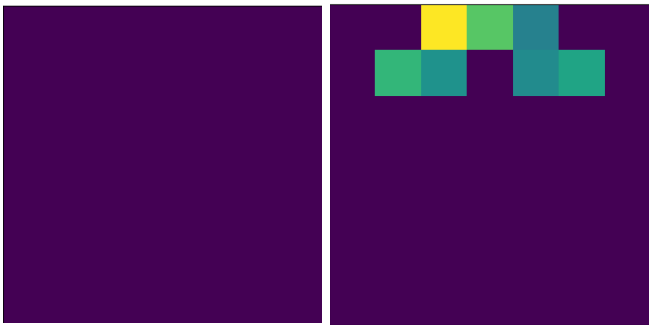

Fig. 32 The feature images output by the fifth pooling layer.

The parameters of the network training are set as follows. Dropout is 0.5, the weight attenuation coefficient is 5×10^{-5}, the center coefficient is 5×10^{-5}, the learning rate gradually decreases from the initial 0.1 to 10^{-5}, the weight initialization is set to a mean of 0, and the truncated Gaussian function with the standard deviation 0.1 is adopted. The experiment based on the proposed improved VGG network on the LFW face database is shown in Fig. 33-34. It can be seen from the above experiment results that the change law of the loss value is gradually reduced, it develops toward the convergence direction and finally stabilizes. After a certain number of iterations, the learning rate gradually decreases, and the change in the value of loss gradually becomes slower. It can be seen from Fig. 33 that the loss value also has little fluctuation, and gradually converges after 10,000 iterations. After 15,000 iterations, the convergence effect is gradually optimized, and the final loss value is about 0.03. As the number of sample iterations increases, the accuracy of face recognition in network training increases gradually, and finally stabilizes. The correct rate of recognition reaches 97.2%. And it only takes 10 hours, which is 5 times less than the initial 40 hours. Through experiments, it can be found that while reducing the full connection layer of the original VGG network, reducing the depth of network training and changing the sampling mode of the pooling layer, the recognition accuracy is not greatly reduced, and this method can effectively reduce network running time, reduce hardware requirements and improve network training efficiency.

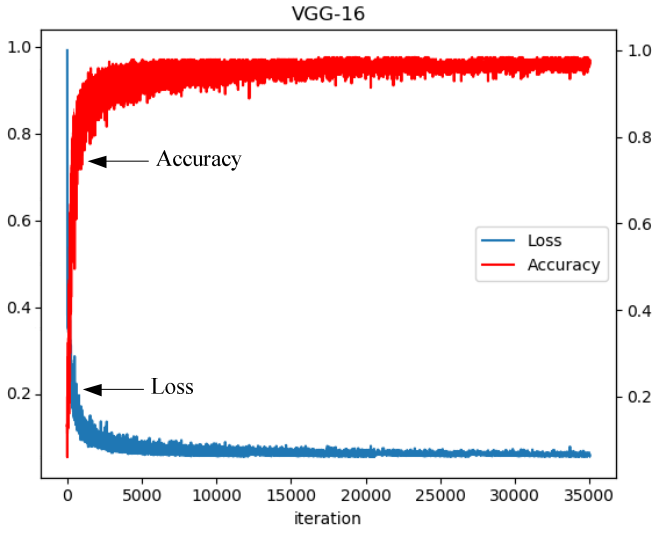

Fig. 33 The loss curve and accuracy of the improved VGG for face recognition.

```
cnn ×
Optimization Iteration:    34600, Training Accuracy:  97.4%
Optimization Iteration:    34700, Training Accuracy:  96.9%
Optimization Iteration:    34800, Training Accuracy:  96.9%
Optimization Iteration:    34900, Training Accuracy:  97.3%
Optimization Iteration:    35000, Training Accuracy:  97.1%
Time usage: 10:11:55
Accuracy: 97.2%
Process finished with exit code 0
```

Fig. 34 The face recognition test time of the improved VGG.

V. Conclusion

This paper firstly performs image size normalization and image de-averaging operation on the standard database. Random clipping is used to enlarge the amount of image data, the histogram homogenization is used to reduce the influence of the light shadow of the image, and the image denoising and enhancement operations are performed by the Gabor wavelet variation. LFW face database is used as the experimental

database. Then the Harr and Adaboost combination algorithm for traditional face detection is introduced in details. The Faster R-CNN network is used to conduct experiments on the LFW database. The VGG network was selected as the experimental network. Through the analysis and calculation of the structure and parameters of the VGG-16 network, the defects of the traditional VGG-16 network are found out. Then some improvements were made to the VGG-16 network. By changing the structure of the VGG-16 network and changing the pooling method of the pooling layer, a new VGG-16 network is obtained and its parameters are calculated. Experiments were carried out on the LFW face database by using this improved VGG-16 network, and good experimental results were obtained.

REFERENCES

[1] M. Ali Akber Dewan, E.Granger, G.-L. Marcialis, R. Sabourin, and F. Roli, "Adaptive Appearance Model Tracking for Still-to-video Face Recognition," *Pattern Recognition*, vol. 49, pp. 129-151, 2016.

[2] J. Neves, F. Narducci, S. Barra, and A. H. Proen, "Biometric Recognition in Surveillance Scenarios: a Survey," *Artificial Intelligence Review*, vol. 46, no. 4, pp. 515-541, 2016.

