International Journal of Innovative Computing, Information and Control
Volume **12**, Number **4**, August **2016**

ICIC International ©2016 ISSN 1349-4198
pp. 1353–1364

A NEW FACE FEATURE EXTRACTION METHOD BASED ON FUSING LBP AND DBNS FEATURES

Yan Wang and Yunyun Wang

School of Computer and Communication
Lanzhou University of Technology
No. 287, Langongping Road, Lanzhou 730050, P. R. China
wangyan@lut.cn; wyy824170148@126.com

Received December 2015; revised May 2016

Abstract. *Deep Belief Networks (DBNs) method ignores the image local structure and is difficult to learn the local characteristics of face image, the network training time is also too long to make full use of the facial texture feature and reduce the bad influence of illumination, and a novel method by fusing Local Binary Pattern (LBP) and DBNs features is proposed in this paper. In this method, the LBP feature is as the input of the DBNs. Furthermore, in order to accelerate the training speed of DBNs, the Extreme Learning Machine (ELM) is introduced into the network training process. Finally, the training network is used to classify and recognize. Experiments on ORL and FERET face databases with different resolution demonstrate that the proposed method is better than other relevant methods.*

Keywords: Feature extraction, Deep Belief Networks, LBP, Extreme Learning Machine

1. **Introduction.** As one of the important biological feature recognition technologies, face recognition has a wide range of applications in information security, interpol detection, video surveillance and human-computer interaction, etc. It is one of the hottest research topics in pattern recognition field [1]. Feature extraction is one of the key issues in face recognition [2]. For the past few years, the existing various face feature extraction methods that are based on pixels and feature expression have achieved good recognition effects in face recognition. The main methods include Principal Component Analysis (PCA) [3], Linear Discriminant Analysis (LDA) [4], Gabor wavelets [5], Local Binary Pattern (LBP) [6] and so on. The PCA method extracts global features based on the gray images and the extracted features are more sensitive to the external environment; the LDA method can realize quick recognition, but it has great limitations because it depends on the gray correlation of the training and testing images; Gabor wavelet can extract multi-scale features effectively, but its calculation is very large, and the redundancy and feature dimension are also high; as an effective texture descriptor, LBP method gets more and more attention for its characteristics such as simple calculation, strong texture expression, robustness for light. With the LBP descriptor having been introduced into the field of face recognition by researchers, the facial feature extraction algorithms based on LBP and its improved algorithms emerge endlessly, and people pay more attention to the feature extraction methods fusing LBP and other algorithms [7]. The algorithm fusing the LBP and Gabor method suggested by [8] has achieved a good result, but its computational complexity and feature dimension are higher as well.

The above traditional feature extraction methods all have good effects in face recognition, but those methods contain a lot of subjective factors in the process of feature extraction, and the representations of features excessively rely on artificial selection while the deep learning network has a strong ability of function representation. It has the excellent properties when learning the essential feature of the data from small sample data [9]. DBNs (Deep Belief Networks) is one of the typical models of deep learning model, which was proposed by Hinton et al. in 2006 [10-12]. It can solve the nonlinear and complex structure problems effectively that the linear feature extraction methods such as PCA and LDA cannot deal with very well, that is, the DBNs can complete the nonlinear dimension reduction of high-dimensional face feature and the extraction of essential feature well. However, with the pixel level face image feature as the input of DBNs directly, the network will learn bad characteristics due to the influence of illumination and other factors. Therefore, in this paper, we consider to extract the LBP feature of the face images first, and then take the LBP feature as the input of DBNs for final recognition, and the obtained features would have a better classification property by combining the LBP features and DBNs; meanwhile, use the fast learning ability of the Extreme Learning Machine (ELM) [13] to improve the training speed of DBNs in the network fine-tuning phase, and finally realize the face recognition on the top of the DBNs. And on this account, we proposed a facial feature extraction method based on LBP and DBNs, and at last show the validity and the correctness of the proposed method through experiments.

2. Theoretical Principle.

2.1. Local Binary Pattern (LBP).

Local Binary Pattern (LBP) method is an effective texture descriptor method which was first proposed by Ojala et al. [6]. It describes the image texture feature very well, and the dimension is not very high, so it has been widely used in recent years. Each pixel in the image is marked to get a binary number in traditional LBP operator, and then the binary number is converted into decimal number to get the LBP code of the pixel [14]. Eventually the histogram composed of the LBP codes of all pixels is regarded as LBP feature of the image. Figure 1 is the schematic diagram of original LBP operator.

In order to improve the defects that original LBP cannot extract the texture features of different scales, the original LBP operator was improved by Ojala et al. That is, the original 3×3 neighbor was extended to arbitrary neighbor, and the square was replaced by the circular (as is shown in Figure 2). The improved LBP operator is usually represented

