Multimed Tools Appl (2018) 77:12919–12934
DOI 10.1007/s11042-017-4923-3

2DPCANet: a deep leaning network for face recognition

Dan Yu[1] · Xiao-Jun Wu[1]

Received: 5 April 2016 / Revised: 31 March 2017 / Accepted: 6 June 2017 /
Published online: 4 July 2017

Abstract This paper proposes a two-dimensional principal component analysis network (2DPCANet), which is a novel deep learning network for face recognition. In our architecture, 2DPCA is employed to learn the filters of multistage layers, and then we exploit binary hashing and the block-wise histograms to generate the local features. Support vector machine (SVM) and extreme learning machine (ELM) are adopted as the classifier. The experimental results obtained on the facial database YALE, XM2VTS, AR, LFW-a, FERET and Extended Yale B show that the recognition performance of 2DPCANet is superior to other reported methods. Another interesting discovery on ELM classifier is that the advantage of ELM being simple and fast will disappear when it is applied to large databases.

Keywords Face recognition · Deep learning · 2DPCA · ELM

1 Introduction

Face recognition is one of the most important research tasks in computer vision and pattern recognition. To choose the right features plays the key role in a recognition system. And many researches show that the features of the best performing recognition models are learned unsupervisedly from raw data.

Recently, Deep Neural Networks (DNNs) especially Convolutional Neural Networks (CNNs) [1], constructed by convolutional and pooling operations, have received intensive attentions, because they can automatically and simultaneously discover low level and high level features, and achieve astonishing results on various recognition tasks. For example, CNNs have recently shown record-breaking results on the large-scale visual recognition challenge (ILSVRC2012) [14] and the best accuracy on the MNIST digit-recognition task [5] and so on. But there are a lot of parameters in a deep CNN to be tuned given the enormous data, which leads to high computational load and requires much time and storage space.

✉ Xiao-Jun Wu
wu_xiaojun@jiangnan.edu.cn

[1] School of Internet of Things Engineering, Jiangnan University, No.1800,Lihu Avenue, Wuxi 214122, China

To solve this problem, Chan et al. [4] proposed PCANet which adopts PCA to learn the filters of the convolutional layers and exploits binary hashing and the block-wise histograms to generate the local features. The experiments show the outstanding performance of PCANet for some recognition tasks. There are many variants of PCANet, such as DLANet [7], which replaces PCA with DLA (discriminative locality alignment network) for scene classification and exhibits excellent performance, and DCTNet [16], which is data-independent network, and so on.

PCA [2] is based on 1D vectors, which means that we must first transform the 2D matrices into 1D vectors. However, concatenating 2D matrices into 1D vectors often leads to a high-dimensional vector space, which leads to much storage space requirement and it is very time-consuming to compute the eigenvectors. Even worse, the intrinsic 2D structure of an image matrix is removed. To solve these problems, Yang et al. [22] proposed two-dimensional principal component analysis (2DPCA), which directly computes eigenvectors based on 2D matrices. It computes the eigenvectors more efficiently and maintains the intrinsic 2D structure of an image matrix since it doesn't need transform matrix to vector.

In this paper, since 2DPCA has many advantages over conventional PCA and the performance of PCANet on face recognition task is pretty good, we deploy the structure of PCANet to learn the local features but explore filters learnt by 2DPCA and implement it for face recognition. Moreover, we compare support vector machine (SVM) with extreme learning machine (ELM) [9] as the final classifier with the deep features by PCANet and 2DPCANet. To evaluate the effectiveness of 2DPCANet on face recognition, we compare it with other face recognition algorithms on the facial databases YALE, XM2VTS, AR, LFW-a, FERET and Extended Yale B.

The rest of the paper is organized as follows. In Section 2, we review related work on 2DPCA and ELM. Then we introduce the proposed 2DPCANet algorithm in detail in Section 3. Section 4 shows the experimental results on the facial databases YALE, XM2VTS, LFW-a, AR, FERET and Extended Yale B. We conclude the paper in Section 5.

2 Related work

2.1 Two-dimensional principal component analysis

Suppose A is an $m \times n$ random image matrix, let $X \in R^{n \times d}$ be a matrix with orthonormal columns, $n \geq d$. Projecting A onto X yields an m by d matrix $Y = AX$.

Define a matrix $G = E[(A - E[A])^T(A - E[A])]$, called the image covariance matrix, which is easy to verify to be a $n \times n$ nonnegative definite matrix. Suppose that there are N training face images $A_k \in R^{m \times n}(k = 1, 2, \ldots, N)$, and the average image is defined as

$$\overline{A} = \frac{1}{N}\sum_k A_k \tag{1}$$

Then G can be evaluated by

$$G = \frac{1}{N}\sum_{k=1}^{N}\left(A_k - \overline{A}\right)^T\left(A_k - \overline{A}\right) \tag{2}$$

The optimal projection matrix $X_{opt} = [X_1, \ldots, X_d]$ is composed by the d orthonormal eigenvectors of G corresponding to the first d largest eigenvalues.

