Personal and Ubiquitous Computing (2019) 23:531–539
https://doi.org/10.1007/s00779-019-01238-9

ORIGINAL ARTICLE

Facial expression recognition method based on deep convolutional neural network combined with improved LBP features

Fanzhi Kong[1]

Received: 29 March 2019 / Accepted: 20 May 2019 / Published online: 8 June 2019

Abstract
Aiming at the disadvantages of the traditional machine-based facial expression recognition method that eliminates the feature of manual selection, a feature extraction method based on deep convolutional neural network to learn expression features is proposed. Since the deep convolutional neural network can directly use the original image as the input image, the image abstract feature interpretation is obtained at the fully connected layer of the image, which avoids the inherent error of image preprocessing and artificial selection features. Then, we reconstruct the traditional local binary pattern (LBP) feature operator for facial expression image and fuse the abstract facial expression features learned by the deep convolution neural network with the modified LBP facial expression texture features in the full connection layer. A new facial expression feature can be obtained, and the classification accuracy can be improved. In general, for the recognition of facial expression images, the proposed method based on the fusion LBP expression features and convolutional neural network expression features is used to obtain the best performance of 91.28% in the comparative experiment. An efficient extension of the expression feature texture expression channel is carried out. On the other hand, convolutional neural networks have incomparable advantages over other methods in abstract information representation of two-dimensional images.

Keywords Facial expression recognition · Machine learning · Deep convolutional neural network · Local binary mode (LBP)

1 Introduction

In recent years, people have tried to use computer vision and image processing technology to analyze the state features of facial expression images and automatically recognize human facial expressions [1–3]. Its application prospects are very broad, such as human-computer intelligent interaction, emotional computing, intelligent monitoring, animation synthesis, video games, psychology, and medical monitoring [4]. Facial expression recognition plays a role in promoting related disciplines and fields. At the same time, related technologies based on facial expression recognition technology are widely welcomed by virtue of their convenience and interesting application to people's lives. Therefore, with the computer, the increasing processing power and the study of more complex and accurate expression recognition technologies are of great significance.

At present, one of the main challenges encountered by face recognition technology is the different postures, that is, the difficulty in recognizing the face caused by the rotation at any depth. The difference in face images caused by rotation, illumination, and angle is often greater than the difference in features between people's identities. However, multi-pose face recognition has great potential for dealing with uncooperative subjects in many applications. One is to re-implement and reuse the face recognition results of the existing passive biometrics technology [5]. In recent years, scholars have invested a lot of energy in attitude-changing face recognition research and proposed many famous methods. Literature [6] uses a salacious representation-based classification (SRC) to identify occluded facial expressions. Firstly, the occlusion dictionary with redundancy is constructed through multi-level image segmentation. Then, the sparse representation coefficient of the test image is obtained through redundancy decomposition. Finally, the representation category of the test image is determined in a single subspace. This method makes the decomposition coefficient of test image sparser and avoids

✉ Fanzhi Kong
kfz3042@126.com

[1] School of Electronics and Information, Communication University of Zhejiang, Hangzhou 310018, Zhejiang, China

the interference of feature recognition to expression classification. Literature [7] proposed a new feature learning method based on depth neural network (DNN) and applied it to multi-angle facial expression recognition. This method extracted a set of scale invariant feature transformation (SIFT) features corresponding to ground punctuation from face images. In reference [8], a facial expression recognition method based on improved sparse representation recognition (MSRR) was studied. Haar-like+LPP was used to extract features and reduce dimensions. Dynamic regularization factors were added in the iterative process to suppress noise and improve accuracy. However, there are still several problems to be solved urgently in face recognition, such as the lack of understanding of the image subspace of attitude change, the intractable problem of 3d face modeling, the overly complex reflection mechanism of face surface, and the traditional machine learning method. However, it does not eliminate the disadvantages of the feature selection stage that the artificially selected features cannot well represent the facial features.

In order to extend and improve the extracted facial features as many ways as possible in the process of facial expression recognition, an improved expression recognition network (LBP–deep convolutional neural network (DCNN)) is designed, which combines the improved LBP facial features with the deep convolution neural network facial features. The LBP operator is modified to match the region of facial expression features. The best performance of convolution neural network is obtained by contrast experiment. Then, the improved LBP feature is extracted from the expression image. Then, a deep convolution neural network is trained to extract the expression depth abstract feature. The abstract feature is fused with the improved LBP feature, and a seven-class output layer is trained. Finally, the expression recognition results are obtained.

2 Convolutional neural network model and algorithm

2.1 Deep convolutional neural network

In the feature extraction of classical pattern recognition, the feature extraction method is based on human feature extraction experience and subjective consciousness. This feature extraction method has many drawbacks. The rules for extracting features are different, even with the same recognition algorithm. It will lead to different classification results, and even the difference in the order of the features will affect the classification results, and the pre-processing effect on the images will also affect the whole feature extraction process.

