ICIC Express Letters
Volume **10**, Number **10**, October **2016**

ICIC International ©2016 ISSN 1881-803X
pp. 2459–2466

A NOVEL UNSUPERVISED CONVOLUTIONAL NETWORK BASED ON GABOR AND $(2D)^2$PCA FOR FEATURE EXTRACTION AND RECOGNITION

Ruru Lu[1], Min Jiang[1], Jun Kong[1,2], Shengwei Tian[2]
and Yilihamu Yaermaimaiti[2]

[1]Key Laboratory of Advanced Process Control for Light Industry (Ministry of Education)
Jiangnan University
No. 1800, Lihu Avenue, Wuxi 214122, P. R. China
minjiang@jiangnan.edu.cn

[2]College of Electrical Engineering
Xinjiang University
No. 14, Shengli Road, Urumqi 830046, P. R. China

Received March 2016; accepted June 2016

Abstract. *Feature extraction and recognition are challenging tasks in computer vision, especially for distortion data. As an effective method of feature extraction, two-directional two-dimensional Principal Component Analysis ($(2D)^2$PCA) has been widely used for its good performance. However, features extracted by $(2D)^2$PCA essentially are low-level and sensitive to distortions. To solve this problem, a new unsupervised deep learning network is proposed in this paper. It is based on a convolutional structure and can extract more useful multi-level features. In the proposed new network, 2D Gabor filters and $(2D)^2$PCA simply denoted as GB$(2D)^2$PCA are applied to learning the multi-stage convolutional filters, which overcomes the drawbacks of convolutional networks. Furthermore, binary hashing and block-wise histograms are used to compute output features, and the obtained output features are used to train Linear Support Vector Machine (LinearSVM) for recognition. Finally, face recognition is applied to verifying the effectiveness of the proposed network. And the effectiveness of feature extraction and recognition is demonstrated by experiments on several benchmark databases including distortion data.*

Keywords: Deep learning networks, Convolutional networks, GB$(2D)^2$PCA, Feature extraction

1. Introduction.

Feature extraction is the core work for improving quality and efficiency of an algorithm in computer vision tasks (e.g., face recognition, object recognition and image segmentation). Traditional hand-crafted feature extraction methods, such as SIFT, HOG and LBP, have been used in specific tasks. However, hand-crafted features are always low-level and almost depended on prior knowledge. It is difficult to take advantage of big data and be used in new tasks without learning new domain knowledge. Therefore, feature extraction and recognition are still challenging tasks in computer vision.

Linear Discriminant Analysis (LDA) and Principal Component Analysis (PCA) [1] are two classical techniques widely used in computer vision. Fisherface and Eigenface [2] are two famous face recognition methods based on those two techniques. Nevertheless, most of the LDA-based methods have the small sample size problem. Besides, PCA-based methods previously transform the 2D matrices into 1D vectors, which often results in a high-dimensional vector space. Two-directional two-dimensional PCA ($(2D)^2$PCA) [3], as a variation of PCA, usually learns more expressive features and is more computationally efficient. However, $(2D)^2$PCA is sensitive to distortion caused by illumination, pose and expression. Moreover, $(2D)^2$PCA could only capture low-level features, which cannot represent more abstract semantics of data.

Recently, it has been paid wide attention to deep learning for its ability of feature learning. Generally speaking, deep learning is composed of multiple stacking processing layers to automatically learn representations of data with multiple levels of abstraction, which remedies the limitation of hand-crafted features [4]. It brings dramatic improvements in many domains. For instance, with stacking Restricted Boltzmann Machines (RBMs) [5], Deep Neural Networks (DNNs) perform much better than traditional neural networks. One of the most representative deep architectures is Convolutional Networks (ConvNets) [6], which is more suitable for image-related tasks. In ConvNets, each stage comprises a convolutional filters layer, a nonlinearity layer and a feature pooling layer. However, for one thing, ConvNets obtain excellent results only if their architecture is deep enough; for another, although GPU is used to accelerate, it also leads to high computation to train such a deep network by using Stochastic Gradient Descent (SGD) in supervised mode. Variations of ConvNets have been proposed in the past few years. Several variations, including sparse coding [7] and convolutional versions of RBMs [8], employ unsupervised learning methods to pre-train in each stage and SGD method to fine-tune, which reduce the number of labeled data and achieve good performance on several vision tasks.