Conventional 2DPCA is essentially working in the row direction of images, which needs more coefficients for image representation than PCA. Zhang et al. [23] proposed two-directional 2DPCA, which needs a much less coefficients for image representation than conventional 2DPCA.

Let $Z \in R^{m \times q}$ be a matrix with orthonormal columns. Projecting the random matrix A onto Z yields a q by n matrix $B = Z^T A$. Similarly, the image covariance matrix $G_t = E[(A - E[A])(A - E[A])^T]$, and the projection matrix Z_{opt} is composed by the orthonormal eigenvectors $Z_1, \ldots, Z_q$ of G_t corresponding to the first q largest eigenvalues, i.e. $Z_{opt} = [Z_1, \ldots, Z_q]$.

Finally, projecting the m by n image A onto X and Z simultaneously, yields a q by d matrix C

$$C = Z^T A X \tag{3}$$

2.2 Extreme learning machine

Given N training data and corresponding class labels $(A_i, t_i), i = 1, \ldots, N$, the main task of ELM is solving the linear equations $H\beta = T$, where H is the hidden layer matrix, β is the output matrix and $T$is the derived output matrix.

$$H = \begin{bmatrix} h(A_1) \\ h(A_2) \\ \vdots \\ h(A_N) \end{bmatrix}, \quad T = [t_1, t_2, \cdots, t_N]^T \tag{4}$$

The essence of ELM is that: the hidden nodes of ELM are generated randomly, and the output weight β can be obtained in different ways [10, 11, 20]. For example, it can be evaluated by the smallest norm least-squares solution [20]:

$$\beta = H^+ T \tag{5}$$

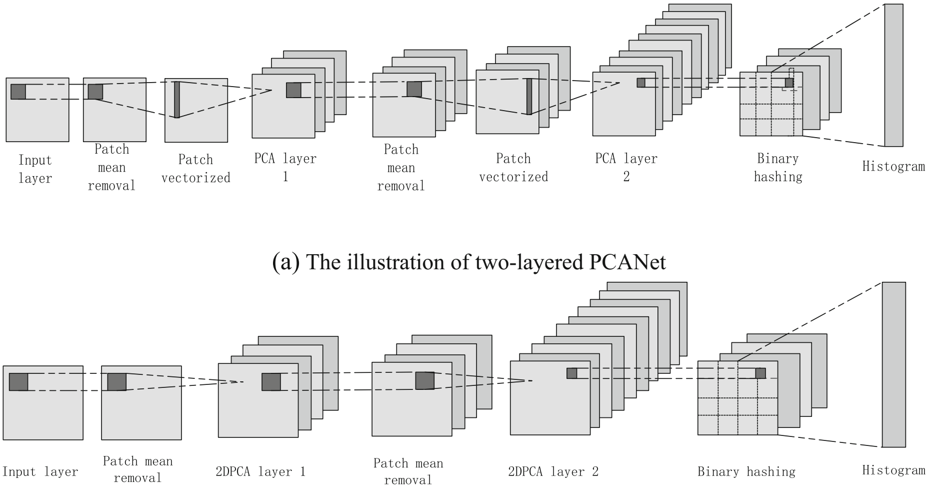
where H^+ is the Moore-Penrose generalized inverse of H.

3 2DPCANet

In this section, we introduce a simple deep learning network, 2DPCANet. In order to deal with the shortcoming of PCA in the PCANet [4], we adopt the structure of PCANet, but use the 2DPCA rather than the PCA as the filter banks. The difference between 2DPCANet and PCANet is shown in Fig.1. The main difference between 2DPCANet and PCANet is that PCANet use the PCA as the filter banks, which must transform 2D matrix into 1D vector before computing eigenvectors, while 2DPCANet uses the 2DPCA as filter banks, which directly computes eigenvectors of the so-called image covariance matrix without matrix-to-vector conversion.

3.1 The first 2DPCA layer

Given N training images $\{A_i\}_{i=1}^N$ of size $m \times n$, and we take the image patches of size $k \times k$. For the *ith* image A_i, we have data matrix $P_i = (p_{i,1}, p_{i,2}, \ldots, p_{i,mn})$ where $p_{i,j} \in R^{k \times k}$ denotes the



(a) The illustration of two-layered PCANet

(b) The illustration of two-layered 2DPCANet

Fig. 1 The difference between 2DPCANet and PCANet

jth patch in A_i. For normalization, each block will subtract its mean and then we obtain the normalized data matrix:

$$\overline{P}_i=\left(\overline{p}_{i,1},\overline{p}_{i,2},\dots,\overline{p}_{i,mn}\right) \tag{6}$$

where $\overline{p}_{i,j}\in R^{k\times k}$ is the mean-removed patch. For all images, we concatenate their normalized data matrices and put them together, then we have

$$\begin{aligned}P&=\left(\overline{P}_1,\overline{P}_2,\dots,\overline{P}_N\right)=\left(\overline{p}_{1,1},\overline{p}_{1,2},\dots,\overline{p}_{1,mn},\dots,\overline{p}_{N,1},\overline{p}_{N,2},\dots,\overline{p}_{N,mn}\right)\\&=(p_1,p_2,\dots,p_{Nmn})\end{aligned} \tag{7}$$