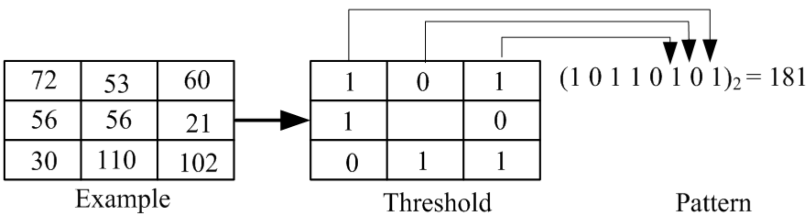

FIGURE 1. The original LBP operator schematic diagram

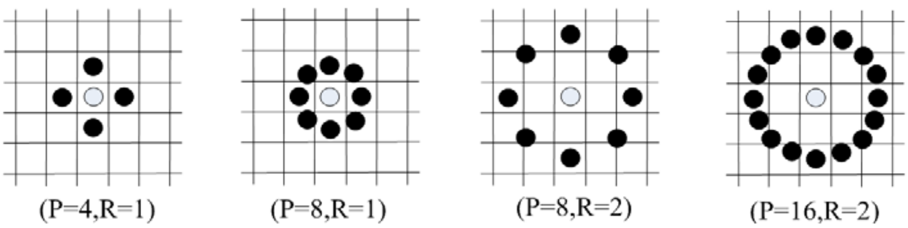

FIGURE 2. 4 kinds of LBP operator

by the parameter (P, R). Different is the (P, R) value, different is the LBP operator. For arbitrary LBP operator, Equation (1) is the code formula:

$$LBP_{P,R} = \sum_{i=0}^{P-1} s(g_i - g_c)2^i, \quad s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} \tag{1}$$

where P and R respectively represent the number of points and the radius in the template. g_i $(i = 0, \ldots, P-1)$ is the neighbor of g_c. 4 kinds of LBP operator are illustrated in Figure 2.

In this paper, we adopt the uniform LBP operator to calculate LBP code which can reduce the feature dimension tremendously.

2.2. **Deep Belief Networks (DBNs).** The concept of Deep Belief Networks (DBNs) was first mentioned by Hinton et al. in 2006 [10]. It has the deep structure of multiple nonlinear mapping, can complete the complex functional approximation, solve the slow convergence speed and is easy to fall into local optimal problems of traditional back propagation algorithm in training multilayer neural network [15]. [16] has proved that it can improve the regressive performance of Gaussian process using DBNs to learn the feature space. A typical DBNs is a highly complex directed acyclic graph, and it is an accumulation of a series of Restricted Boltzmann Machines (RBM) [17]. In order to get the DBNs, we first need to get a group RBM by using the greedy learning algorithm layer-by-layers [18], and then make use of the training data to adjust the entire model supervised (fine-tuning) to get the final model of the DBNs. The training of DBNs is accomplished by training these RBM layer-by-layers from bottom to top. RBM can use Contrastive Divergence (CD) algorithm for fast training [19]. In consequence, it is a good way to reduce the high complexity on the whole training of DBNs by training RBM, and simplify the RBM training to a one by one process.

2.2.1. *Restricted Boltzmann Machines (RBM).* RBM is also a typical neural network [20]. RBM is composed of visible layer and hidden layer which are connected with each other, but there is no mutual connection of visible layer-visible layer, hidden layer-hidden layer in RBM, and a typical RBM model is as shown in Figure 3.

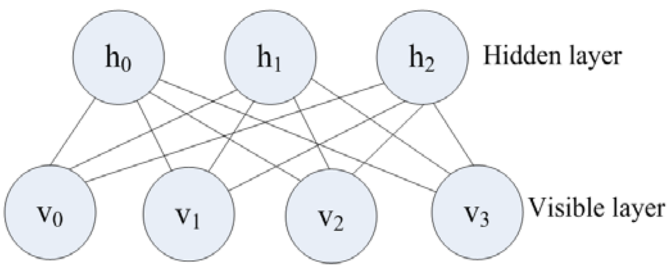

Figure 3. A typical RBM model

In RBM, the hidden layer can gain the high order correlation of the input visible layer. Use greedy unsupervised training method to train RBM layer-by-layer to obtain the generative weight, that is, during the training, maps the eigenvalues of the visible layer to the hidden layer, and rebuild the visible layer through the hidden layer. This new visible layer maps eigenvalues to the hidden layer again to obtain the newer hidden layer, and the process of performing this step repeatedly is the gibbs sampling. The joint distribution of RBM under the condition of the given model parameters is defined as:

$$p(v, h; \theta) = \exp(-E(v, h; \theta))/Z \tag{2}$$

where $Z = \sum_v \sum_h \exp(-E(v, h; \theta))$ is normalization factor or subdivision function.

The marginal probability which is given to the visual vector v by the model is expressed as:

$$p\left(v;h\right) = \sum\nolimits_h \exp\left(-E\left(v,h;\theta\right)\right) \Big/ Z \tag{3}$$

The energy function for RBM is defined as:

$$E\left(v,h;\theta\right) = -\sum_{i=1}^{I}\sum_{j=1}^{J} w_{ij} v_i h_j - \sum_{i=1}^{I} b_i v_i - \sum_{j=1}^{J} a_j h_j \tag{4}$$

where v_i is a real-value that satisfies the Gaussian distribution with the mean value being $\sum\limits_{j=1}^{J} w_{ij} h_j + b_i$ and the variance being 1, and w_{ij} is the connection weight between the visible layer and the hidden layer; b_j and a_i are the offset; I and J are the numbers of the visible layer and the hidden layer. The calculation of the conditional probability is as Equation (5):

$$p\left(h_j = 1\left|v;\theta\right.\right) = \delta\left(\sum_{i=1}^{I} w_{ij} v_i + a_j\right) \qquad p\left(v_i = 1\left|h;\theta\right.\right) = \delta\left(\sum_{i=1}^{I} w_{ij} h_j + b_i\right) \tag{5}$$

where $\delta(x) = 1/(1 + \exp(x))$. The detailed training process of RBM is as shown in [17].