The initial motivation of our research is to reduce apparent differences between (2D)^{2}PCA and ConvNets. Inspired by the mentioned works above, in this paper, we employ very basic operations to emulate the processing layers in a typical ConvNets: (2D)^{2}PCA is selected as the convolution filters in each stage; simple binary hashing is chosen as the nonlinear layer; block-wise histograms of the binary codes are used as the feature pooling layer, which is considered as the final output features. The proposed multi-stage network is named as (2D)^{2}PCANet. Compared with the traditional feature extraction method of (2D)^{2}PCA, our feature extraction method is a multi-stage unsupervised convolutional network, which ensures the extracted features are more beneficial to represent abstract semantics of the data. It also should be noted that our unsupervised convolutional network does not need to learn convolutional filters by iteration, which overcomes the drawbacks of ConvNets. Moreover, in order to further increase the robustness of (2D)^{2}PCA features against distortion, we replace the (2D)^{2}PCA filters with GB(2D)^{2}PCA filters which are learned by 2D Gabor feature extraction method along with (2D)^{2}PCA, called GB(2D)^{2}PCANet. For the best reason that Gabor can optimally localize in the space and frequency domains [9], we conduct experiments on several benchmark databases to verify the effectiveness of GB(2D)^{2}PCANet, and the results demonstrate the effectiveness of our multi-stage network for feature extraction and recognition.

The rest of work is organized as follows. Section 2 describes the proposed network. Section 3 presents experimental results and analysis. Section 4 concludes the paper.

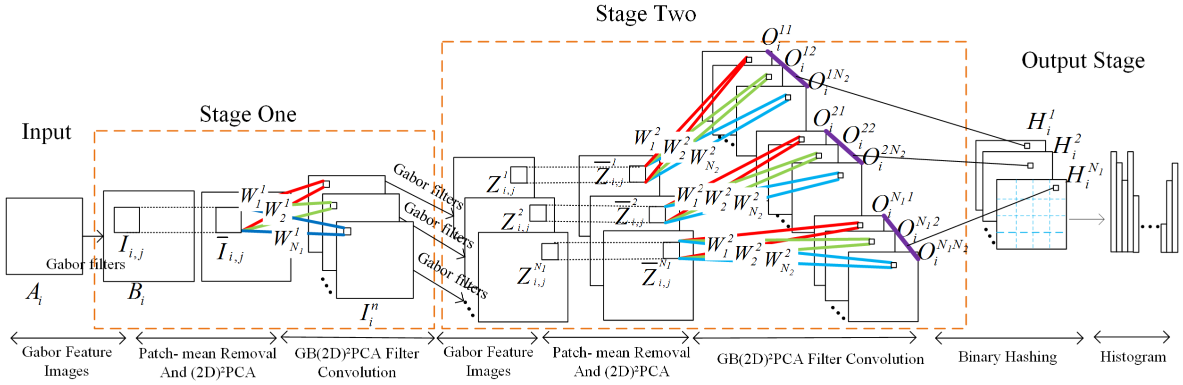


FIGURE 1. The framework of the proposed GB(2D)^{2}PCANet

2. **GB(2D)^{2}PCANet.** Different from other deep learning models mentioned above, the proposed network is composed of several feature extraction stages and one nonlinear output stage. To extract more discriminative features, output maps of all cascaded feature extraction stages serve as the input of the nonlinear output stage, which is the main difference with others. In the feature extraction stage, GB(2D)^{2}PCA is used to learn the convolutional filters. Original inputs convoluted with the learned convolutional filters produce a set of feature maps, which serve as inputs to the next feature extraction stage. Next, the outputs of all cascaded feature extraction stages are used as the inputs of nonlinear output stage. Then, in nonlinear output stage, binary hashing and block-wise histograms, instead of (Rectified Linear Unit) ReLU function and Max-Pooling [6], are employed to compute the final features. Finally, the final output features are sent to train Linear Support Vector Machine (LinearSVM) classifier for recognition. Figure 1 shows the proposed network with two feature extraction stages and one output stage. Given N input training images $\{\boldsymbol{A}_i\}_{i=1}^N$, each image size is $p \times q$. Then, we describe each stage precisely.