Although the classical pattern recognition feature extraction has a strong intuition, its main idea is to analyze the correlation of the extracted features, so as to determine the most representative features of classification, and then remove the redundant features irrelevant to classification, reduce the dimensionality of data, and achieve classification. This method is difficult to achieve effective classification with a general processing method. The whole structure of convolution neural network is shown in Fig. 1 [9, 10]. The hidden layer of convolution neural network is composed of convolution layer and maximum pool sampling layer alternately, except the full connection layer closest to the output layer. As one of the best performance pattern recognition systems, convolutional neural networks have been successfully applied in the field of handwritten character recognition.

2.2 Convolutional neural network forward propagation

In the following, some important operations of the network forward propagation process will be described mathematically.

1. Convolution layer

In each convolutional layer, there are several learnable convolution kernels. Each feature graph of the upper layer is convolved with the learnable convolution kernel. After an activation function, the output feature graph of the convolutional layer can be obtained.

$$x_j^l = f\left(u_j^l\right) \tag{1}$$

$$u_j^l = \sum_{i \in M_j} x_i^{l-1} \times x_{ij}^l + b_j^l \tag{2}$$

Among them, u_j^l is called the net activation of the jth channel of the convolutional layer l, which is obtained by the convolutional summation and offset operation of the previous layer output feature map, and x_j^l is the convolutional layer l. The output of the jth channel, $f(\cdot)$ is called the activation function, and the commonly used activation functions are sigmoid and tanh. M_j is the subset of the upper feature map used to calculate the u_j^l input, x_{ij}^l is the convolution kernel matrix, and b_j^l is the offset after the feature map convolution operation. For a convolution kernel corresponding to an output feature map x_j^{l-1}, x_{ij}^l may be different.

2. Downsampling layer

Each input feature map is downsampled by the following formula and the feature map is output:

$$x_j^l = f\left(u_j^l\right) \tag{3}$$

$$u_j^l = \beta_j^l down\left(x_j^{l-1}\right) + b_j^l \tag{4}$$

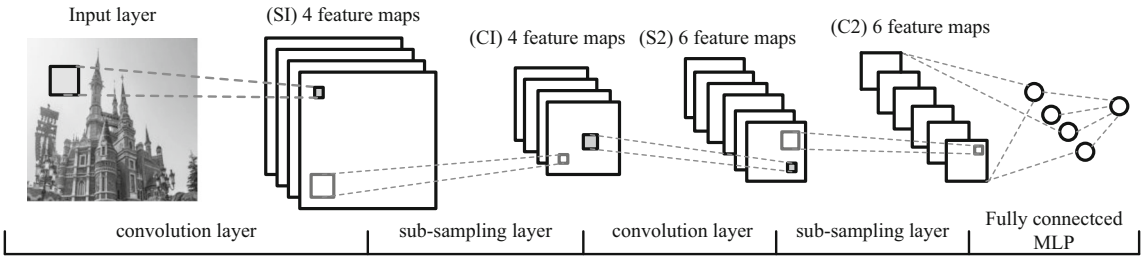

Fig. 1 Deep neural networks based on convolutional neural networks

The net activation of the jth channel, referred to as the downsampling layer l, is to output the feature map x_j^{l-1} of the previous layer, β is the weighting factor of the downsampling layer, and b_j^l is the bias term of the downsampling layer. The operation symbol $down(\cdot)$ represents a downsampling function. For the input feature map x_j^{l-1}, the $n \times n$ image block of the sliding window method is used and all pixels in each image block are summed, averaged, or maximized. The resulting value is output, and the output image is then reduced by n times in both dimensions of the horizontal and vertical axis.

3. Fully connected layer

The full connection layer should be arranged as a single-dimensional feature vector of the upper layer according to a certain rule, and then it can be used as an input, and the well-ordered feature vector can be weighted and summed by the activation function to obtain the full output of connection layer l:

$$x^l = f\left(u^l\right) \tag{5}$$

$$u^l = w^l x^l + b \tag{6}$$

Among them, u^l is called the net activation of the fully connected layer l. The weighting and offset of the previous layer output feature map x_j^{l-1} can be used to obtain net activation, and w^l is called the weight coefficient of the fully connected network, and b^l is this layer's bias item.

2.3 Backpropagation algorithm for convolutional neural networks

For the inverse algorithm of the convolutional neural network, the parameters that need to be optimized are mainly the convolution kernel parameters, the downsampling layer weight and the full connection weight, and the offset parameters in each layer [11–14]. The essence of the backpropagation algorithm is to optimize the network parameters continuously by reducing the effective error between the network layers and solving the minimum overall error of the network, so as to make the network parameters develop towards the direction of overall error minimization and make the actual output of the network approach the target output to the maximum extent.