Assuming that the number of filters in layer i is D_i, and let $X\in R^{k\times D_1}$ be a matrix with orthonormal columns, $k\geq D_1$. In 2DPCA, for a matrix p_i in P, projecting the matrix p_i onto X yields a k by D_1 matrix $Y=p_iX$. Then the Eq.(2) could be rewritten as $G=\frac{1}{Nmn}\sum_{k=1}^{Nmn}(p_k-\overline{p})^T(p_k-\overline{p})$, where $\overline{p}$is the average data matrix of all data matrices of all training images. And the optimal value for the projection matrix X_{opt} is composed by the orthonormal eigenvectors $X_1,\dots,X_{D_1}$ of G corresponding to the D_1 largest eigenvalues, i.e. $X_{opt}=[X_1,\dots,X_{D_1}]$. Similarly, Let $Z\in R^{k\times D_1}$ be a matrix with orthonormal columns. Projecting the random matrix p_i onto Z yields a D_1 by k matrix $B=Z^Tp_i$. The image covariance matrix $G=\frac{1}{Nmn}\sum_{k=1}^{Nmn}(p_k-\overline{p})(p_k-\overline{p})^T$ and the projection matrix Z_{opt} is obtained by column-wisely stacking the eigenvectors $Z_1,\dots,Z_{D_1}$ of G_t corresponding to the D_1 largest eigenvalues, i.e. $Z_{opt}=[Z_1,\dots,Z_{D_1}]$.

The 2DPCA filters are therefore expressed as

$$W^1_{d_1} = Z_{d_1} X^T_{d_1}, \ \ d_1 = 1, 2, \ldots, D_1 \tag{8}$$

Then we have the output maps

$$C^1_{i,d_1} = A_i{}^* W^1_{d_1}, \ \ d_1 = 1, 2, \ldots, D_1 \tag{9}$$

where C^1_{i,d_1} is the output of the *ith* image about the *dth* filter. * is the 2D convolution with the zero-padding to make sure that the output map has the same size of the input image. Then the output maps are treated as the input of the second layer.

3.2 The second 2DPCA layer

With the input maps $\left\{C_{i,d_1}\right\}^{N,D_1}_{i,d_1=1,1}$, we have the mean-removed patches $P \in R^{k \times k \times ND_1 mn}$ and get the filters $W^2_{d_2}, d_2 = 1, 2, \ldots, D_2$ just as section 3.1. For the each input of the second layer, we will have output

$$C^2_{i,d_1,d_2} = C^1_{i,d_1}{}^* W^2_{d_2}, \ d_2 = 1, 2, \ldots, D_2 \tag{10}$$

Now we have $D_1 D_2$ output maps, if a deeper architecture is needed, we just need to repeat the above process.

3.3 Hashing and histogram

For each inputA_i, we have D_1 output maps in the first 2DPCA layer, and for each input of the second layer C^1_{i,d_1}, we have D_2 output maps $C^2_{i,d_1,1}, C^2_{i,d_1,2}, \ldots, C^2_{i,d_1,D_2}$. We binarize these output maps with Heaviside step function $H(\cdot)$

$$H(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases} \tag{11}$$

For each pixel, we regard the vector of D_2 binary bits as a decimal number, and convert the vector to an integer-value

Table 1 Recognition rate on YALE database

	2	3	4	5	6	7
PCANet	0.9133 ±0.0217	0.9458 ±0.0168	0.9705 ±0.0116	0.9895 ±0.0089	0.9853 ±0.0111	0.9867 ±0.0100
2DPCANet	0.9593 ±0.0145	0.9817 ±0.0090	0.9895 ±0.0089	0.9933 ±0.0073	0.9947 ±0.0040	0.9983 ±0.0050
PCANet(ELM)	0.9289 ±0.0217	0.9675 ±0.0180	0.9714 ±0.0217	0.9844 ±0.0158	0.9867 ±0.0157	0.9917 ±0.0145
2DPCANet(ELM)	0.9770 ±0.0130	0.9842 ±0.0114	0.9905 ±0.0085	0.9962 ±0.0111	0.9987 ±0.0065	0.9983 ±0.0050

Table 2 Performance on YALE database

	2		3		4		5		6		7	
	Training Time(s)	Testing Time(s)	Training Time(s)	Testing Time(s)	Training Time(s)	Testing Time(s)	Training Time(s)	Testing Time(s)	Training Time(s)	Testing Time(s)	Training Time(s)	Testing Time(s)
PCANet	4.08	0.17	6.11	0.17	7.43	0.16	9.82	0.17	11.08	0.16	13.16	0.16
2DPCANet	**3.58**	0.17	**5.23**	0.15	**6.62**	0.16	**8.50**	0.16	**10.00**	0.16	**11.97**	0.16
PCANet(ELM)	7.67	**0.06**	10.46	**0.06**	13.36	**0.07**	15.59	**0.07**	18.25	**0.07**	20.88	**0.08**
2DPCANet(ELM)	**7.53**	**0.07**	**10.07**	**0.07**	**12.63**	**0.08**	**15.01**	**0.08**	**17.74**	**0.09**	**20.00**	**0.09**