2.1. **The first feature extraction stage.** The characteristics of 2D Gabor filters are invariance to scale, translation and less sensitive to distortion in illumination, expression and noise [9]. Thus, the Gabor filters are chosen to represent the original input images. In the spatial domain, 2D Gabor filters are acquired by modulating a Gaussian kernel function with a sinusoid plane wave, defined as [10]:

$$G(x,y) = \frac{f^2}{\pi\gamma\eta}\exp\left(-\left(\frac{f^2}{\gamma^2}x'^2 + \frac{f^2}{\eta^2}y'^2\right)\right)\exp\left(j2\pi f x'\right), \tag{1}$$

where $x' = x\cos\theta + y\sin\theta$, $y' = -x\sin\theta + y\cos\theta$, f is the central frequency of the filter, γ and η correspond to the two perpendicular axes of the Gaussian, and θ is the rotation angle.

In order to extract useful features from images, a group of Gabor filters with different scales and orientations is usually required. GB(2D)^{2}PCANet gets feature images by using forty Gabor filters with five scales and eight orientations. In our method, we set $\gamma = \eta = \sqrt{2}$. Since adjacent pixels are generally highly correlated, we reduce information redundancy by downsampling the feature images with a factor of $d = 4$ [10]. For the i-th input image $\boldsymbol{A}_i$, we define the corresponding feature matrix as $\boldsymbol{B}_i \in R^{s\times t}$, where $s = \frac{p\times q}{d^2}$ is the number of pixels after downsampling and $t = 40$ is the number of Gabor filters. As a result, the whole feature matrix $\{\boldsymbol{B}_i\}_{i=1}^N$ can be obtained for all input images $\{\boldsymbol{A}_i\}_{i=1}^N$.

To some extent, local receptive fields and shared weights in ConvNets ensure invariance to shift, distortions and scale [6]. Inspired by the architecture, by taking an $l_1\times l_2$ patch of the i-th feature image $\boldsymbol{B}_i$ at every $b = 1$ pixel, we can obtain $m\times n$ patches for each feature image, where m is $\lceil\frac{s-l_1}{b}\rceil + 1$ and n is $\lceil\frac{t-l_2}{b}\rceil + 1$. All the patches form a matrix denoted as $\boldsymbol{b}_i = [\boldsymbol{b}_{i,1}, \boldsymbol{b}_{i,2}, \ldots, \boldsymbol{b}_{i,mn}]$, where $\boldsymbol{b}_{i,j}$ indicates the j-th patch in $\boldsymbol{B}_i$. Inspired by the idea of local contrast normalization [11], we subtract the mean of each patch and obtain matrix $\bar{\boldsymbol{b}}_i = [\bar{\boldsymbol{b}}_{i,1}, \bar{\boldsymbol{b}}_{i,2}, \ldots, \bar{\boldsymbol{b}}_{i,mn}] \in R^{l_1\times mnl_2}$, where $\bar{\boldsymbol{b}}_{i,j}$ is the j-th mean-removed patch in $\boldsymbol{b}_i$. By assembling $\bar{\boldsymbol{b}}_i$ after all feature images $\{\boldsymbol{B}_i\}_{i=1}^N$ are constructed in the same way, we form a large matrix: $\boldsymbol{I} = [\bar{\boldsymbol{b}}_{1,1}, \ldots, \bar{\boldsymbol{b}}_{1,mn}, \bar{\boldsymbol{b}}_{2,1}, \ldots, \bar{\boldsymbol{b}}_{2,mn}, \ldots, \bar{\boldsymbol{b}}_{N,1}, \ldots, \bar{\boldsymbol{b}}_{N,mn}] \in R^{l_1\times Nmnl_2}$. For convenient description, we rewrite $\boldsymbol{I}$ as the concatenation of vectors with successive index, i.e., $\boldsymbol{I} = [\bar{\boldsymbol{I}}_1, \bar{\boldsymbol{I}}_2, \ldots, \bar{\boldsymbol{I}}_k, \ldots, \bar{\boldsymbol{I}}_{Nmn}]$. Here, $\bar{\boldsymbol{I}}_k$ is the j-th mean-removed patch in image $\boldsymbol{B}_i$, $k = (i-1)\times m\times n + j$.