2.2.2. *Deep Belief Networks (DBNs).* DBNs is a probability generation model, Figure 4 illustrates that a typical DBNs can be regarded as a superposition of many RBM, and a Typical DBNs with l hidden layers can use the joint probability distribution to express the relationship between the input vector v and the hidden vector h (as shown in Equation (6)):

$$P\left(v, h^2, \cdots, h^l\right) = \left(\prod\nolimits_{k=2}^{l-2} P\left(h^k \middle| h^{k+2}\right)\right) P\left(h^{l-2}, h^l\right) \tag{6}$$

where, $v = h^0$, $P\left(h^k | h^{k+2}\right)$ is a conditional probability distribution. The learning process of DBNs is the process of learning the joint probability distribution.

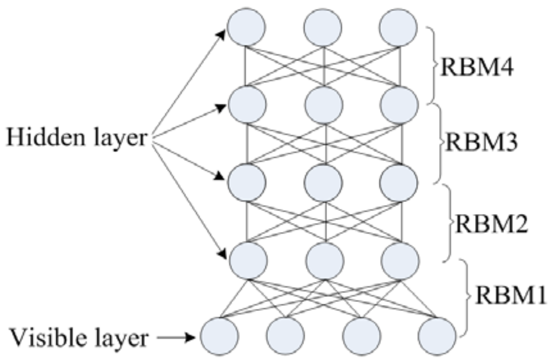

FIGURE 4. A typical DBNs model

3. **The Feature Extraction Method of Fusing LBP and DBNs.** In this paper, the feature extracted by LBP method is as the input of DBNs, and then use the DBNs to learn more abstractive and effective features automatically. In order to shorten the network training time, use ELM algorithm to fine-tune the network, and classify the face images on the top of the DBNs. LBP method is used to extract the local feature of face image, which is conducive to the classification and recognition of DBNs. The feature extraction flow chart adopted in this paper is shown in Figure 5.

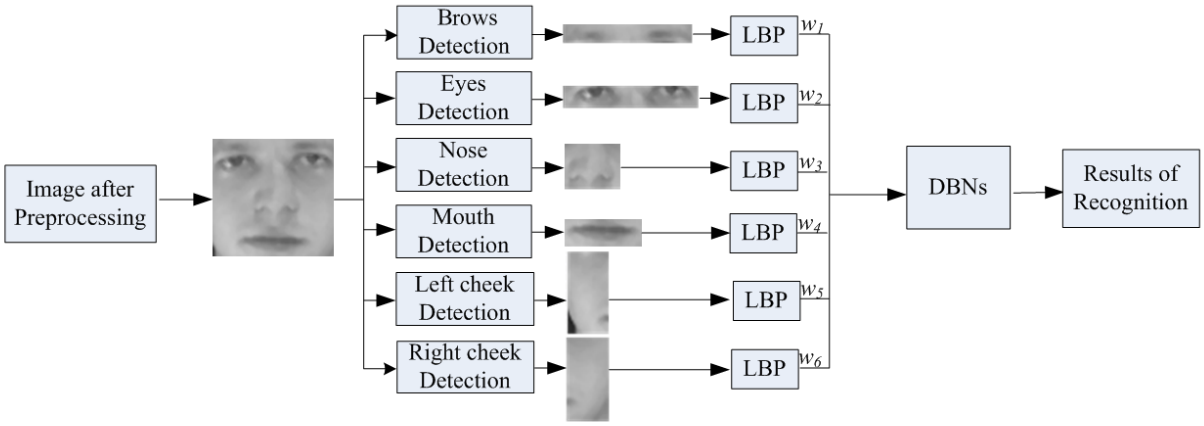

Figure 5. The feature extraction flow chart based on fusion feature extraction of LBP and DBNs