Table 3 Recognition rate on XM2VTS database

	2	3	4	5	6	7
PCANet	0.8787 ±0.0049	0.9332 ±0.0052	0.9586 ±0.0043	0.9714 ±0.0052	0.9785 ±0.0043	0.9814 ±0.0061
2DPCANet	0.9246 ±0.0044	0.9555 ±0.0032	0.9700 ±0.0030	0.9763 ±0.0014	0.9847 ±0.0021	0.9892 ±0.0054
PCANet(ELM)	0.8855 ±0.0068	0.9368 ±0.0034	0.9610 ±0.0055	0.9718 ±0.0072	0.9797 ±0.0054	0.9834 ±0.0044
2DPCANet(ELM)	0.9491 ±0.0052	0.9712 ±0.0020	0.9725 ±0.0025	0.9856 ±0.0031	0.9919 ±0.0023	0.9932 ±0.0032

$$F_{i,d_1} = \sum_{d_2=1}^{D_2} 2^{d_2-1} H\left(C^2_{i,d_1,d_2}\right) \tag{12}$$

Where F_{i,d_1} is an integer of range $\left[0, 2^{D_2-1}\right]$. For each $F_{i,d_1}(d_1 = 1, 2, \ldots, D_1)$, we partition it into B blocks. We compute the histogram with 2^{D_2}bins of the decimal values in each block, and concatenate all the Bhistograms into one vector and denotes as $hist\left(F_{i,d_1}\right)$. Finally we put all D_1 histograms together, we get

$$f_i = \left[hist\left(F^1_{i,1}\right), hist\left(F^1_{i,2}\right), \ldots, hist\left(F^1_{i,D_1}\right)\right]^T \in R^{2^{D_2}D_1B} \tag{13}$$

For N training images $\{A_i\}_{i=1}^N$, we get N features $\{f_i\}_{i=1}^N$ through a series of above processes.

3.4 Extreme learning machine and support vector machine

We get N features $\{f_i\}_{i=1}^N$ through a series of above processes, which are sent to extreme learning machine and support vector machine for classification. In the ELM, hidden nodes are generated randomly and only the weight vector between hidden and output nodes needs to be adjusted. Few parameters need to be adjusted, and thus the training can be very fast. However, the parameters of ELM is calculated directly via least square method, which makes its generalization not very good, especially for large database. SVM is regarded as one of the best classifiers, we use the extreme learning machine and support vector machine as classifiers to investigate the classification abilities of the two classifiers with the deep features.

3.5 The differences between the proposed 2DPCANet and "2DPCANet: Dayside aurora Classfication based on deep learning"

It should be noted that 2DPCANet was proposed in [13] to solve the problem of aurora image classification, whose main idea of replacing 2DPCA with PCA is the same. However, some different works are done in the current paper. Another work of this paper is that we have studied the performance of SVM and ELM based on deep features extracted by 2DPCANet. Since ELM is a controversial classifier in the field of pattern recognition, so we compare the classifiers ELM and SVM. Furthermore, we investigated the classified abilities of ELM and SVM with the deep features. Interesting observations on the performance of ELM has been

Table 4 Performance on XM2VTS database

	2		3		4		5		6		7	
	Training Time(s)	Testing Time(s)	Training Time(s)	Testing Time(s)	Training Time(s)	Testing Time(s)	Training Time(s)	Testing Time(s)	Training Time(s)	Testing Time(s)	Training Time(s)	Testing Time(s)
PCANet	753.07	0.27	1163.18	0.26	1466.35	0.27	1821.72	0.26	2311.89	0.26	2585.24	0.26
2DPCANet	**200.84**	0.29	**269.16**	0.26	**393.02**	0.29	**484.34**	0.27	**550.61**	0.27	**793.91**	0.27
PCANet(ELM)	1013.10	**0.19**	1446.64	**0.17**	1924.84	**0.18**	2067.06	**0.14**	2307.03	**0.14**	2692.17	**0.14**
2DPCANet(ELM)	**259.52**	**0.19**	**372.02**	**0.17**	**402.39**	**0.16**	**500.94**	**0.14**	**543.87**	**0.13**	**715.24**	**0.15**

Table 5 Recognition rate on AR database

	Sunglasses	Scarf	Sunglasses and scarf
PCANet	0.8897±0.0034	0.8628±0.0036	0.9186±0.0029
2DPCANet	0.9285±0.0025	0.9074±0.0028	0.9492±0.0021
PCANet(ELM)	0.9061±0.0047	0.8661±0.0054	0.9174±0.0038
2DPCANet(ELM)	0.9290±0.0044	0.9089±0.0033	0.9504±0.0030

found through experiments. And the target applications are different: face recognition vs. Dayside aurora.

4 Experimental results

In this section, we evaluate the performance of the proposed 2DPCANet in six different facial databases: YALE, XM2VTS, AR, LFW-a, FERET and Extended Yale B. We use the 2DPCANet to extract the features. SVM and ELM are compared in terms of classification performance.