Compared with PCA, (2D)^{2}PCA can learn more features. Furthermore, it is more efficient in computation. The reasons depend on two aspects. One is that (2D)^{2}PCA uses image matrix to construct the covariance matrix directly. The other is that it is employed in the direction of the row and column simultaneously [3]. Thus, we use (2D)^{2}PCA to select the most discriminative features of the Gabor space ulteriorly.

Suppose N_1 is the number of the convolutional filters in the first stage. For each $l_1{\times}l_2$ patch $\bar{\boldsymbol{I}}_i$, from the row direction of patch $\bar{\boldsymbol{I}}_i$, we project $\bar{\boldsymbol{I}}_i$ onto $\boldsymbol{X}_{l_2\times N_1}(N_1{\leq}l_2)$ by: $\boldsymbol{E} = \bar{\boldsymbol{I}}_i\boldsymbol{X}{\in}R^{l_1\times N_1}$. And the row covariance matrix $\boldsymbol{G}^{row}$ is given by:

$$\boldsymbol{G}^{row} = \frac{1}{Nmn}\sum\nolimits_{i=1}^{Nmn}\left(\bar{\boldsymbol{I}}_i - \bar{\boldsymbol{I}}\right)^T\left(\bar{\boldsymbol{I}}_i - \bar{\boldsymbol{I}}\right), \tag{2}$$

where $\bar{\boldsymbol{I}} = \frac{1}{Nmn}\sum_{i=1}^{Nmn}\bar{\boldsymbol{I}}_i$ is the mean image of all training images. It has been verified that the optimal projection axis $\widehat{\boldsymbol{X}}$ is composed by the orthonormal eigenvectors $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_{N_1}$ of $\boldsymbol{G}^{row}$, corresponding to the N_1 largest eigenvalues, $\widehat{\boldsymbol{X}} = [\boldsymbol{X}_1, \ldots, \boldsymbol{X}_{N_1}]$.

Similarly, from the column direction of patches, we project $\bar{\boldsymbol{I}}_i$ onto $\boldsymbol{Y}_{l_1\times N_1}(N_1{\leq}l_1)$ by: $\boldsymbol{F} = \boldsymbol{Y}^T\bar{\boldsymbol{I}}_i{\in}R^{N_1\times l_2}$. And the column covariance matrix is given as follows:

$$\boldsymbol{G}^{col} = \frac{1}{Nmn}\sum\nolimits_{i=1}^{Nmn}\left(\bar{\boldsymbol{I}}_i - \bar{\boldsymbol{I}}\right)\left(\bar{\boldsymbol{I}}_i - \bar{\boldsymbol{I}}\right)^T. \tag{3}$$

The optimal projection axis $\widehat{\boldsymbol{Y}} = [\boldsymbol{Y}_1, \boldsymbol{Y}_2, \ldots, \boldsymbol{Y}_{N_1}]$, where $\boldsymbol{Y}_i$ shows the orthogonal eigenvectors of $\boldsymbol{G}^{col}$ corresponding to the N_1 largest eigenvalues.

Then, we denote $\boldsymbol{W}_n^1 = \boldsymbol{Y}_n\boldsymbol{X}_n^T{\in}R^{l_1\times l_2}$ as the learned convolutional filters in the first stage, where $n = 1, 2, \ldots, N_1$. In the first stage, for each original input $\boldsymbol{A}_i$, the outputs of the n-th filter are:

$$\boldsymbol{I}_i^n = \boldsymbol{A}_i{*}\boldsymbol{W}_n^1{\in}R^{p\times q}, \quad i = 1, 2, \ldots, N, \tag{4}$$

where $*$ is 2D convolution. $\boldsymbol{I}_i^n$ is convolutional output of the i-th image $\boldsymbol{A}_i$ with the n-th filter. Before convolution, to make $\boldsymbol{I}_i^n$ with the same size of $\boldsymbol{A}_i$, the boundary of $\boldsymbol{A}_i$ is padded with zero. For each input $\boldsymbol{A}_i$, N_1 feature maps $\{\boldsymbol{I}_i^n\}_{n=1,2,\ldots,N_1}$ are produced in the first stage.