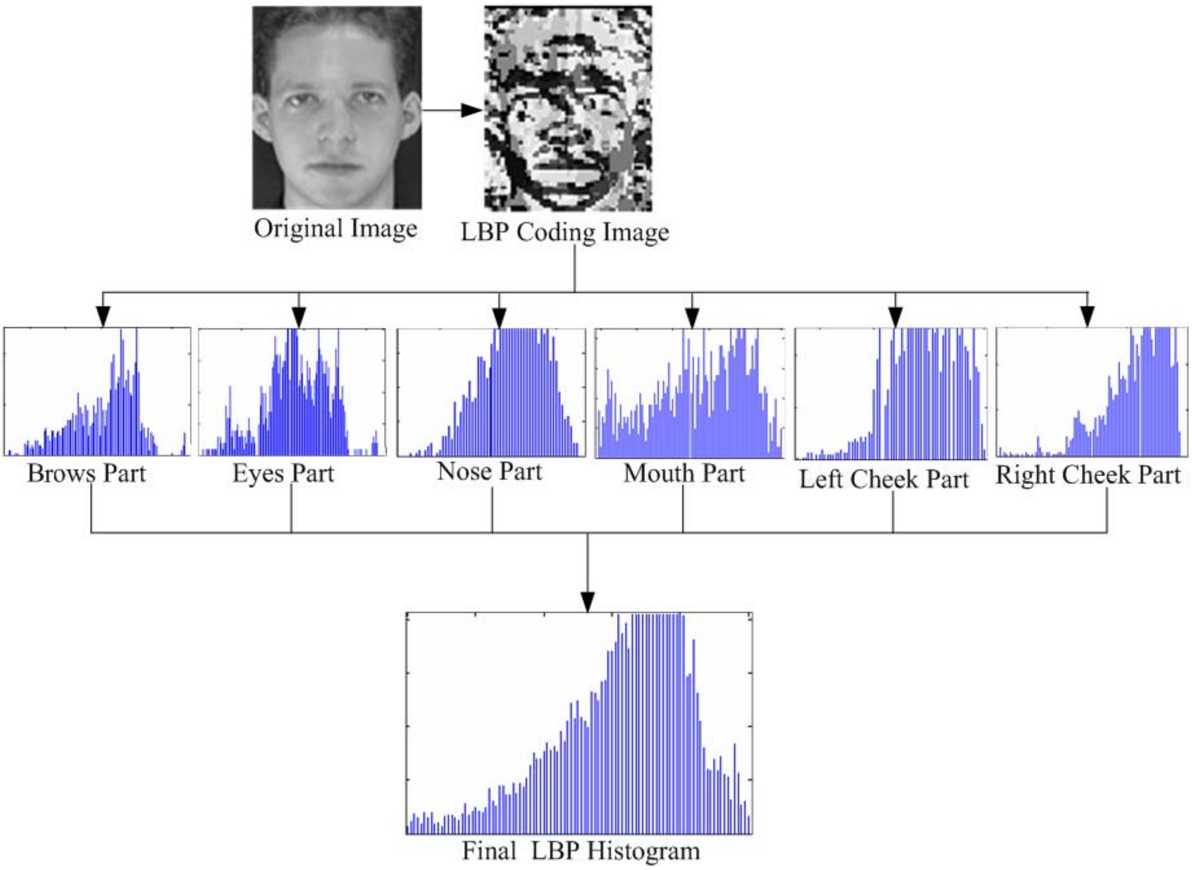

Figure 6. The LBP feature extraction process

In order to get the LBP features of the face image, this article divides the input image into six regions in advance: brows, eyes, nose, mouth, left cheek and right cheek, and then extracts the uniform LBP features of each region separately. Firstly, detect each region using the quadratic gray-level integral projection method. Secondly, scan each area using LBP operator to get the corresponding LBP coding image. Finally, extract the LBP histogram, and finish the local feature extraction based on LBP method. The LBP feature extraction process is shown in Figure 6.

After the LBP feature extraction, use the LBP texture feature as the input of DBNs, so that DBNs can effectively capture the local information of face image, and the joint distribution for the deep network at this moment is as follows:

$$p\left(H,h^{(1)},h^{(2)},\cdots,h^{(l)}\right)=P\left(H|h^{(1)}\right)P\left(h^{(1)}|h^{(2)}\right)\cdots P\left(h^{(l)}|h^{(l-1)}\right) \tag{7}$$

where H is LBP feature, $h^{(1)}$, $h^{(2)}$,...,$h^{(l)}$ are the advanced features of different layers that the DBNs learnt from the input LBP features H, and its advantages and disadvantages concern the effectiveness of DBNs learning. In this paper, H is uniform LBP texture feature and its feature dimension is only $P(P-1)+3$, which can effectively reduce the feature dimension, and it also can reduce the adverse effects of noise.

The accuracy that DBNs learn from the input uniform LBP features mainly depends on how well the network training is. The training process of DBNs follows a two-phase training strategy of unsupervised greedy pre-training followed by supervised fine-tuning. In the pre-training stage, each RBM is trained greedily, overlapping layer by layer, and use the output data of the lower RBM layer as the input of the higher RBM layer and finally structure a DBNs. In the fine-tuning stage, fine-tune the network using the traditional global optimization method. It would take a lot of time for the traditional global optimization method to adjust the weights and bias of the entire network. While the ELM algorithm does not need to adjust the weights and bias of every RBM, we consider using ELM algorithm to fine-tune the entire network in order to reduce the network training time and increase the network training speed. Assuming that the number of the nth hidden nodes is $\tilde{N}$ and the $(n-1)$th hidden node number is m. Then the neural network of the depth structure can be expressed as follows:

$$\sum_{i=1}^{\tilde{N}}\beta_i g(W_i\cdot H_{n-1,j}+b_i)=o_j, \quad j=1,\cdots,m \tag{8}$$

where W_i and b_i are respectively the input weight and bias from the $(n-1)$th hidden layer to the nth hidden layer, H_{n-1} is the advanced features of the $(n-1)$th layer that is learned by DBNs to the input features, and β_i is the output weight from the nth hidden layer to the output layer. To make the output error of DBNs be minimization, let $\sum\limits_{j=1}^{\tilde{N}}\|o_j-t_j\|=0$. That is, existing β_i makes $\sum\limits_{i=1}^{\tilde{N}}\beta_i H_{n,j}=t_j$, $j=1,\cdots,m$. The matrix representation is: $H_n\beta=T$, H_n is the output of DBNs from the $(n-1)$th hidden layer to the nth hidden layer, β is the output weight, and T is the expected output, among which