4.1 DataSets and experimental setups

YALE [4] contains 15 individuals and 11 images for each individual which showing varying facial expressions and configurations. The images are cropped with dimension32 × 32. We randomly select p(=2,3,4,5,6,7) images per individual for training, and use the rest images for testing. We adopt the SVM [3, 6, 19] implementation from the libsvm with the default parameters as the classifier. For the ELM, the number of the hidden nodes is 1000, the parameters of 2DPCANet are set as $D_1 = D_2 = 5$, $k = 8$, $B = 8$. We repeat all trails ten times, and then calculate the average recognition results and the root mean square error(RMSE).

XM2VTS [15] contains 295 individuals and 8 images for each individual with size of 55 × 51. We randomly select p(=2,3,4,5,6,7) images per individual for training, and use the rest images for testing. Like the YALE, we adopt the SVM implementation from the libsvm with the default parameters as the classifier. For the ELM, the number of the hidden nodes is 5000. We repeat all trails ten times, and then calculate the average recognition results the root mean square error(RMSE).

AR [21] contains two-session data of 50 male and 50 female subjects, each person in each session has 13 pictures with 7 images with only illumination and expression change and 3 images wearing sunglasses and 3 images wearing scarf. The images are normalized as 60 × 43.

Table 6 Performance on AR database

	Sunglasses		Scarf		Sunglasses and scarf	
	Training Time(s)	Testing Time(s)	Training Time(s)	Testing Time(s)	Training Time(s)	Testing Time(s)
PCANet	302.13	0.21	296.16	0.21	350.84	0.22
2DPCANet	**193.86**	0.23	**181.34**	0.21	**231.62**	0.23
PCANet(ELM)	384.33	**0.12**	374.91	**0.13**	429.47	**0.14**
2DPCANet(ELM)	**262.52**	**0.13**	**270.18**	**0.13**	**341.82**	**0.14**

Table 7 Recognition rate on LFW-a database

	2	3	4	5	6	7
PCANet	0.2722 ±0.0112	0.3483 ±0.0141	0.4101 ±0.0099	0.4503 ±0.0177	0.4780 ±0.0176	0.5314 ±0.0271
2DPCANet	0.2926 ±0.0093	0.3638 ±0.0076	0.4331 ±0.0071	0.4863 ±0.0136	0.5127 ±0.0126	0.5586 ±0.0176
PCANet(ELM)	0.2763 ±0.0075	0.3507 ±0.0125	0.4071 ±0.0125	0.4602 ±0.0085	0.4965 ±0.0132	0.5312 ±0.0178
2DPCANet(ELM)	0.2979 ±0.0063	0.3738 ±0.0127	0.4363 ±0.0125	0.4848 ±0.0068	0.5218 ±0.0095	0.5675 ±0.0130

We discuss three different circumstances: (1) Images wearing scarf is contained in training images. We randomly select 1 image wearing scarf and 7 images with only illumination and expression change per individual in session1 for training and the remaining in session1 and all images in session2 for testing. (2) Images wearing sunglasses is contained in training images. We randomly select 1 image wearing sunglasses and 7 images with only illumination and expression change per individual in session1 for training and the remaining in session1 and all images in session2 for testing. (3) Images wearing scarf and sunglasses are contained in training images. We randomly select 1 image wearing scarf and 1 image wearing sunglasses and 7 images with only illumination and expression change per individual in session1 for training and the remaining in session1 and all images in session2 for testing. We adopt the SVM implementation from the libsvm with the default parameters as the classifier. For the ELM, the number of the hidden nodes is 5000, and the parameters of 2DPCANet are set as $D_1 = D_2 = 5$, $k = 10$, $B = 8$. We repeat all trails ten times, and then calculate the average recognition results the root mean square error(RMSE).

LFW-a [24] is a version of LFW after alignment using commercial face alignment software. The LFW database contains 5794 individuals in unconstrained environment. We gathered the subjects including no less than ten samples and then form a dataset with 158 subjects from LFW-a. We randomly select p(=2,3,4,5,6,7) images per individual for training, and use the rest images for testing. The parameters of SVM and ELM is the same as the XM2VTS, and the parameters of 2DPCANet are set as $D_1 = D_2 = 5$, $k = 8$, $B = 8$. We repeat all trails ten times, and then calculate the average recognition results the root mean square error(RMSE).

FERET [17, 18] contains 1196 individuals in different lighting conditions with non-neural expressions. The complete dataset is partitioned into disjoints sets: gallery and probe. The probe set is further subdivided into four categories: Fb with different expression changes; Fc with different lighting conditions; Dup-1 was obtained anywhere between 1 min and 1031 days after their respective gallery matches; Dup-2 taken only at least 18 months after their gallery entries. We train our our method on the FERET generic training set, which contains 1002 images and test on the four probe subsets. We use the SVM with the default parameters as the classifier, and the parameters of 2DPCANet are set as $D_1 = D_2 = 10$, $k = 15$, $B = 10$. We also compare it with other relevant methods, such as LDANet [4], RandNet [4],DLANet [7] and CNN. We use Caffe framework [12] and the architecture of CNN is the same as AlexNet proposed by Krizhevsky et al. [14]. The CNN is trained on the gallery images without pre-training for 2000 epochs.