2.2. **The second feature extraction stage.** For all input images $\{\boldsymbol{A}_i\}_{i=1}^N$, the outputs of the first stage $\{\boldsymbol{I}_i^n\}_{n=1,2,\ldots,N_1,i=1,2,\ldots,N}$ are used as the original input to the second stage. Like the first stage, after the operation of Gabor filters and downsampling, the output of the i-th image with the n-th filter in the first stage $\boldsymbol{I}_i^n$ is further converted into the corresponding feature matrix $\boldsymbol{Z}_i^n$. By collecting all patches of $\boldsymbol{Z}_i^n$ and subtracting the mean of each patch, $\bar{\boldsymbol{Z}}_i^n = \left[\bar{\boldsymbol{Z}}_{i,1}^n, \bar{\boldsymbol{Z}}_{i,2}^n, \ldots, \bar{\boldsymbol{Z}}_{i,mn}^n\right] \in R^{l_1\times mnl_2}$ is formed, where $\bar{\boldsymbol{Z}}_{i,j}^n$ is the j-th mean-removed patch in $\boldsymbol{Z}_i^n$. Then, by constructing all the outputs from the n-th filter $\{\boldsymbol{I}_i^n\}_{n=1,2,\ldots,N}$ in the same way, all mean-removed patches are further collected and $\boldsymbol{Z}^n = \left[\bar{\boldsymbol{Z}}_1^n, \bar{\boldsymbol{Z}}_2^n, \ldots, \bar{\boldsymbol{Z}}_N^n\right] \in R^{l_1\times Nmnl_2}$ is defined. Finally, by concatenating all the vectors in $\boldsymbol{Z}^n$ for all the filter outputs, $\boldsymbol{Z} = \left[\boldsymbol{Z}^1, \boldsymbol{Z}^2, \ldots, \boldsymbol{Z}^{N_1}\right] \in R^{l_1\times Nmnl_2N_1}$ is obtained. Suppose N_2 is the number of the convolutional filters in the second stage. Repeat the same process as the first stage, we can compute the outputs of the n_2-th filter in the second stage:

$$\boldsymbol{O}_i^{nn_2} = \boldsymbol{I}_i^n{*}\boldsymbol{W}_{n_2}^2{\in}R^{p\times q}, \tag{5}$$

where $n_2 = 1, 2, \ldots, N_2$, $\boldsymbol{O}_i^{nn_2}$ is the convolutional output of $\boldsymbol{I}_i^n$ with the n_2-th filter in the second stage. For each original input $\boldsymbol{A}_i$, $N_1{\times}N_2$ feature maps $\{\boldsymbol{O}_i^{nn_2}\}_{n=1,2,\ldots,N_1,n_2=1,2,\ldots,N_2}$ are produced in the second stage. The above process can be simply repeated in the same way if more feature extraction stages are needed.

2.3. **The output stage: Binary hashing and block-wise histograms.** In the second stage, each input image $\boldsymbol{I}_i^n$ produces N_2 outputs. We binarize these N_2 outputs by $H(\boldsymbol{O}_i^{nn_2})$, where the value of $H(\cdot)$ is one for positive inputs and zero for other input cases.

The binary bits at the same pixel location in all the N_2 output maps compose a binary vector, which can be viewed as a decimal number. Then, one can obtain an integer-valued image $\boldsymbol{H}_i^n$ by the following formula:

$$\boldsymbol{H}_i^n = \sum\nolimits_{n_2=1}^{N_2} 2^{n_2-1}H\left(\boldsymbol{O}_i^{nn_2}\right), \tag{6}$$

where 2^{n_2-1} is weight for the N_2 binary bits.

Each integer-valued image $\boldsymbol{H}_i^n$, $n = 1, 2, \ldots, N_1$, is parted into K blocks (each block size is $[b_1\ b_2]$), which can be either overlapping (the block overlap ratio is α) or non-overlapping. In each block, the histogram is calculated by using the decimal values. By concatenating all the K histograms, one can drive a vector denoted as $Bhist\left(\boldsymbol{H}_i^n\right)$. After above encoding processing, a group of block-wise histograms:

$$\left[Bhist\left(\boldsymbol{H}_i^1\right), \ldots, Bhist\left(\boldsymbol{H}_i^{N_1}\right)\right]$$

are the final features of the input image $\boldsymbol{A}_i$.