$$\begin{aligned}&H_n\left(W_1,\cdots,W_{\tilde{N}},b_1,\cdots,b_{\tilde{N}},H_{n-1,1},\cdots,H_{n-1,m}\right)\\ &=\begin{bmatrix} g\left(W_1\cdot H_{n-1,1}+b_1\right) & \cdots & g\left(W_{\tilde{N}}\cdot H_{n-1,1}+b_{\tilde{N}}\right)\\ \vdots & \cdots & \vdots \\ g\left(W_1\cdot H_{n-1,m}+b_1\right) & \cdots & g\left(W_{\tilde{N}}\cdot H_{n-1,m}+b_{\tilde{N}}\right)\end{bmatrix}_{m\times\tilde{N}}\end{aligned} \tag{9}$$

$$\beta=\begin{bmatrix}\beta_1^T\\ \vdots\\ \beta_{\tilde{N}}^T\end{bmatrix}_{\tilde{N}\times m}, \quad T=\begin{bmatrix}T_1^T\\ \vdots\\ T_N^T\end{bmatrix}_{N\times m} \tag{10}$$

Training $\hat{W}_i$, $\hat{b}_i$ and $\hat{\beta}$, to make $\left\|H_n\left(\hat{W}_i,\hat{b}_i\right)\hat{\beta}-T\right\|=\min\limits_{W,b,\beta}\|H_n\left(W_i,b_i\right)\beta-T\|$, $(i=1,\cdots,\tilde{N})$. It is equivalent to minimize loss function

$$E=\sum_{j=1}^{m}\left(\sum_{i=1}^{\tilde{N}}\beta_i g\left(W_i\cdot H_{n-1,j}+b_i\right)-t_j\right)^2.$$

Randomly initialize the weight W_i and bias b_i from the $(n-1)$th hidden layer to the nth hidden layer, and obtain the unique output matrix H_n of the hidden layer. Fine-tune DBNs can be transformed into solving a linear system $H_n\beta = T$. And the output weight β can be identified as $\hat{\beta} = H_n^+T$, H_n^+ is the generalized inverse of the matrix Hn. Also the norm of the obtained $\hat{\beta}$ is minimum and unique.

The training process of DBNs is as follows.

For the pre-training stage, use the LBP texture feature as the input of the first RBM layer, do Gibbs sampling to each layer RBM, and get each RBM using CD algorithm. To a DBNs with n hidden layer nodes, we can obtain $n-1$ RBM by the greedy training method, including the input layer, the first hidden layer, $\ldots$, the $(n-1)$th hidden layer. While the weights and bias from the $(n-1)$th hidden layer to the nth hidden layer and the nth hidden layer to the output layer are determined by ELM algorithm, that is, we use ELM algorithm to fine-tune the established model which does not need to adjust parameters. Figure 7 illustrates the DBNs structure used in this paper.

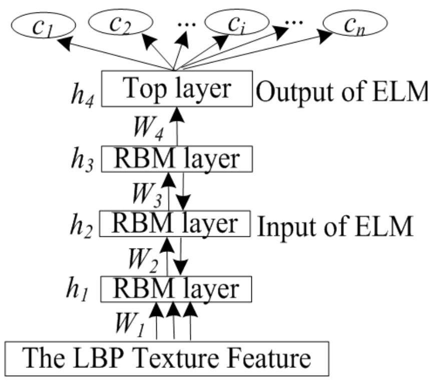

FIGURE 7. The DBNs structure of this paper

The experimental operating steps of our method are as follows:

Step1: Image preprocessing: Normalization, that is, cut the face images into the same size in order to calculate conveniently and avoid the high dimension problems;

Step2: Divide the face images into six regions after preprocessed: brows, eyes, nose, mouth, left cheek and right cheek, which are expressed as R_1, R_2, R_3, R_4, R_5, R_6, and set a weight factor w_i $(i = 1, 2, \ldots, 6)$ for each region according to their different importance. The larger the w_i is, the more important the region i is. The weight factors of those six areas in this paper are respectively set to be: $w_1 = 2$, $w_2 = 4$, $w_3 = 2$, $w_4 = 4$, $w_5 = 1$, $w_6 = 1$;

Step3: Extract LBP features of each region, and normalize the feature vector to the range of $[-1, 1]$;

Step4: Multiply the normalized feature by its corresponding weight to be the region feature vector, and connect all the region feature vectors to get the LBP feature of the entire face image;

Step5: Use the uniform LBP texture feature of the training sample as the visible layer input of DBNs, train the network layer by layer and use the ELM algorithm to fine-tune the whole network;

Step6: Use the uniform LBP texture feature of the test sample as the visible layer input of DBNs after the deep network training ended, then learn and extract abstract features of the test sample from the bottom up using the optimized network. Finally, classify the face image on the top of DBNs and output the correct recognition rate.

4. **The Experimental Results and Analysis.** The ORL and FERET face databases are used for the experiments to test the recognition performance of the proposed algorithm. First, do image preprocessing before the experiments, including geometry and illumination normalization. The geometry normalization mainly contains rotation, shear and scaling, normalizing the image respectively to the size of 64×64; the illumination normalization contains gamma compression difference of Gaussian filter and so on. The computer configuration of the experiments in this paper is: Intel(R) Core(TM) i3 CPU M370 @ 2.40 GHz 2.39 GHz, memory of 2GB; simulation software is MATLAB R2008a.