Taking the squared error loss function as an example, the backpropagation is used to realize the multi-classification process. The square of the difference between the expected output value and the actual output value at the output is defined as the total error in the training process for a multi-class problem:

$$E(w, \beta, k, b) = \frac{1}{2}\sum_{n=1}^{N} \|t_n - y_n\|^2 \tag{7}$$

where t_n is the true value of the category label to which the nth sample belongs and y_n is the nth sample predicted by the forward propagation network to predict its category label at the output end. For multi-classification problems, one-dimensional vectors are often used to represent output category labels. The category label dimensions corresponding to the input samples are positive, and the values of the dimensions of other output category labels are 0 or negative, which is determined by the type of activation function selected. It is decided that, for other category labels, when sigmoid is selected as the activation function, the value of the other dimensions of the output label is 0, and when the activation function is tanh, it is − 1. In the process of backpropagation algorithm, the optimization of network parameters based on the gradient descent method is mainly to adjust the network parameters to the direction of training error reduction by gradient descent. Here, the backpropagation algorithm will be introduced by taking the convolutional neural network structure in which the "convolution layer-pooling layer" is connected to multiple fully connected layers. The sensitivity of the first layer of the network is:

$$\delta^L = \frac{\partial E}{\partial u^l} \tag{8}$$

Among them, δ^l characterizes the degree of change between the total error and the net activation, and the total error E varies with the net activation u^l. The idea of backpropagation is actually to establish the total error and the partial derivative of the network parameters for the sensitivity of all network layers in the network, so that the training error changes in the direction of decreasing [15].

In order to obtain the sensitivity of the layer l of the convolution layer, the sensitivity of the layer of $l+1$ to be used,

that is, the sensitivity of the downsampling layer connected to the layer l of the convolution layer, is used to express the sensitivity of the layer l of the convolution layer. Then, the total error and the rate of change of the convolutional layer parameters, that is, the partial derivatives of the convolution kernel parameter k and the offset parameter b, are obtained. Since the sensitivity of the downsampling layer is not the same as the size of the convolution layer sensitivity, this step needs to upsample the sensitivity of the downsampling layer $l+1$ to obtain the sensitivity of the convolutional layer l, and then the partial derivative of the activation function of the l layer. The sensitivity obtained by sampling with the $l+1$th layer is multiplied item by item, whereby the jth channel sensitivity in the lth layer can be obtained. Then, the sensitivity and the parameters in the convolutional layer l can be biased. The partial derivative of the total error E versus the bias term b_j^l can be obtained by summing all the nodes in the convolutional layer l sensitivity.

$$\frac{\partial E}{\partial b_j^l} = \sum_{u,v}\left(\delta_j^l\right)_{u,v} \tag{9}$$

When using chain derivation, each of the feature map elements multiplied by the convolution kernel is used to obtain the partial derivative, so that the partial error can be obtained from the convolution kernel parameter:

$$\frac{\partial E}{\partial k_{ij}^l} = \sum_{u,v}\left(\delta_j^l\right)_{u,v}\left(p_i^{l-1}\right)_{u,v} \tag{10}$$

where $\left(p_i^{l-1}\right)_{u,v}$ is the x_i^{l-1} element multiplied by k_{ij}^l element when x_i^l is calculated.

The sensitivity of the downsampling layer l is first expressed by the sensitivity of the convoluted layer $l+1$, and then, the total error E and the downsampling parameter weighting factor β and the partial derivative of the biasing parameter b are calculated. The sensitivity δ of the previous layer and the corresponding sensitivity of the next layer are used to recursively calculate the sensitivity of the lower sampling layer l. In addition, the connection weights between the input of this layer, such as the feature graph and the output feature graph, are multiplied, namely the convolution kernel parameters. As shown in Formula (11), it is the sensitivity solution process of the channel in the l layer and j layer:

$$\delta^L = f'\left(u^L\right)\circ conv2\left(\delta_j^{l+1}, rot180\left(k_j^{l+1}\right)', full'\right) \tag{11}$$

In MATLAB, the convolution kernel can be rotated by 180° to calculate the cross-correlation. The convolution boundary is processed by zero-padding. Then, the sensitivity of all elements can be summed to obtain the partial error-offset bias; the solution steps are the same as before:

$$\frac{\partial E}{\partial b_j^l} = \sum_{u,v}\left(\delta_j^l\right)_{u,v} \tag{12}$$

First, define the downsampling operator $d_j^l = down\left(x_j^{l-1}\right)$ and then, calculate the partial derivative of the total error E to the sampling weight β by the following Formula (13):

$$\frac{\partial E}{\partial \beta_j^l} = \sum_{u,v}\left(\delta_j^l \circ d_j^l\right)_{u,v} \tag{13}$$

Here is the assumption that the next layer of the downsampling layer is a convolutional layer, and a similar derivation can be made even if the next layer is a fully connected layer.