Extended Yale B [8] contains 38 individuals captured under various laboratory controlled lighting conditions. For each subject, we select frontal illumination as the gallery images, and following [8], the images are grouped into four subsets according to the lighting angle with respect to the camera axis. The first two subsets cover the angular range 0° to 25°, the third

Table 8 Performance on LFW-a database

	2		3		4		5		6		7	
	Training Time(s)	Testing Time(s)	Training Time(s)	Testing Time(s)	Training Time(s)	Testing Time(s)	Training Time(s)	Testing Time(s)	Training Time(s)	Testing Time(s)	Training Time(s)	Testing Time(s)
PCANet	42.17	0.17	63.40	0.17	87.57	0.18	124.65	0.19	144.73	0.25	275.83	0.35
2DPCANet	**38.93**	0.19	**59.09**	0.17	**81.09**	0.17	**97.13**	0.17	**134.22**	0.27	**247.37**	0.37
PCANet(ELM)	64.47	**0.06**	88.03	**0.04**	103.12	**0.05**	122.59	**0.05**	191.38	**0.08**	340.57	**0.10**
2DPCANet(ELM)	**55.93**	**0.06**	**70.80**	**0.05**	**95.80**	**0.05**	**119.98**	**0.06**	**181.95**	**0.10**	**194.66**	**0.07**

Table 9 Recognition rate on FERET database

	Fb	Fc	Dup-1	Dup-2	Avg.
LDANet	0.9502	0.9948	0.9312	0.9205	0.9492
RandNet	0.9113	0.9323	0.8374	0.8598	0.8852
DLANet	0.9540	**1**	0.9448	**0.9372**	0.9590
CNN-2	0.8285	0.8368	0.7874	0.8000	0.8132
PCANet	0.9481	0.9812	0.9122	0.9037	0.9363
2DPCANet	**0.9565**	**1**	**0.9452**	0.9368	**0.9596**

subset covers 25° to 50°, and the fourth subsets covers 50° to 77°. We use the SVM with the default parameters as the classifier, and the parameters of 2DPCANet are set as $D_1 = D_2 = 7$, $k = 8$, $B = 8$. We also compare 2DPCANet with LDANet [4], RandNet [4],DLANet [7] and CNN. We use Caffe framework and the architecture of CNN is same as AlexNet proposed by Krizhevsky et al. [14]. The CNN is trained on the gallery images without pre-training for 2000 epochs.

4.2 Results

Tables 1 and 2 have shown the results on YALE, Tables 3 and 4 have shown the results on XM2VTS, Tables 5 and 6 have shown the results on AR, and Tables 7 and 8 have shown the results on LFW-a. From these results, we can see that: (1) the 2DPCANet outperforms the PCANet in terms of recognition accuracy and RMSE or time-consumption, which may because that 2DPCA is based on the image matrix, it extracts the image features directly using the 2D image matrices and maintains the intrinsic 2D structure of an image matrix. (2) The classifier ELM is better than SVM for both PCANet and 2DPCANet in the terms of accurate recognition rate. Unfortunately, the classifier ELM is not as fast as SVM in training time, a possible reason is that the number of the nodes of hidden layer is much large in order to reach good accuracy. And the advantage is smaller on XM2VTS than YALE in recognition rate, which may because the ELM is calculated directly via least square method, which makes its generalization not very good, especially for large or variable database. (3) For the case that images wearing scarf and sunglasses are contained in training images on AR, the accuracy is more than 95%, which indicates the 2DPCANet is robust to occlusions.

The results on FERET are shown in Table 9, and the results on Extended Yale B are given in Table 10, from which we can see that the results on 2DPCANet is superior to other methods, which indicates the advantage of 2DPCANet. In Table 9, 2DPCANet achieves the accuracy 95.65% on Fb and 100% on Fc, which shows that the 2DPCANet is robust to expression and insensitive to illumination. From Table 10, we can see that with the increase of lighting angle,

Table 10 Recognition rate on Extended Yale B database

	Subset1&2	Subset3	Subset4	Avg.
LDANet	0.9941	0.9780	0.9658	0.9793
RandNet	0.9400	0.8793	0.8421	0.8871
CNN-2	0.8603	0.3202	0.1134	0.4322
DLANet	**0.9956**	0.9868	0.9737	0.9854
PCANet	0.9927	0.9868	0.9658	0.9817
2DPCANet	**0.9956**	**0.9890**	**0.9816**	**0.9887**