The ORL face database includes 40 people's images, and each has 10 images with the size of 92×112, and these images are shot in different time, different positions and different illumination. The posture change of the face is relatively small. In the experiments, select 5 images of each person randomly as the training sample, and the rest as the test sample. The structure of deep belief network is set to be: m-1000-250-40, among the network, m is the dimension of the input feature vector, 40 is the node number of the output unit, namely the number of categories.

The FERET face database contains two subsets named *fa* and *fb*, which include 14051 images of 1196 people; these images are all in different postures, different facial expression and different illumination. Compared with the ORL database, the FERET database has more face images, and the images also contain obvious changes of posture and expression, and of course the illumination change, so it can be used to test recognition performance of our method in larger face database. This experiment still adopts the same experimental strategy as on the ORL database for the experiment configuration. Select 500 people's images in the experiment, subset *fa* as the training set and *fb* as the test set. Because of the change of the category, under the condition of ensuring the effective feature learning and reasonable learning rate, the structure of deep belief network is set to be: m-2000-1000-500, among the network, m is the dimension of the input feature vector, and 500 is the node number of output unit.

In the stage of training, cycle training 200 rounds on each RBM layer of the input module, and then do 200 rounds circuit training on each RBM layer in DBNs using the obtained generative features. In the experiments, compare the time-consumption and recognition rate of the DBNs method in [20], LBP+DBNs method in [11] and LBP+DBNs+ELM method proposed in this paper under different number of the hidden unit to verify the feasibility and effectiveness of the proposed algorithm. The comparison results on ORL and FERET databases are respectively shown in Table 1 and Table 2.

Table 1. The recognition rate and algorithm time-consumption comparison under different number of hidden unit on ORL database

Method	Comparative item		The number of hidden units				
			1000	2000	3000	4000	5000
DBNs [20]	time-consumption	Training time/s	137.78	342.82	694.73	1143.37	1630.15
		Testing time/s	5.63	13.27	30.34	45.39	57.58
	Recognition rate (%)		44.23	67.78	80.48	90.14	94.87
LBP+ DBNs [11]	time-consumption	Training time/s	149.17	382.52	720.31	1164.53	1706.08
		Testing time/s	6.12	15.82	28.45	47.39	58.54
	Recognition rate (%)		52.12	73.03	84.93	91.72	96.67
LBP+ DBNs +ELM	time-consumption	Training time/s	98.97	178.22	512.86	1056.86	1475.76
		Testing time/s	3.32	11.08	26.89	40.13	53.76
	Recognition rate (%)		54.32	75.23	82.87	93.38	98.07

Table 2. The recognition rate and algorithm time-consumption comparison under different number of hidden unit on FERET database

Method	Comparative item		The number of hidden units				
			1000	2000	3000	4000	5000
DBNs [20]	time-consumption	Training time/s	156.58	410.27	775.16	1183.29	1830.05
		Testing time/s	6.37	18.25	29.33	48.21	68.14
	Recognition rate (%)		34.78	57.35	75.22	82.65	91.24
LBP+ DBNs [11]	time-consumption	Training time/s	163.17	448.52	801.31	1203.52	1873.13
		Testing time/s	6.72	19.82	30.45	49.36	68.73
	Recognition rate (%)		37.53	60.17	77.29	88.86	93.37
LBP+ DBNs +ELM	time-consumption	Training time/s	138.76	389.99	649.76	1053.37	1675.39
		Testing time/s	5.90	13.75	24.58	41.49	63.74
	Recognition rate (%)		40.38	62.46	80.58	90.71	95.53

After analyzing the experimental data of the above two tables, we can know as follows.

(1) The more the number of hidden units is, the better the face image features expressed by the deep network are. However, with the increasing of the number of hidden units, the training time and testing time of the network would also be increased, and the calculation is increased gradually. Compared with the methods in [11,20], the network training time and testing time of LBP+DBNs+ELM method are the least, namely using the ELM method to optimize the network can reduce the training time of the network. From the two tables, we can also learn that, under the same number of hidden units, the recognition effect obtained by the LBP+DBNs method is the best when using ELM algorithm to fine-tune the entire network, followed by the LBP+DBNs method using traditional global optimization algorithm to optimize the network, and the DBNs method that uses the pixel level as the input of DBNs has the lowest recognition rate. Compared with the traditional LBP, the uniform LBP can not only reduce the feature dimension in quantity, but also reduce the negative influence of the noise, and it also has better robustness to the illumination and small rotation. It can be seen from the above two tables that the uniform LBP makes the performance improved instead of making the deep network learning ability decreased. And using ELM algorithm to optimize network not only reduces the network training time and testing time, but improves the accuracy of recognition for the fast learning ability of ELM algorithm. Therefore, taking the uniform LBP texture feature instead of the original pixel level as the input of the DBNs can not only reduce the feature dimension, but denote the face image information better and is more advantageous to the deep network learning. And optimizing the network using ELM method can promote the learning performance of DBNs and save the training time of the network.