The sensitivity calculation formula for the fully connected layer l is as follows:

$$\delta_j^l = \left(w^{l+1}\right)^T \delta^{l+1} \circ f'\left(u^l\right) \tag{14}$$

Calculating the neuron sensitivity of the output layer can be obtained by the following formula:

$$\delta^L = f'\left(u^L\right)\left(y^n - t^n\right) \tag{15}$$

The total error of the network is biased to the bias term as follows:

$$\frac{\partial E}{\partial b^l} = \frac{\partial E}{\partial u^l}\frac{\partial u^l}{\partial b^l} = \delta^l \tag{16}$$

The next step is to update the weights for each neuron with sensitivity. For a given fully connected layer l, the direction of weight update is represented by the inner product of input x^{l-1} and sensitivity δ^l of the layer:

$$\frac{\partial E}{\partial w^l} = x^{l-1}\left(\delta^l\right)^T \tag{17}$$

3 Proposed face recognition method

3.1 Detection and location of the face area

According to the idea of AdaBoost algorithm—multiple weak classifiers are aggregated into strong classifiers, and some high-priority features are selected from many Haar-like features. Each Haar-like feature is regarded as a weak classifier, and several weak classifiers of Haar-like feature are cascaded to form a strong classifier. Finally, several strong classifiers are connected in series to obtain a cascaded AdaBoost algorithm classifier that can detect the face region. Essentially, the cascaded classifier can be seen as a decision tree and the strong classifiers combined by the AdaBoost algorithm are distributed at each layer. By setting a threshold for each layer,

more than 99% of the face areas can pass through the detection window, while the non-face area cannot pass. Figure 2 shows the results of our detection of the database using the AdaBoost algorithm.

Since the integral graph is used to obtain the rectangular feature value, it only uses the values of the four vertices of the rectangular frame and has the same calculation amount for the rectangular frames of different sizes, so that the face region can be detected quickly. In practice, the method we use is to continuously enlarge the size of the detection window and make the size of the image unchanged, so as to know the area to be detected. In the process of equal scale magnification of the detection window, the magnification factor we adopted is $n = 1.5$, in order to prevent the increase of the missed detection rate caused by the excessive magnification factor.

Through observation, we found that, for the face image detected by AdaBoost algorithm, not every pixel or every region of the whole image contains the emotional information of people. For example, in daily life, people's perception of other people's current emotions can only be identified by the posture of several facial organs. Other face areas (such as hair, skin color, ears, etc.) are basically unrelated to one's emotional information, and this also means redundant information for the expression learning process [15]. Therefore, in this chapter, the face image is cropped. The actual training sample data and test sample data used in the experimental part are only the two most representative facial expressions after face preprocessing and face expression segmentation. The area image includes an eyebrow area and a mouth area.

3.2 Local binary expression feature principle

LBP is called local binary mode because its processing method takes the gray value as the starting point, and arbitrarily selects one pixel on the image as the center point to set the threshold, and binarizes the surrounding pixels: if the pixel value is greater than or equal to the threshold, it will be marked as 1; otherwise, it will be marked as 0. It is an operator that performs a binary description of the gray value of an image.

We assume that T is the LBP binary coding in a certain region and T can be written as $T = (g_c, g_0, \cdots, g_{p-1})$ with pixel points as variables. $g_c(x_c, y_c)$ in the formula represents the gray value of the pixel in the center of the coding region, while $g_p(p \in 0 \sim P-1)$ is jointly determined by the P value and the specified radius R. These g_p points surround the g_c points to form a circular region with g_c as the center. If the coordinate of $g_c$is 0,0, then we can calculate the value of the coordinate of g_p: $g_p = (-Rsin(2\pi p/P), Rcos(2\pi p/P))$. Figure 3 illustrates the coordinate position diagram of g_p under different P and R values. The gray value of each g_p point can be directly obtained when the coordinate is at the center of the pixel, and the pixel value of the point can be obtained by interpolation method when the coordinate is not at the center of the pixel.

According to the basic idea of LBP gray pixel binarization interpolation, we can write the T function as:

$$T = t\left(g_c, g_0 - g_c, g_1 - g_c, \cdots, g_{p-1} - g_c\right) \tag{18}$$

If we assume that the variables $g_p - g_c$ and g_c are independent of each other, then the above formula can be rewritten as:

$$\begin{aligned} T &\approx t(g_c) \cdot t\left(g_c, g_0 - g_c, g_1 - g_c, \cdots, g_{p-1} - g_c\right) \\ &\approx t\left(g_c, g_0 - g_c, g_1 - g_c, \cdots, g_{p-1} - g_c\right) \end{aligned} \tag{19}$$

$$T \approx t\left(s(g_0 - g_c), s(g_1 - g_c), \cdots, s\left(g_{p-1} - g_c\right)\right) \tag{20}$$

among them:

$$s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} \tag{21}$$

If we multiply each factor 2^p by a factor $s(g_p - g_c)$, we can transform Formula (21) into a unique LBP code. This unique code represents the LBP feature of the local image region based on P, R description, so we record it as: $LBP_{P,\ R}$. Because of its own algorithm, the LBP operator has a linear invariant property for gray values in any local differential transform, which can eliminate the texture effect of illumination on the image. That is to say, under the influence of the uniform illumination distribution, as long as the order of the gray values of the image does not change, the output value of $LBP_{P,\ R}$ remains unchanged.

