Journal of Electronic Imaging

JElectronicImaging.org

# L1-2D$^2$PCANet: a deep learning network for face recognition

Yun-Kun Li
Xiao-Jun Wu
Josef Kittler

SPIE. IS&T imaging.org

Yun-Kun Li, Xiao-Jun Wu, Josef Kittler, "L1-2D$^2$PCANet: a deep learning network for face recognition," *J. Electron. Imaging* **28**(2), 023016 (2019), doi: 10.1117/1.JEI.28.2.023016.

# L1-2D$^2$PCANet: a deep learning network for face recognition

**Yun-Kun Li,[a] Xiao-Jun Wu,[a,*] and Josef Kittler[b]**
[a]Jiangnan University, Jiangsu Provincial Engineering Laboratory of Pattern Recognition and Computational Intelligence, Wuxi, China
[b]University of Surrey, Center for Vision, Speech and Signal Processing (CVSSP), Guildford, United Kingdom

**Abstract.** We propose a deep learning network L1-2D$^2$PCANet for face recognition, which is based on L1-norm-based two-dimensional principal component analysis (L1-2DPCA). In our network, the role of L1-2DPCA is to learn the filters of multiple convolution layers. After the convolution layers, we deploy binary hashing and blockwise histogram for pooling. We test our network on some benchmark facial datasets, including Yale, AR face database, extended Yale B, labeled faces in the wild-aligned, and Face Recognition Technology database with the convolution neural network, PCANet, 2DPCANet, and L1-PCANet as comparison. The results show that the recognition performance of L1-2D$^2$PCANet in all tests is better than baseline networks, especially when there are outliers in the test data. Owing to the L1-norm, L1-2D$^2$PCANet is robust to outliers and changes of the training images.  [DOI: 10.1117/1.JEI.28.2.023016]

Keywords: face recognition; deep learning; L1-norm-based two-dimensional principal component analysis; outlier.

Paper 181055 received Dec. 21, 2018; accepted for publication Feb. 26, 2019; published online Mar. 20, 2019.

## 1 Introduction

In pattern recognition and computer vision, face recognition is a very important research field.[1–6] Due to the complexity of facial features and the difficulty of manual feature selection,[1,5,6] it is commonly agreed that the best features can be obtained by using unsupervised feature extraction methods.[3–5]

Recently, with Google Alpha Go Zero defeating many Go masters, deep learning has received intensive attentions.[7,8] As a classical deep learning model, convolution neural networks (CNNs) with convolution and pooling layers have achieved astonishing results in many image recognition tasks, reaching an unprecedented accuracy.[9,10] However, CNN still has many shortcomings. During the process of training a CNN model, researchers need to obtain a huge number of parameters, which leads to high computational cost.[11]

To solve this problem, researchers are committed to finding a simple CNN model that requires a small number of parameters. Chan et al.[12] proposed PCANet, which is a simple deep learning network based on unsupervised learning. PCANet uses PCA to learn the filters and deploys simple binary hashing and block histogram for indexing and pooling. Unlike other CNNs that learn filters by backpropagation, PCANet learns filters using the PCA method. Thus, PCANet requires less computational cost, less time, and storage space. The experimental results show the astonishing performance of PCANet.

The PCA method used by PCANet is based on one-dimensional (1-D) vectors. Before deploying PCA, we need to convert two-dimensional (2-D) image matrices into 1-D vectors, which will cause two major problems: (1) Some spatial information of image is implied in the 2-D structure of the image.[13,14] Obviously, the intrinsic information is discarded when the image matrix is converted into 1-D vector.[13,15] (2) The long 1-D vector leads to the requirement of large computational time and storage space in computing the eigenvectors. To solve these problems, Yu et al.[16] proposed 2-D principal component analysis network (2DPCANet), which replaces PCA with 2DPCA.[15,17–19] And Tian et al.[20] proposed multiple scales principal component analysis network (MS-PCANet).