(2) After comparing the above two tables, we can know that for either DBNs method or LBP+DBNs method, the recognition rates on the FERET database are all lower than those on the ORL database, and the time-consumptions on FERET database are also relatively more than those on ORL database. This is because that the face images in the FERET database are more than that in the ORL database, and the changes of the posture, facial expression and illumination in FERET database are more obvious than that in ORL database, and all of these have a great influence on the network learning. The training time and testing time of LBP+DBNs method are more than those of DBNs method in the above two tables when using the traditional global optimization algorithm to optimize the network, and the cause of the problem will be the next content of our study.

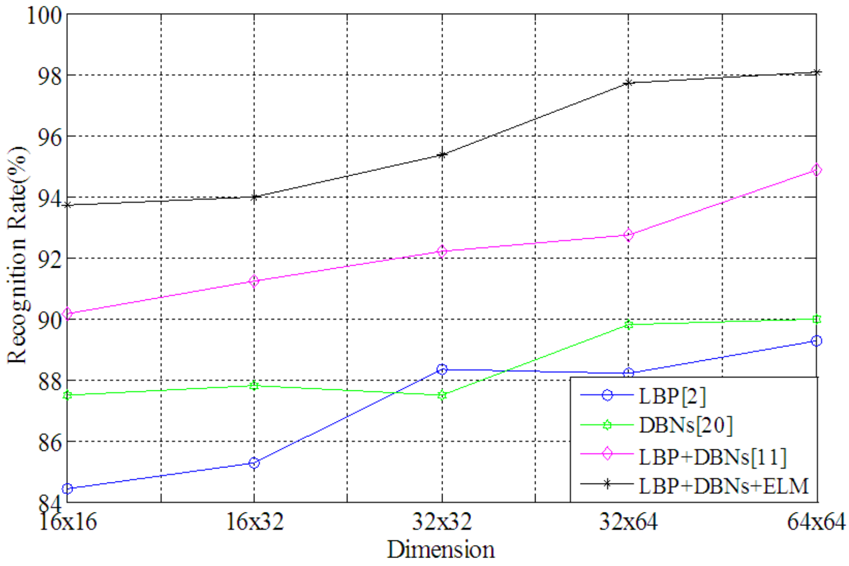

Figure 8. Recognition on ORL database

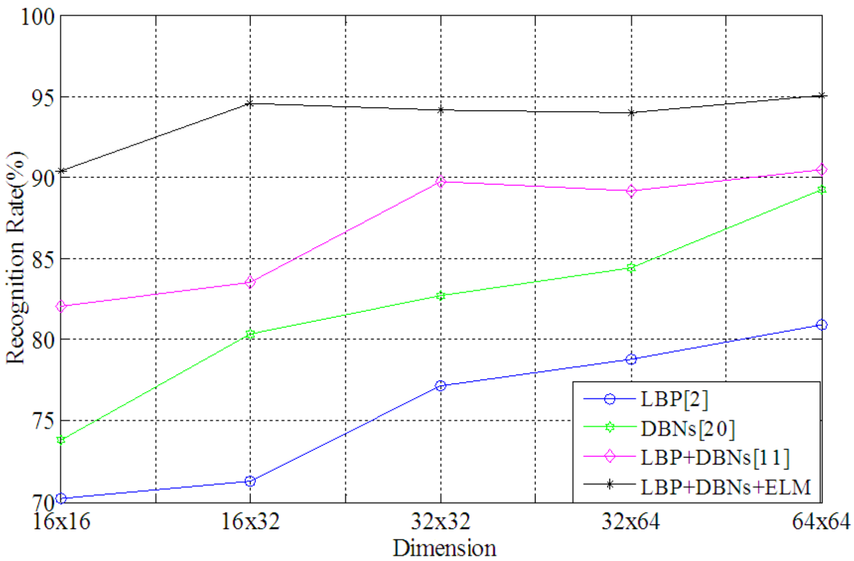

Figure 9. Recognition on FERET database

To further prove the effectiveness of our method, this paper respectively does the experiments of the methods in [2,11,20] and our method on the ORL and FERET face databases under the different dimensions. The recognition results are respectively shown in Figure 8 and Figure 9.

From the figures, we can also know, the recognition results of our method are all better than other methods under the condition of different dimensions.

In order to verify the robustness of our method to the illumination change experiment on the Yale-B face database which has a larger illumination change, select 12 images of each person randomly as the training sample, and the rest as the test sample, and the results are shown in Table 3. The results show that our method is superior to other methods, namely the method has strong robustness to illumination change.

According to the above analysis, we know that, in the process of face recognition, the LBP+DBNs+ELM method not only overcomes the weakness that DBNs cannot well

Table 3. The recognition rates of different algorithms on Yale-B database

Methods	PCA ([3])	LDA ([4])	Gabor+DBNs ([5])	LBP+DBNs+ELM
Recognition rate (%)	71.56	83.81	85.36	89.53

learn local structural features of face image, but also makes DBNs have better robustness to light and small translation by using the LBP texture features as the input of DBNs. Furthermore, reduce the training time of network and improve the learning performance of DBNs by using ELM algorithm to optimize network. Thus, the recognition effect of our method is superior to the recognition effects of other methods. It can reduce the network's associative memory to the redundant information when using LBP features as the input of the network, and can extract more abundant and effective texture information than LBP and DBNs methods. The recognition result of our method is much better when using the ELM algorithm to fine-tune the network than using the traditional global optimization algorithm to fine-tune the entire network.