Fig. 2 Results of the detection of the database based on the AdaBoost algorithm

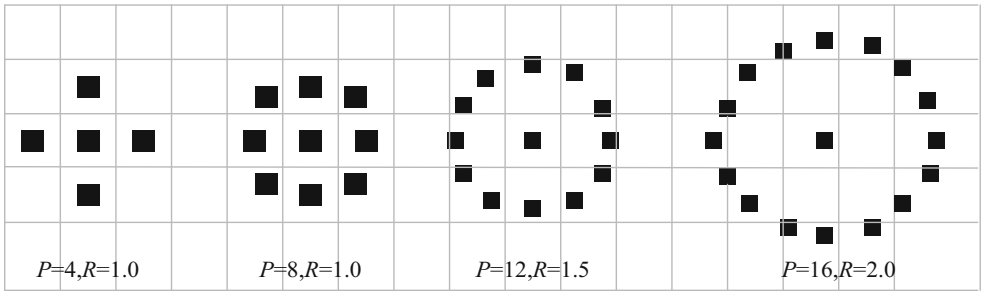

Fig. 3 Coordinate position map of LBP operator under different values of P and R

3.3 Improved local binary expression features

In the traditional LBP method for extracting image features, when performing LBP feature extraction on a two-dimensional image, the image is divided into $n \times n$ sub-regions in advance and all pixel points in each sub-region are calculated according to the LBP operator algorithm to obtain the LBP code. Although each sub-area may still include many pixels, according to the LBP operator and the expansion theory, the type of LBP encoding in each sub-area is fixed: the basic LBP mode includes 2^p, $LBP^{u2}_{P,R}$ mode. The code has $P(P-1)+2$, and the $LBP^{u2}_{P,R}$ -mode code has $P+1$. Then, we count the coding types in each sub-area and obtain the histogram of the statistics as the LBP feature of the sub-area. Then, the histograms in each sub-area are connected in the order of the sub-areas to form a total histogram vector, which is the LBP texture feature description of the entire facial expression image.

Since the object of the LBP operator operation is generally a grayscale image, when the original face image input is not a grayscale image, the original image needs to be converted into a grayscale image [16, 17]. The dimensions of the histogram obtained after feature extraction are determined by the P and n of the LBP algorithm and the mode of the LBP operator employed. In fact, the traditional LBP feature extraction methods in facial expression recognition have been used many times but in our study, facial expression of the judgment is often mainly composed of some organs on the face pose information which can get effective judgment, such as the eyebrow area and mouth area, while other face regions for very small contribution to the expression for the main characteristics. If extracting the features of the whole face image by using the traditional LBP operator, the extracted LBP feature vectors representing the texture information of the image will not grasp the focus of the expression features. On the contrary, it will increase some redundant information and enlarge the dimension of the feature vectors, even to the extent that the texture features learned is not related to expression. Therefore, the traditional LBP feature operator for the above situation is improved.

The specific method is as follows: according to the selection of expression feature area, the local images of the eyebrow area and the mouth area cut out in the face area are used as the learning area of LBP operator to extract expression features. If this operator is used in different databases, or under the background of different input images with different sizes, it needs to preprocess the image in advance to adjust the size and gray level of the image. In this experiment, because the size of the database is consistent, this processing is not done.

After selecting two expression regions of the face image, the next step is to extract the LBP features for each segment. This chapter uses the coding mode of $LBP^{u2}_{P,R}$ to extract expression features. The two expression regions are divided into 2×2 sub-blocks, and the four sub-block histograms in the $LBP^{u2}_{P,R}$ mode are extracted, and then, the histograms are connected end to end to form an $LBP^{u2}_{P,R}$ histogram of each expression region. Assuming that the radius of the LBP sample is R and the number of points sampled is P, then, the histogram dimension obtained in the $LBP^{u2}_{P,R}$ mode is $P+1$ dimension and the histogram dimension of each expression region is $4 \times (P+1)$ dimension. In order to fuse the LBP feature histograms of the two expression regions, an LBP expression feature of the entire expression region is obtained. The method we use is to normalize and fuse the features of the two expression regions [18]. Because the feature histogram dimensions of the two expression regions are small, we can use the following distribution function for normalization.

The original histogram for each region is assumed to be as follows: $Y_k = n_k$, $1 \le k \le P+1$, K is the type of LBP encoding, and Y_k is the corresponding statistical value encoded in the expression region.

Then, we calculate the cumulative sum of the histogram values: $s = \sum_{j=1}^{k} Y$.

Normalize the histogram: $H_k = \frac{n_k}{s}$, $1 \le k \le P+1$.

Then, the normalized two expression area histograms are connected end to end and merged into an LBP feature histogram of the expression area, and the histogram dimension is $2 \times 4 \times (P+1)$.