[3] Y. Xu, Z. Zhang, G. M. Lu, and J. Yang, "Approximately Symmetrical Face Images for Image Preprocessing in Face Recognition and Sparse Representation Based Classification," *Pattern Recognition*, vol. 54, pp. 68-82, 2016.

[4] Y. Lei, Y. Guo, M. Hayat, M. Bennamoun, and X. Zhou, "A Two-phase Weighted Collaborative Representation for 3D Partial Face Recognition with Single Sample," *Pattern Recognition*, vol. 52, pp. 218-237, 2016.

[5] L. Liu, P. Fieguth, G. Y. Zhao, M. Pietikäinen, D. W. Hu, "Extended Local Binary Patterns for Face Recognition," *Information Sciences*, vol. 358-359, pp. 56-72, 2016.

[6] T. Valentine, M. B. Lewis, and P. J. Hills, "Face-space: A Unifying Concept in Face-recognition Research," *Quarterly Journal of Experimental Psychology*, vol. 69, no. 10, pp. 1996-2019, 2016.

[7] X. Yin, and X. M. Liu, "Multi-task Convolutional Neural Network for Pose-invariant Face Recognition," *IEEE Transactions on Image Processing*, vol. 27, no. 2, pp. 964-975, 2018.

[8] S. Bashbaghi, E. Granger, R. Sabourin, and G. A. Bilodeau, "Dynamic Ensembles of Exemplar-SVMs for Still-to-video Face Recognition," *Pattern Recognition,* vol. 69, pp. 61-81, 2017.

[9] H. Li, and C. Y. Suen, "Robust Face Recognition Based on Dynamic Rank Representation," *Pattern Recognition*, vol. 60, pp. 13-24, 2016.

[10] M. Haghighat, M. Abdel-Mottaleb, and W. Alhalabi, "Fully Automatic Face Normalization and Single Sample Face Recognition in Unconstrained Environments," *Expert Systems with Applications,* vol. 47, pp. 23-34, 2016.

[11] X. H. Chen, J. S. Wang, Y. L. Ruan, and S. Z. Gao, "An Improved Iris Recognition Method Based on Discrete Cosine Transform and Gabor Wavelet Transform Algorithm," *Engineering Letters*, vol. 27, no. 4, pp. 676-685, 2019.

[12] J. S. Wang, Y. L. Ruan, B. W. Zheng, and S. Z. Gao, "Face Recognition Method Based on Improved Gabor Wavelet Transform Algorithm," *IAENG International Journal of Computer Science*, vol. 46, no. 1, pp. 12-24, 2019.

[13] M. Takalkar, M. Xu, Q. Wu, and Z. Chaczko, "A Survey: Facial Micro-expression Recognition," *Multimedia Tools and Applications*, vol. 77, no. 15, pp. 1-25, 2018.

[14] P. Sukhija, S. Behal, and P. Singh, "Face Recognition System Using Genetic Algorithm," *Procedia Computer Science*, vol. 85, pp. 410-417, 2016.

[15] Z. P. Hu, F. Bai, M. Wang, and Z. Sun, "Regularized Robust Sparse Representation Face Recognition Algorithm Based on Supervised Low-rank Subspace Recovery," *Journal of Signal Processing*, vol. 32, no. 11, pp. 1299-1307, 2016.

[16] Z. Lu, and L. Zhang, "Face Recognition Algorithm Based on Discriminative Dictionary Learning and Sparse Representation," *Neurocomputing*, vol. 174, pp. 749-755, 2016.

[17] J. X. Mi, and T. Liu, "Multi-step Linear Representation-based Classification for Face Recognition," *IET Computer Vision*, vol. 10, no. 8, pp. 836-841, 2016.

[18] G. Zhang, W. Zou, X. Zhang, X. Hu, and Y. Zhao, "Singular Value Decomposition Based Sample Diversity and Adaptive Weighted Fusion for Face Recognition," *Digital Signal Processing*, vol. 62, pp. 150-156, 2017.

[19] G. Feng, J. S. Zhang, H. J. Li, and J. W. Dong, "Face Recognition Based on Volterra Kernels Direct Discriminant Analysis and Effective Feature Classification," *Information Sciences: An International Journal*, vol. 441, pp. 187-197, 2018.

[20] A. Khatami, M. Babaie, H. R. Tizhoosh, A. Khosravi, T. Nguyen, and S. Nahavandi, "A Sequential Search-space Shrinking Using CNN Transfer Learning and a Radon Projection Pool for Medical Image Retrieval," *Expert Systems with Applications*, vol. 100, pp. 224-233, 2018.

Cheng Xing received her B. Sc. degree in automation from University of Science and Technology Liaoning, China in 2011, and her M. Sc. degree in computer science and engineering from Hanyang University, Korea in 2013. She is a Ph.D candidate in the School of Electronic and Information Engineering, University of Science and Technology Liaoning. Her main research interest is modeling of complex industry process.

Jie-Sheng Wang received his B. Sc. And M. Sc. degrees in control science from University of Science and Technology Liaoning, China in 1999 and 2002, respectively, and his Ph. D. degree in control science from Dalian University of Technology, China in 2006. He is currently a professor and Master's Supervisor in School of Electronic and Information Engineering, University of Science and Technology Liaoning. His main research interest is intelligent control and Computer integrated manufacturing.

Bo-Wen Zheng is with the School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, 114051, PR China. His main research interest is modeling methods of complex process and intelligent optimization algorithms.