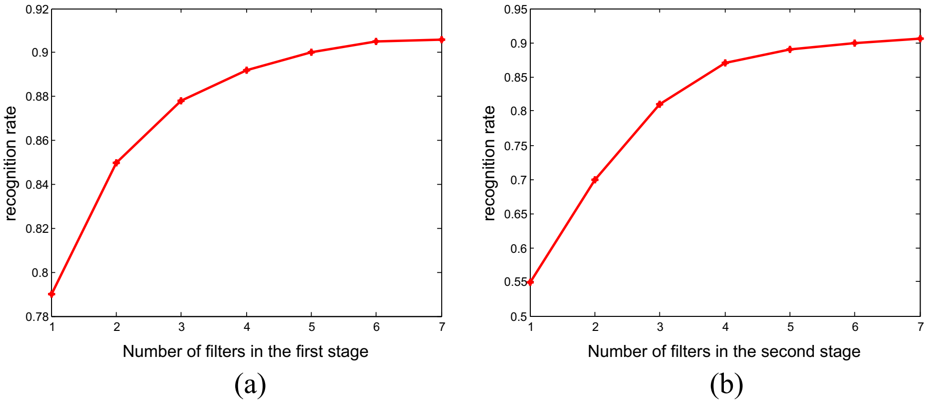


Fig. 2 Impact of the number of filters

the recognition accuracy does not drop significantly, the reason could be that the 2DPCANet is insensitive to lighting variation.

Furthermore, We also investigate the impact of the number of filters and the block size on the facial database XM2VTS for selecting the first two images per individual for training, and the rest images for testing. We evaluate the performance of the proposed 2DPCANet with SVM as classifier.

A. Impact of the number of filters. The filter size of the networks is set as $k = 14$ and their non-overlapping blocks is of size 8 × 8. We set $D_2 = 6$ and then vary D_1 from 1 to 7. Then we fix $D_1 = 6$ and vary D_2 from 1 to 7, the result is shown in Fig. 2. One can see that the recognition rate increases rapidly with the increase of D_1 and D_2 when they are small, and we achieve the best result when $D_1\,, D_2 \geq 6$. It is because the eigenvector of 2DPCA corresponds to the largest eigenvalue, and with the increase of the number of eigenvector, the eigenvalue is decrease, which means its contribution decreases accordingly.

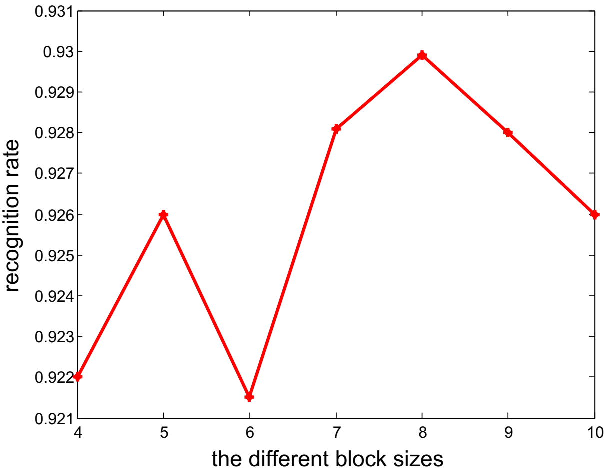


Fig. 3 Impact of the block size

B. Impact of the block size. We next examine the impact of the block size (for histogram computation). The filter size of the networks is set as $k = 8$ and $D_1 = D_2 = 6$. We consider the block sizes from 4×4 to 10×10. The result is shown in Fig. 3. The results suggest that the best block size is 8×8. When the block is too small, it may not contain the local information perfectly, and it may get more global information rather than local information when the block size is too big. The recognition accuracy drops significantly when using the block size 6, which is strange, it may because the output maps can't be partitioned exactly, and we padding it with zero, which introduces more noise to the features.

5 Conclusion

2DPCANet is studied which is a simple deep learning network for face recognition. We use the 2DPCA as the filter banks rather than the PCA in PCANet because of the attractive advantages of 2DPCA. It is easy to evaluate the covariance matrix accurately and maintain the intrinsic 2D structure of an image matrix. In order to verify the effectiveness of 2DPCANet, we evaluate its performance in the facial databases YALE, XM2VTS, AR, LFW-a, FERET and Extended Yale B respectively. We compare the proposed 2DPCANet with the PCANet and compare the two classifiers ELM and SVM with the obtained deep features of 2DPCANet and PCANet. Experimental results show that the recognition accuracy of 2DPCANet is superior to PCANet. And 2DPCANet is much faster than PCANet in training time. Furthermore, 2DPCANet is insensitive to lighting variation and robust to occlusion. Therefore, 2DPCANet is an efficient and robust network for face recognition.

However, the performance comparison of SVM and ELM suggests that although ELM provides better accurate recognition rate, this advantage will deteriorate when the scale of the problem becomes large. Furthermore, the training time of ELM is apparent larger than that of SVM especially for large database.

It should be noted that 2DPCANet was proposed in [13] to solve the problem of aurora image classification, which was discovered by reviewing experts. We gratefully acknowledge the contributions of original authors of [13] and reviewing experts.

Acknowledgements The paper is supported by the National Natural Science Foundation of China (Grant No.61373055, 61672265), Industry Project of Provincial Department of Education of Jiangsu Province (Grant No. JH10-28), and Natural Science Foundation of Jiangsu Province, China (Grant No. BK20151358).