3.4 Structure and parameter design of LBP-DCNN expression recognition model

LBP features and deep convolution neural network–proposed fusion expression characteristics of facial expression

recognition method mainly have the following two advantages: (1) using deep convolution neural network expression characteristics of the study on the one hand can directly input the original image and avoids the prophase of illumination and noise influence factors such as pretreatment process; on the other hand, it can avoid the congenital errors caused by the inaccuracy of artificial selection for facial features [19, 20]. (2) Because of the characteristics of deep convolution neural network, the features it learns in depth often represent the highly abstract information of images. From another point of view, this abstract information also means ignoring the details and textures of many images. Therefore, the features obtained by deep learning of the network are not comprehensive so we introduce them outside the deep network. LBP improves the features of expression sensitive region, absorbs the advantages of LBP operator in extracting image details and texture features, and fuses the two features to obtain a fusion feature of multi-channel expression of expression information. The flow chart of the proposed LBP-DCNN expression recognition model is shown in Fig. 4:

For each database, 40,000 face image samples are got according to the method of face region detection. The extended image sample is subjected to eyebrow area, mouth area detection and cropping, and the LBP expression features of each area image are extracted, and then, the LBP features of the same image 2 area LBP features are connected end to end to obtain 40,000 LBP feature vectors. Setting the P value to 8, then, we can know that the dimension of the LBP histogram feature vector for each facial expression image is 72 dimensions according to the theory in section 3.3 of this chapter.

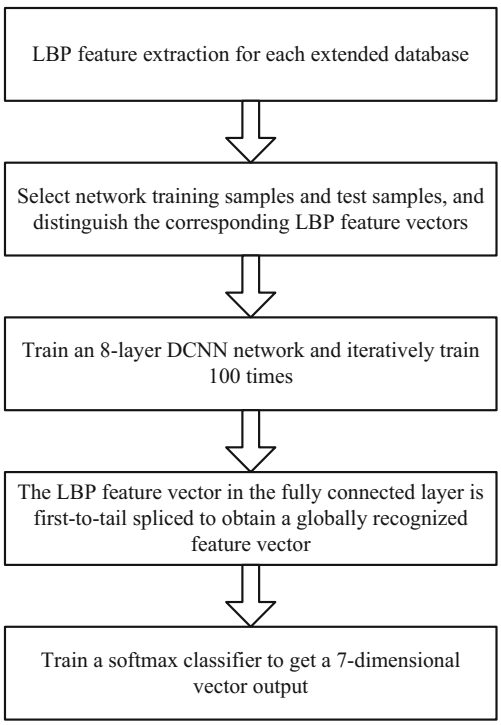

Fig. 4 Partially different levels of feature map visualization

On the other hand, construct a three-layer convolutional layer, a three-layer pooling layer, and two layers of fully connected layers. Then, randomly extract 30,000 images of each expression image library as the training samples of the network and the remaining 10,000 sheets are used as test samples of the network. The training samples and test samples do not cross each other, and each experiment library is subjected to separate experiments, and the corresponding LBP feature vectors are also distinguished; then, an 8-layer DCNN network is trained and the local image of the expression feature is taken as the input image, the network is trained, and 30,000 pictures in the sample set are randomly iterated, and a total iteration is set $n = 100$ times. In the second fully connected layer of the network, the LBP feature vector is first-to-tailed with the feature vector of the first fully connected layer to obtain a 272-dimensional global recognition feature vector.

Finally, the 407-dimensional global expression feature vector is used as the input value of the final output layer, and a SoftMax classifier is trained to obtain a 7-dimensional vector output. Each dimension of the vector represents the probability of the expression image in each class.

4 Experimental results and data analysis

4.1 Experimental computing environment

Face database can be accessed mainly by the following two methods: (1) downloading commonly used Face databases, such as ORL Face database, CMU-PIE Face database, FERET Face database, and face-scrub Face database, and (2) self-built database, which can capture, scan, and download a large number of face images on the Internet and process them into images meeting network requirements. This experiment uses the ORL face database and CMU-PIE face database. The memory of the PC is 8G, and the processor is Intel I5 processor. We did not use GPU acceleration. The experimental platform is MATLAB 2013b.

4.2 Comparison of experimental results

4.2.1 Network depth contrast experiment

In order to verify the influence of network layers on experimental recognition rate and to find an optimal network depth parameter, we simply used the depth convolutional neural network with different layers to conduct experiments on the two databases and obtained the following experimental results for comparison, as shown in Table 1.

It can be concluded from the experimental results in the above table that, for the facial expression database used in

Table 1 The influence of network layers on recognition results

DCNN network layer number	Average recognition rate of two databases (%)	Average training time
2 + 2	86.32	More than 7 h
4 + 2	88.93	More than 10 h
6 + 2	91.28	More than 12 h
8 + 2	89.61	More than 18 h

our experiment, when the number of network layers increases, the training iteration time increases continuously and the training iteration time is kept within a relatively constant time basically due to the limitation of PC hardware configuration. But we can know from the experimental results that, when we set the layers of deep convolution neural network to 6 + 2 layers, that is, three convolution layers, three pooling layers, and two fully connected layers, we get the best recognition performance. Therefore, eight layers are chosen as the layer parameters of deep convolution neural network used in our experiments.

4.2.2 Comparison of recognition results

After determining the optimal framework of deep convolution neural network, the network is constructed according to the fusion recognition method of LBP features and DCNN network expression features described in this chapter. The network is trained iteratively on ORL and CMU-PIE databases, and the results of the comparative experiments of Figs. 5, 6, 7, and 8 are obtained by combining the features of LBP with shallow learning methods.