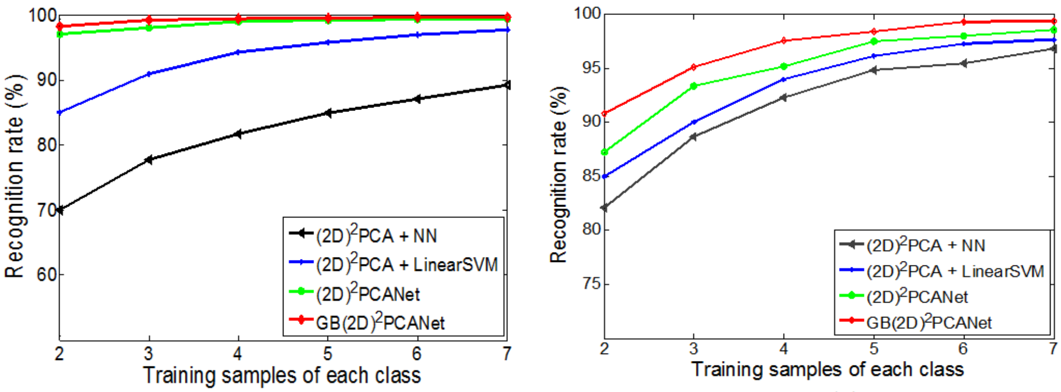
Eventually, the final output features of the GB(2D)^{2}PCANet are sent to train LinearSVM classifier for recognition.

The parameters of GB(2D)^{2}PCANet include the number of stages, the patch size l_1, l_2, the number of filters in each stage N_1, N_2, the block size for local histograms in the output stage b_1, b_2 and the block overlap ratio α.

3. **Experiments.** In the following experiments, we apply two-stage (two feature extraction stages) GB(2D)^{2}PCANet to several benchmark databases, such as XM2VTS, ORL and AR. The multi-stage network whose convolutional filters are pre-fixed is inspired by (2D)^{2}PCA. Therefore, (2D)^{2}PCA with LinearSVM classifier and the nearest neighbor (NN) classifier will be employed for comparison to verify the effectiveness of the multi-stage network. (2D)^{2}PCANet will also be discussed to check the effect of Gabor in the GB(2D)^{2}PCANet. For a fair comparison, all experimental configurations keep fixed and all images are converted to grayscale map. The following experimental results show that two-stage GB(2D)^{2}PCANet leads to excellent results in many ways.

3.1. **Insensitivity to the training sample size.** In this part, several experiments are conducted on XM2VTS and ORL databases to study the effect of the training sample size. XM2VTS contains 295 subjects. Each subject provides 8 different images. Images are cropped to 55 × 51. ORL includes 40 subjects. Each subject provides 10 different images. Images are resized to 32 × 32. We randomly select S ($S = 2, 3, 4, 5, 6, 7$) samples of each class as training samples, and the rest are used for testing. We take the patch size $l_1 = l_2 = 5$, the number of filters in each stage $N_1 = N_2 = 5$, the block size $b_1 = b_2 = 7$, the block overlap ratio $\alpha = 0.5$ on XM2VTS and $l_1 = l_2 = 7$, $N_1 = N_2 = 5$, $b_1 = b_2 = 5$, $\alpha = 0.5$ on ORL.

The experimental results are given in Figure 2. Based on two databases, one can see that (2D)^{2}PCA with LinearSVM classifier outperforms NN classifier. The reason is that NN classifier is not discriminative enough to well select the relevant samples. Besides,



(a) XM2VTS

(b) ORL

Figure 2. Recognition rate of different methods under different training samples

compared with (2D)^{2}PCA, multi-stage networks proposed in this paper perform better for the deep architectures, which can learn more abstract and more hierarchical features. Meanwhile, GB(2D)^{2}PCANet yields better performance than (2D)^{2}PCANet obviously, demonstrating the effectiveness of Gabor. Moreover, from Figure 2(a), it keeps perfect performance even though only 3 training samples are selected for each class on XM2VTS. Thus, we can draw a conclusion that GB(2D)^{2}PCANet is less-sensitive to the number of training samples.