References

1. Arel I et al (2010) Deep machine learning-a new frontier in artificial intelligence research [research frontier]. Comput Intell Mag, IEEE 5(4):13–18
2. Belhumeur PN, Hespanha JP, Kriegman DJ (1997) Eigenfaces vs. fisherfaces: recognition using class specific linear projection. IEEE Trans Pattern Anal Mach Intell 19(7):711–720

3. Burges CJC (1998) A tutorial on support vector machines for pattern recognition. Data Min Knowl Disc 2(2):121–167
4. Chan TH, Jia K, Gao S et al (2015) PCANet: a simple deep learning baseline for image classification? IEEE Trans Image Process 24(12):5017–5032
5. Ciresan D, et al. (2012). Multi-column deep neural networks for image classification. Computer Vision and Pattern Recognition (CVPR), 2012 I.E. Conference on, IEEE
6. Andrew AM (2000) An introduction to support vector machines and other kernel-based learning methods by Nello Christianini and John Shawe-Taylor, Cambridge University Press, Cambridge, 2000, xiii+ 189 pp., ISBN 0-521-78019-5
7. Feng Z et al (2015) DLANet: a manifold-learning-based discriminative feature learning network for scene classification. Neurocomputing 157:11–21
8. Georghiades AS et al (2001) From few to many: illumination cone models for face recognition under variable lighting and pose. IEEE Trans Pattern Anal Mach Intell 23(6):643–660
9. Huang, G.-B., et al. (2004). Extreme learning machine: a new learning scheme of feedforward neural networks. Neural Networks, 2004. Proceedings. 2004 I.E. International Joint Conference on, IEEE
10. Huang GB, Chen L, Siew CK (2006) Universal approximation using incremental constructive feedforward networks with random hidden nodes. IEEE Trans Neural Netw 17(4):879–892
11. Huang G-B et al (2006) Extreme learning machine: theory and applications. Neurocomputing 70(1):489–501
12. Jia Y (2013). Caffe: An open source convolutional architecture for fast feature embedding. Availab le: http://goo.gl/Fo9YO8
13. Jia Z, Han B, Gao X. (2015). 2DPCANet: Dayside Aurora Classification Based on Deep Learning. CCF Chinese Conference on Computer Vision. Springer Berlin Heidelberg: 323–334.
14. Krizhevsky A, et al. (2012). Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems: 1097–1105.
15. Messer K, et al. (1999). XM2VTSDB: The extended M2VTS database. Second international conference on audio and video-based biometric person authentication, Citeseer
16. Ng CJ, Teoh ABJ (2015) DCTNet: A simple learning-free approach for face recognition. 2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA). IEEE, 2015: 761–768.
17. Phillips PJ, et al. (1996). FERET (face recognition technology) recognition algorithm development and test results. Adelphi: Army Research Laboratory
18. Phillips PJ et al (2000) The FERET evaluation methodology for face-recognition algorithms. IEEE Trans Pattern Anal Mach Intell 22(10):1090–1104
19. Smola AJ et al (2004) A tutorial on support vector regression. Stat Comput 14(3):199–222
20. Widrow B et al (2013) The no-prop algorithm: a new learning algorithm for multilayer neural networks. Neural Netw 37:182–188
21. Wright J et al (2009) Robust face recognition via sparse representation. IEEE Trans Pattern Anal Mach Intell 31(2):210–227
22. Yang J, Zhang D, Frangi AF et al (2004) Two-dimensional PCA: a new approach to appearance-based face representation and recognition. IEEE Trans Pattern Anal Mach Intell 26(1):131–137
23. Zhang D, Zhou Z-H (2005) (2D) 2PCA: two-directional two-dimensional PCA for efficient face representation and recognition. Neurocomputing 69(1):224–231
24. Zhu, P., et al. (2012). Multi-scale patch based collaborative representation for face recognition with margin distribution optimization. Computer Vision–ECCV 2012, Springer: 822–835

Dan Yu received her B.S. degree in information security from school of Internet of Things Engineering, Jiangnan University, in 2014. She is currently a postgraduate in Jiangnan University. Her research interests include pattern recognition and deep learning.

Xiaojun Wu received his B.S. degree in mathematics from Nanjing Normal University, Nanjing, PR China in 1991 and M.S. degree in 1996, and Ph.D. degree in Pattern Recognition and Intelligent System in 2002, both from Nanjing University of Science and Technology, Nanjing, PR China, respectively. He was a fellow of United Nations University, International Institute for Software Technology (UNU/IIST) from 1999 to 2000. From 1996 to 2006, he taught in the School of Electronics and Information, Jiangsu University of Science and Technology where he was an exceptionally promoted professor. He joined the School of Information Engineering, Jiangnan University in 2006 where he is a professor. He won the most outstanding postgraduate award by Nanjing University of Science and Technology. He has published more than 200 papers in his fields of research. He was a visiting researcher in the Centre for Vision, Speech, and Signal Processing (CVSSP), University of Surrey, UK from 2003 to 2004. His current research interests are pattern recognition, computer vision, fuzzy systems, neural networks and intelligent systems.