It can be seen from the above experimental results that it has a gap of 10% in the recognition result between the shallow learning method SVM and the two depth network methods, both on the single expression recognition rate and the overall average recognition rate, and the K-NN classifier even differs with the SVM by 20%. The recognition rate can be achieved more than 80% by the SVM, which is the optimal classifier in the general machine learning methods. This fully demonstrates the power of deep learning networks in learning abstract features such as facial expressions. In addition, compared with the result of deep confidence network, our proposed method wins by an advantage, because the deep convolutional neural network has better ability to interpret and describe abstract features than the deep confidence network in processing two-dimensional images.

From the point of view of recognition rate between classes, both shallow and deep networks have low recognition rate of neutral expressions. In naked eye judgment of facial expression images, we find that neutral expressions are most easily confused with expressions such as disgust, anger and fear. In experiments, we find that neutral expressions are also easily misclassified into these three expressions by computers, while other expressions are also easily misclassified by computers. The three types of expressions, sadness, happiness and surprise, can achieve a higher classification accuracy because of

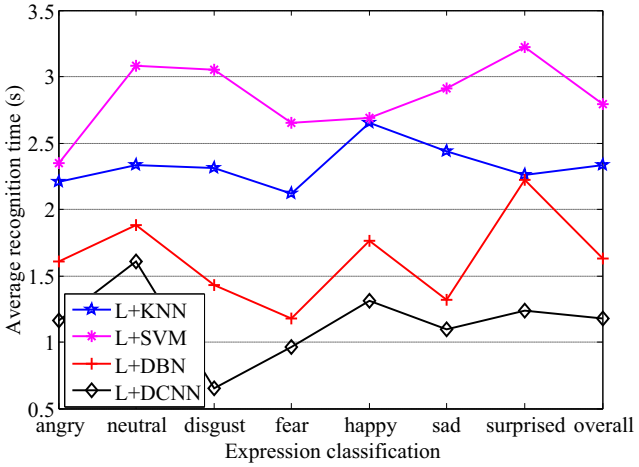

Fig. 6 The average recognition time comparison of several methods in the ORL database

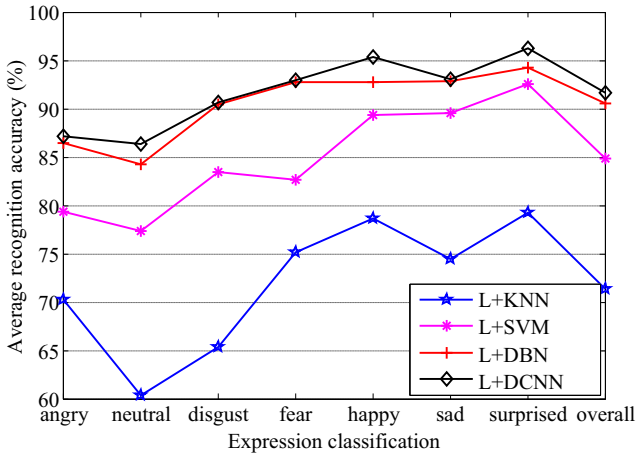

Fig. 5 The comparison of average recognition accuracy of several methods in the ORL database

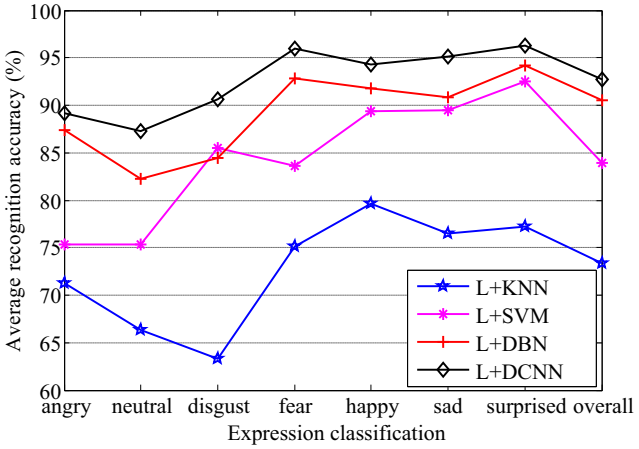

Fig. 7 The comparison of average recognition accuracy of several methods in the CMU-PIE database

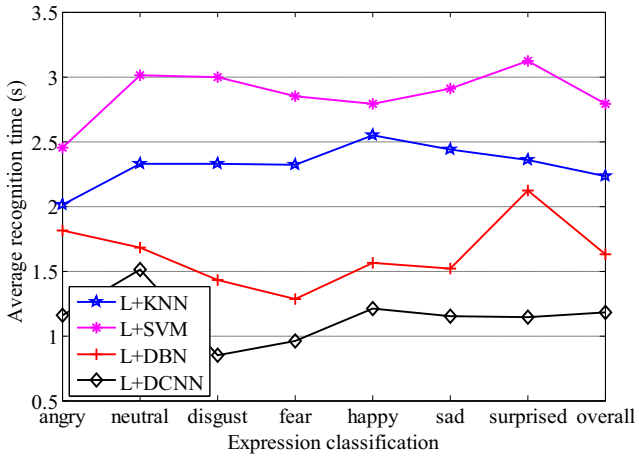

Fig. 8 The comparison of average recognition time of several methods in the CMU-PIE database

the obvious differences between the features. Generally speaking, for the recognition of facial expression images, the proposed recognition method achieves the best performance in comparative experiments. On the one hand, LBP features expand the expression channels of facial expression features effectively, on the other hand, convolutional neural networks have incomparable advantages over other methods in expressing abstract information of two-dimensional images.