However, both PCA and 2DPCA are based on L2-norm method. It is well known that the methods based on L2-norm are sensitive to outliers so that data with outliers can totally ruin the results from the desired methods.[5,21,22] To solve this problem, Kwak[23] proposed a PCA method based on L1-norm. L1-norm is widely considered to be more robust to outliers.[21,24] L1-PCA adopts the L1-norm for measuring the reconstruction error. On this basis, Xuelong et al.[14] proposed L1-norm-based 2DPCA.

In this paper, L1-norm was introduced into PCANet to get L1-PCANet. Then, we generalize L1-PCANet to L1-2D$^2$PCANet, which shares the same structure with 2DPCANet to generate the feature of input data but L1-2D$^2$PCANet learns filters by L1-2DPCA. In addition, we use support vector machine (SVM) as classifiers for the features generated by the networks. To test the performance of L1-2D$^2$PCANet, we compare it with other three networks (PCANet, 2DPCANet, and L1-PCANet) on Yale, AR,[25] extended Yale B,[26] labeled faces in the wild-aligned (LFW-a),[27] and Face Recognition Technology database (FERET)[28] face databases.

The rest of paper is organized as follows. Sections 2.1 and 2.2 review related work on L1-PCA and L1-2DPCA. L1-PCANet and L1-2D$^2$PCANet are given in Sec. 2.3. Section 3 reports the detail of experiments. Section 4 reports the results and the analysis of the experiments and Sec. 5 concludes this paper.

## 2 Materials and Methods

### 2.1 *L1-Norm-Based PCA*

The proposed L1-PCANet is based on L1-PCA.[21,23] L1-PCA is considered as the simplest and most efficient among many models of L1-norm-based PCA. Let $X = [x_1, x_2, \ldots, x_N] \in \mathbb{R}^{D\times N}$, with $x_i = \mathrm{mat}_D(I_i) \in \mathbb{R}^{D\times 1} (i = 1,2,\ldots,N)$.

*Address all correspondence to Xiao-Jun Wu, E-mail: wu_xiaojun@jiangnan.edu.cn

1017-9909/2019/$25.00 

**Algorithm 1** L1-PCA method.

Input:

- training set: $X = [x_1, x_2, \ldots, x_N] \in \mathbb{R}^{D\times N}$

Output:

- filters $w^*$

1: set $w(0) = 0$ and t = 0

2: For all i $\in \{1,2,\ldots,N\}$, calculate $p_i(t)$ by using Eq. (2)

3: Let t = t + 1 and $\mathrm{w(t)} = \sum_{i=1}^{N} p_i(t-1)x_i$. Then let $\mathrm{w(t)} = w(t)/\|\mathrm{w(t)}\|_2$

4: If $w(t) \neq w(t-1)$, go back to Step 2. Otherwise, set $w^* = w(t)$ and stop.

The $\mathrm{mat}_D(I)$ is a function that maps a matrix $I \in \mathbb{R}^{m\times n}$ to a vector $v \in \mathbb{R}^{D\times 1}$ and $D = m \times n$. Suppose $\mathbf{w} \in \mathbb{R}^{D\times 1}$ be the principal vector to be obtained. Here, we set the number of principal vectors to one to simplify the procedure. The objective of L1-PCA is to maximize the L1-norm variance in the feature space and the successive greedy solutions are expected to provide a good approximation as the following:

$$f(w) = \|w^T X\|_1 = \sum_{i=1}^{N} |w^T x_i|, \quad \text{subject to } \|w\|_2 = 1, \quad (1)$$

where $\|\cdot\|$ denotes L2-norm and $|\cdot|$ denotes L1-norm.

To solve the computational problems posed by the symbol of absolute value, we introduce a polarity parameter $p_i$ in Eq. (1):

$$p_i = \begin{cases} 1, & \text{when } w^T x_i \geq 0 \\ -1, & \text{when } w^T x_i < 0 \end{cases}. \quad (2)$$

By introducing $p_i$, Eq. (1) can be rewritten as follows:

$$f(w) = \sum_{i=1}^{N} p_i w^T x_i. \quad (3)$$

The process of maximization is achieved by Algorithm 1. Here, $t$ denotes the number of iterations and $w(t)$ and $p_i(t)$ denote $w$ and $p_i$ during iteration $t$.