5. **Conclusions.** This paper combines the LBP method with DBNs method effectively, and uses the Extreme Learning Machine (ELM) algorithm in the fine-tuning stage of DBNs. Using the DBNs to learn the LBP features not only can make good use of the local texture feature of face image extracted by LBP method, but also has a stronger robustness to the illumination and small translation, and using the LBP feature as the input of DBNs can help the network to better understand the distribution of the image features and further reduce the network learning the bad features, as well as effectively avoid the intervention of excessive active factors through deep learning and automatic feature extraction of DBNs to the input data. This article uses ELM algorithm instead of the traditional global learning algorithm to fine-tune the established model in the fine-tuning stage which does not need to adjust parameters, can save time of network training and optimize the learning performance of DBNs, that is, the proposed method not only guarantees the accuracy of training and testing, but obviously enhances the training speed of deep learning.

The experiment results show that using extreme learning machine to fine-tune the network can significantly reduce the network learning time and improve the learning efficiency. Therefore, trying experimenting with effective combination of extreme learning machine and other algorithms in face recognition is our next research emphasis.

Acknowledgment. This work is partially supported by the National Natural Science Foundation of China under Grant (No. 61263019), the Fundamental Research Funds for the Gansu Universities. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] X. Tan, S. Chen, Z. H. Zhou et al., Face recognition from a single image per person: A survey, *Pattern Recognition*, vol.39, no.9, pp.1725-1745, 2006.

[2] C. Y. Xiong, D. T. Li and B. Y. Da, Feature extraction using LBP and PCA for face recognition, *Journal of South-Central University for Nationalities (Nat. Sci. Edition)*, vol.30, no.2, pp.75-79, 2011.

[3] S. M. Prakash and C. J. Kalpna, Face recognition using PCA, *International Journal of Artificial Intelligence & Knowledge Discovery*, vol.1, no.1, pp.25-28, 2011.

[4] I. L. Chellappa and T. Etemad, Linear discriminant analysis for recognition of human face images, *Journal of the Optical Society of America*, vol.14, no.8, pp.1724-1773, 1997.

[5] R. M. Chai and Z. J. Cao, Face recognition algorithm based on Gabor wavelet and deep belief networks, *Journal of Computer Applications*, vol.34, no.9, pp.2590-2594, 2014.

[6] T. Ojala, M. Pietikainen and T. Maenpaa, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.24, no.7, pp.971-987, 2002.

[7] L. Liu, Y. X. Xie, Y. M. Wei and S. Y. Lao, Survey of local binary pattern method, *Journal of Image and Graphics*, vol.19, no.12, pp.1696-1720, 2014.

[8] H. P. Sun and D. Y. Bi, Face recognition based on local binary pattern Gabor wavelet transform, *Computer Applications and Software*, vol.28, no.5, pp.247-248,261, 2011.

[9] B. C. Yin, W. T. Wang and L. C. Wang, Review of deep learning, *Journal of Beijing University of Technology*, vol.41, no.1, pp.48-59, 2015.

[10] G. Hinton, S. Osindero and Y. Teh, A fast learning algorithm for deep belief nets, *Neural Computation*, vol.18, no.7, pp.1527-1554, 2006.

[11] S. F. Liang, Y. H. Liu and L. C. Li, Face recognition under unconstrained based on LBP and deep learning, *Journal on Communications*, vol.35, no.6, pp.154-160, 2014.

[12] W. Zhang and W. W. Wang, Face recognition based on local binary pattern and deep learning, *Journal of Computer Applications*, vol.35, no.5, pp.1474-1478, 2015.

[13] G. B. Huang, H. Zhou, X. Ding et al., Extreme learning machine for regression and multiclass classification, *IEEE Trans. Systems, Man, and Cybernetics*, vol.42, no.2, pp.513-529, 2012.

[14] T. Ahonen, A. Hadid and M. Pietikainen, Face description with local binary patterns: Application to face recognition, *IEEE Trans. Pattern Analysis and Machine Intelligence (S0162-8828)*, vol.28, no.12, pp.2037-2041, 2006.

[15] B. Y. Cheng, Q. Liu and X. W. Li, Local perception-based intelligent building outline aggregation approach with back propagation neural network, *Neural Processing Letters*, vol.41, no.2, pp.273-292, 2015.

[16] R. Salakhutdinov, *Learning Deep Generative Models*, Ph.D. Thesis, Graduate Department of Computer Science, University of Toronto, Toronto, 2009.

[17] G. Hinton, *A Practical Guide to Training Restricted Boltzmann Machines*, University of Toronto, Toronto, 2010.

[18] Y. Bengio, P. Lamblin, D. Popovici et al., Greedy layer-wise training of deep networks, *Advances in Neural Information Processing Systems*, vol.19, pp.153-160, 2007.

[19] R. D. Hjelm, V. D. Calhoun, R. Salakhutdinov et al., Restricted Boltzmann machines for neuroimaging: An application in identifying intrinsic networks, *Neuroimage*, vol.96, pp.245-260, 2014.

[20] M. Z. Lin, *Face Recognition Research Based on Deep Learning*, Master Thesis, Dalian University of Technology, Dalian, 2013.