In addition, in terms of identification time, the identification method proposed in this paper is due to several comparison methods. This fully demonstrates the power of deep learning networks in learning abstract features such as facial expressions.

5 Conclusion

In this paper, a method based on the convolutional neural network to extract the features of facial expression image is proposed, which integrates the features extracted by LBP operator in the facial feature area to enrich and expand the expression channels of full-connection layer features in the deep network, and obtains superior performance in the application of facial expression recognition. This paper introduces the algorithm of the basic LBP operator and the improvement of the expression-sensitive area. The structure of the expression recognition network is analyzed, and the optimal solution of the number of layers of the convolution network is selected through comparison experiments. In general, the method based on deep learning network has strong innate advantages, while the replacement experiment of LBP features or the transformation of DCNN network structure is a breakthrough that can be further explored in the field of face recognition.

Funding information This work was supported by the Public Welfare Technology Project of the Science Technology Department of Zhejiang Province (No. LGG18F010001).

References

1. Zheng W (2014) Multi-view facial expression recognition based on group sparse reduced-rank regression. IEEE Trans Affect Comput 5(1):71–85
2. Zhao X, Shi X, Zhang S (2015) Facial expression recognition via deep learning. IETE Tech Rev 32(5):347–355
3. Boughrara H, Chtourou M, Amar CB et al (2016) Facial expression recognition based on a mlp neural network using constructive training algorithm[J]. Multimed Tools Appl 75(2):709–731
4. Guo Y, Zhao G, Pietikainen M (2016) Dynamic facial expression recognition with atlas construction and sparse representation. IEEE Trans Image Process 25(5):1977–1992
5. Li W, Ke W, Li R (2015) Unsupervised feature selection based on spectral regression from manifold learning for facial expression recognition. Computer Vision Iet 9(5):655–662
6. Zhu MH, Li ST, Ye H (2014) An occluded facial expression recognition method based on sparse representation. Pattern Recogn Artif Intell 27(8):708–712
7. Zhang T, Zheng W, Cui Z et al (2016) A deep neural network driven feature learning method for multi-view facial expression recognition. IEEE Trans Multimedia 18(12):1–11
8. Wei W, Lihong X (2016) A modified sparse representation method for facial expression recognition. Comput Intell Neurosci 2016:1–12
9. Aifanti N, Delopoulos A (2014) Linear subspaces for facial expression recognition. Signal Process Image Commun 29(1):177–188
10. Gui L, Yang X (2017) Automatic renal lesion segmentation in ultrasound images based on saliency features, improved LBP, and an edge indicator under level set framework. Med Phys 45(1)
11. Majumder A, Behera L, Subramanian VK (2017) Automatic facial expression recognition system using deep network-based data fusion. IEEE Trans Cybernetics 48(1):103–114
12. Fei L, Bartlett MS (2016) Video-based facial expression recognition using learned spatiotemporal pyramid sparse coding features. Neurocomputing 173(109):2049–2054
13. Wei Y, Lin G, Sha Y et al (2014) An improved LBP algorithm for texture and face classification. Signal Image Video Process 8(1):155–161
14. Eleftheriadis S, Rudovic O, Pantic M (2014) Discriminative shared Gaussian processes for multiview and view-invariant facial expression recognition. IEEE Trans Image Process 24(1):189–204
15. Zhang Y, Xin DR, Wu YH (2016) Pedestrian detection for traffic safety based on accumulate binary Haar features and improved deep belief network algorithm. Transp Plan Technol 39(8):1–10
16. Zhang Y, Xin DR, Wu YH (2016) Pedestrian detection for traffic safety based on accumulate binary Haar features and improved deep belief network algorithm[J]. Transp Plan Technol 39(8):1–10
17. Wen G, Zhi H, Li H et al (2017) Ensemble of deep neural networks with probability-based fusion for facial expression recognition. Cogn Comput 2017(4):1–14
18. Moeini A, Faez K, Moeini H, Safai AM (2017) Facial expression recognition using dual dictionary learning ☆. J Vis Commun Image Represent 45(C):20–33
19. Dong EZ, Fu YH, Tong JG (2015) Face recognition by PCA and improved LBP fusion algorithm. Appl Mech Mater 734(12):562–567
20. Chao WL, Ding JJ, Liu JZ (2015) Facial expression recognition based on improved local binary pattern and class-regularized locality preserving projection. Signal Process 117(C):1–10