By the above algorithm, we can obtain the first principal vector $\mathbf{w}_1^*$. To compute $\mathbf{w}_k^*(k > 1)$, we have to update the training data as follows:

$$x_i^k = x_i^{k-1} - x_i^{k-1}(\mathbf{w}_{k-1}^* \mathbf{w}_{k-1}^{*T}). \quad (4)$$

## 2.2 *L1-Norm-Based 2DPCA*

In this section, we extend L1-PCA to L1-2DPCA.[14] As mentioned above, 2DPCA computes eigenvectors with 2-D input. Suppose $I_i(\mathrm{i} = 1,2,\ldots,N)$ denote $N$ input training images and $D = m \times n$ being the image size. Let $w \in \mathbb{R}^{w\times 1}$ be the first principal component to be learned. Let $X = [x_1, x_2, \ldots, x_N] \in \mathbb{R}^{D\times N}$, with $x_i = [x_{i1}, x_{i2}, \ldots, x_{ih}]^T \in \mathbb{R}^{h\times w}(i = 1, 2, \ldots, N)$. Note, $x_{ij} \in \mathbb{R}^{1\times w}$ The objective of

**Algorithm 2** L1-2DPCA method.

Input:

- training set: $X = [x_1, x_2, \ldots, x_N] \in \mathbb{R}^{D\times N}$

Output:

- filters $\mathrm{w}^*$

1: Set $w(0) = 0$ and t = 0

2: For all $i \in \{1,2,\ldots,N\}$ and $j \in \{1,2,\ldots,h\}$, calculate $p_{ij}(t)$ by using Eq. (6).

3: Let t = t + 1 and $\mathrm{w(t)} = \sum_{i=1}^{N}\sum_{j=1}^{h} p_{ij}(t-1)x_{ij}$. Then we initialize $\mathrm{w(t)} = w(t)/\|\mathrm{w(t)}\|_2$

4: If $w(t) \neq w(t-1)$, go back to Step 2. Otherwise, set $w^* = w(t)$ and stop.

L1-PCA is to maximize the L1-norm variance in feature space as follows:

$$f(w) = \|Xw\|_1 = \sum_{i=1}^{N}\sum_{j=1}^{h} |x_{ij}w|, \quad \text{subject to } \|w\|_2 = 1. \quad (5)$$

The polarity parameter $p_{ij}$ can be computed as follows:

$$p_{ij} = \begin{cases} 1, & \text{when } x_{ij}w \geq 0 \\ -1, & \text{when } x_{ij}w < 0 \end{cases}. \quad (6)$$

The process of maximization is achieved by Algorithm 2. To compute $\mathrm{w}_k^*(k > 1)$, we have to update the training data as follows:

$$x_{ij}^k = x_{ij}^{k-1} - x_{ij}^{k-1}(\mathbf{w}_{k-1}^* \mathbf{w}_{k-1}^{*T}). \quad (7)$$

At this point, we can find that the difference between L1-PCA and L1-2DPCA is that L1-PCA converts an image matrix into a vector, however, L1-2DPCA directly uses each row in the original image matrix as a vector.

## 2.3 *Proposed Method*

### 2.3.1 *L1-PCANet*

In this section, we propose a PCA-based deep learning network, L1-PCANet. To overcome the sensitivity to outliers in PCANet due to the use of L2-norm, we use the L1-PCA rather than the PCA to learn the filters. L1-PCANet and PCANet[12] share the same network architecture, which is shown in Fig. 1.

Suppose there are $N$ training images $I_i(i = 1,2,\ldots,N)$ of size $m \times n$, and we get $D = m \times n$ patches of size $k \times k$ around each pixel in $I_i$. Then, we take all overlapping patches and map them into vectors:

$$[x_{i,1}, x_{i,2}, \ldots, x_{i,mn}] \in \mathbb{R}^{k^2\times mn}. \quad (8)$$

And we remove the patch mean from each patch and obtain as follows:

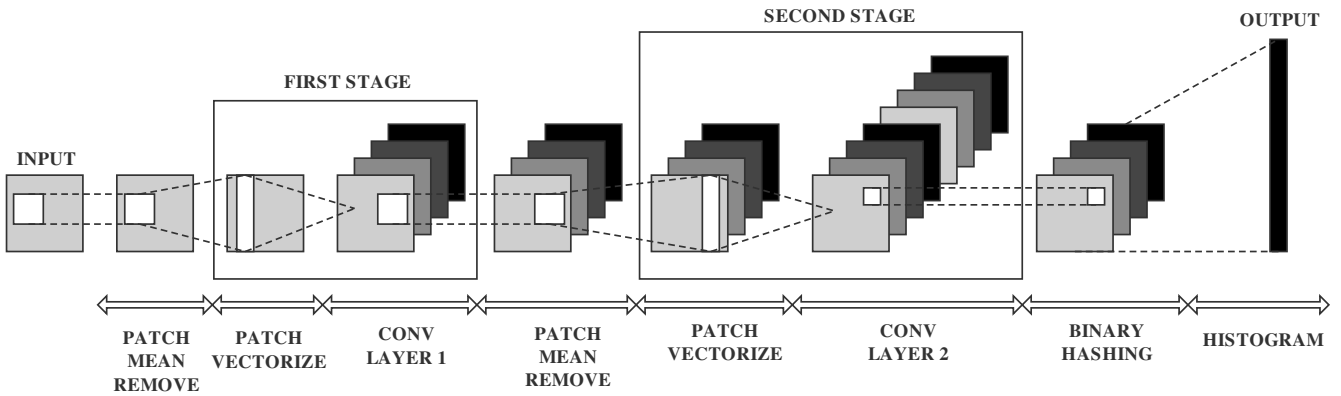



**Fig. 1** The illustration of two-layer L1-PCANet.

$$\bar{X} = [\bar{x}_{i,1}, \bar{x}_{i,2}, \ldots, \bar{x}_{i,mn}] \in \mathbb{R}^{k^2 \times mn}. \tag{9}$$

For all input images, we construct the same matrix and combine them into one matrix to obtain as follows:

$$X = [\bar{X}_1, \bar{X}_2, \ldots, \bar{X}_N] \in \mathbb{R}^{k^2 \times Nmn}. \tag{10}$$

Then, we use L1-PCA mentioned above to learn the filters in stage 1. The filter we want to find is $w \in \mathbb{R}^{k^2 \times 1}$. We take $X$ as the input data of L1-PCA. Assuming that the number of filters in stage 1 is $L_1$, we can obtain the first stage filters $\{\mathbf{w}_1^*, \ldots, \mathbf{w}_{L_1}^*\}$ by repeatedly calling Algorithm 1. The L1-PCA filters of stage 1 are expressed as follows:

$$W_p^1 = \mathrm{mat}_{k,k}(\mathbf{w}_p^*) \in \mathbb{R}^{k \times k}, \tag{11}$$

where $p = 1,2,\ldots,L_1$.

The output of stage 1 can expressed as follows:

$$O_i^p = I_i * W_p^1, i = 1,2,\ldots,N, \tag{12}$$

where $*$ denotes 2-D convolution. We set the boundary of the input image to zero-padding to make sure that $O_i^p$ is of the same size as $I_i$. We can get the filters of the second and subsequent layers by simply repeating the process of the first layer design. The pooling layer of L1-PCANet is almost the same as the pooling layer of L1-2D$^2$PCANet.

### 2.3.2 *L1-2D$^2$PCANet*

In this section, we generalize L1-PCANet to L1-2D$^2$PCANet, which shares the same network with 2DPCANet,[16] as shown in Fig. 2.

*First stage of L1-2D$^2$PCANet.* Let all the assumptions be the same as in Section III. We get all the overlapping patches:

$$x_{i,j} \in \mathbb{R}^{k \times k}, \quad j = 1,2,\ldots,mn, \tag{13}$$

and subtract the patch mean from each of them and we form a matrix:

$$\bar{X}_{x,i} = [\bar{x}_{i,1}, \bar{x}_{i,2}, \ldots, \bar{x}_{i,mn}] \in \mathbb{R}^{k \times kmn}. \tag{14}$$