3.2. **Insensitivity to distortion in illumination, expression and occlusion.** In this section, we conduct experiment on AR databases to study the effect of the distortion. AR includes over 4000 images from 126 subjects. These images contain different illumination, expression and occlusion conditions. Each subject has 26 images in two sessions. Each section has 13 images. The second session has the same conditions as the first session. In this experiment, we choose a subset of the database consisting of 50 males and 50 females. All images are resized to 60 × 43. For all images of each subject, images of neural expression and frontal illumination in the first session are selected as the training samples. The rest 19 images are used as test set T. We further divide T into four test sets according to the possible variations, namely *Exps* (expression), *Illum* (illumination), *Occlus* (occlusion) and *Illum+Occlus* (illumination-plus-occlusion). We take $l_1 = l_2 = 4$, $[N_1\ N_2]$ as [3 4], $[b_1\ b_2]$ as [3 2] and $\alpha = 0.7$.

Table 1 lists the results of different methods. The experimental results are consistent with those on XM2VTS and ORL datasets: GB(2D)^{2}PCANet outperforms other methods on all test sets. Obviously, when there exists occlusion, the performance of (2D)^{2}PCA decreases significantly because (2D)^{2}PCA is sensitive to distortion. However, GB(2D)^{2}PCANet overcomes the drawback and is more effective in dealing with distortion. It incorporates the virtues of Gabor filters, that is, less sensitive to distortion in illumination, expression and noise. Especially, we examine the performance of Gabor with LinearSVM. Obviously, Gabor has played a large role in the robustness against expression, illumination and occlusion.

Table 1. Recognition rates of different methods on AR. The numbers in bold denote the best assessment value.

Test sets	*Illum*	*Exps*	*Occlus*	*Illum+Occlus*	*T*
Gabor+LinearSVM	96.67%	95.25%	91.00%	97.13%	91.16%
(2D)^{2}PCA+NN	77.67%	67.25%	18.75%	20.62%	39.05%
(2D)^{2}PCA+LinearSVM	87.33%	86.00%	27.50%	23.87%	47.74%
(2D)^{2}PCANet	**100%**	99.00%	98.25%	96.63%	98.00%
GB(2D)^{2}PCANet	**100%**	**99.50%**	**99.00%**	**97.38%**	**98.47%**

3.3. **Impact of parameters.** Next, the impacts of two parameters of GB(2D)^{2}PCANet to the recognition performance are examined on XM2VTS, ORL and AR databases. One is the block size for local histograms b_1, b_2. The other is the block overlap ratio α. Besides, in order to examine the influence of b_1, b_2 on robustness against image occlusion, the *Occlus* test set of AR is tested. Figure 3 indicates the relationship between the values of parameters and the recognition rates.

Impact of the block size: The block size $[b_1\ b_2]$ is varied from [3 3] to [15 15] and other parameters are fixed. From Figure 3(a), in general, the recognition rates decrease when the block size $[b_1\ b_2]$>[7 7] on the three databases. Moreover, on *Occlus* test set of AR, GB(2D)^{2}PCANet achieves excellent results when the block size tends to be small.

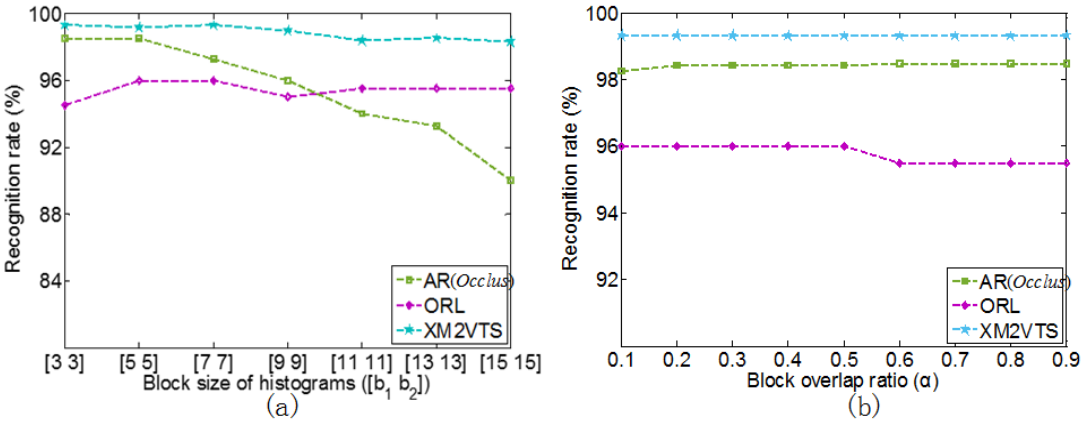


Figure 3. Recognition rates with the changing parameters

Therefore, we can draw a conclusion that small block size is less sensitive to occlusion and suggest that the value of the block size is less than or equal to [7 7].

Impact of the block overlap ratio: To examine the effect of the block overlap ratio α, we vary α from 0.1 to 0.9 while other parameters are fixed. From Figure 3(b), the biggest difference between the highest and lowest recognition rates is only 0.5% on ORL. Thus, GB(2D)^{2}PCANet is quite insensitive to α.

4. **Conclusion.** In this paper, a novel unsupervised deep convolutional network is proposed for feature extraction and recognition, which can extract more useful multi-level features. The network is composed of two feature extraction stages and one nonlinear output stage. The outputs of all cascaded feature extraction stages are used as the input of nonlinear processing stage, which is the main difference with other deep learning networks. In each feature extraction stage, GB(2D)^{2}PCA is used to learn convolutional filters, which overcomes the drawbacks of convolutional networks. In nonlinear output stage, binary hashing and block-wise histograms are employed to compute output features, which are sent to train LinearSVM for recognition. Experimental results have shown that GB(2D)^{2}PCANet with two feature extraction stages is quite effective for feature extraction and recognition. In the future, we plan to apply ensemble learning approaches to our feature extraction method for dealing with much larger databases.

Acknowledgment. This work is partially supported by the National Natural Science Foundation of China (61201429, 61362030), the Project Funded by China Postdoctoral Science Foundation (2015M581720), and the Science and Research Key Project of Xinjiang Uygur Autonomous Region University (XJEDU2012I08).

REFERENCES

[1] F. Z. Chelali, A. Djeradi and R. Djeradi, Linear discriminant analysis for face recognition, *International Conference on Multimedia Computing and Systems*, 2009.

[2] M. Sharkas and M. Abou Elenien, Eigenfaces vs. Fisherfaces vs. ICA for face recognition; A comparative study, *International Conference on Signal Processing*, pp.914-919, 2008.

[3] D. Zhang and Z. Zhou, (2D)^{2}PCA: Two-directional two-dimensional PCA for efficient face representation and recognition, *Neurocomputing*, vol.69, no.1, pp.224-231, 2005.

[4] Y. LeCun, Y. Bengio and G. Hinton, Deep learning, *Nature*, vol.521, no.7553, pp.436-444, 2015.

[5] S. Zhou, Q. Chen and X. Wang, Deep networks for online handwriting Chinese character recognition, *ICIC Express Letters*, vol.9, no.6, pp.1783-1789, 2015.

[6] A. Krizhevsky, I. Sutskever and G. E. Hinton, Imagenet classification with deep convolutional neural networks, *Advances in Neural Information Processing Systems*, vol.25, no.2, pp.1097-1105, 2012.

[7] J. Liu, B. Y. Liu and H. Q. Lu, Detection guided deconvolutional network for hierarchical feature learning, *Pattern Recognition*, vol.48, no.8, pp.2645-2655, 2015.

[8] H. Lee, R. Grosse and R. Ranganath, Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations, *International Conference on Machine Learning*, 2009.

[9] L. Shen and L. Bai, A review on Gabor wavelets for face recognition, *Pattern Analysis and Applications*, vol.9, nos.2-3, pp.273-292, 2006.

[10] M. Saeed, A. Ali and S. Hadi, Face recognition using Gabor-based direct linear discriminant analysis and support vector machine, *Computers and Electrical Engineering*, vol.39, no.3, pp.727-745, 2013.

[11] Y. LeCun, K. Kavukcuoglu and C. Farabet, Convolutional networks and applications in vision, *Proc. of IEEE International Symposium on Circuits and Systems (ISCAS)*, vol.14, no.5, pp.253-256, 2010.