And we use the transpose of $x_{i,j}$ to form matrix:

$$\bar{X}_{y,i} = [\bar{x}_{i,1}^T, \bar{x}_{i,2}^T, \ldots, \bar{x}_{i,mn}^T] \in \mathbb{R}^{k \times kmn}. \tag{15}$$

For all input images, we construct the matrix by the same way and put them into one matrix, we can obtain as follows:

$$X_x = [\bar{X}_{x,1}, \bar{X}_{x,2}, \ldots, \bar{X}_{x,N}] \in \mathbb{R}^{k \times Nkmn}, \tag{16}$$

$$X_y = [\bar{X}_{y,1}, \bar{X}_{y,2}, \ldots, \bar{X}_{y,N}] \in \mathbb{R}^{k \times Nkmn}. \tag{17}$$

Then, we use L1-2DPCA mentioned above to learn the filters in stage 1. We want to obtain filters $\mathbf{w}_{x,p}^* \in \mathbb{R}^{k \times 1}$

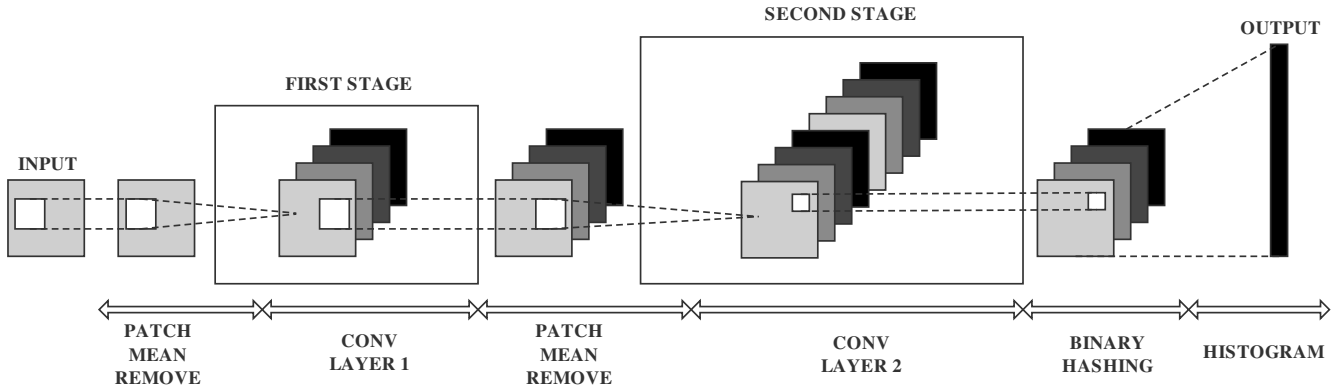



**Fig. 2** The illustration of two-layer L1-2D$^2$PCANet.

and $\mathbf{w}^*_{y,p} \in \mathbb{R}^{k\times 1}$, where $p = 1,2,\ldots,L_1$. $X_x$ and $X_y$ are the input data for L1-2DPCA. Assuming that the number of filters in stage 1 is $L_1$, the first stage filters $\{\mathbf{w}^*_{x,1},\ldots,\mathbf{w}^*_{x,L_1}\}$ and $\{\mathbf{w}^*_{y,1},\ldots,\mathbf{w}^*_{y,L_1}\}$ are obtained by repeatedly calling Algorithm 2.

The filters we need in stage 1 can finally be expressed as follows:

$$W^1_p = \mathbf{w}^*_{x,p} \times \mathbf{w}^{*T}_{y,p} \in \mathbb{R}^{k\times k}. \tag{18}$$

The output of stage 1 will be

$$O^p_i = I_i * W^1_p, i = 1,2,\ldots,N. \tag{19}$$

*Second stage of L1-2D$^2$PCANet.* Like in the first stage, we can start with the overlapping patches of $O^p_i$ and remove the patch mean from each patch. Then, we have

$$Y^p_{x,i} = [\bar{y}_{i,p,1},\ldots,\bar{y}_{i,p,mn}] \in \mathbb{R}^{k\times kmn}, \tag{20}$$

$$Y^p_{y,i} = [\bar{y}^T_{i,p,1},\ldots,\bar{y}^T_{i,p,mn}] \in \mathbb{R}^{k\times kmn}. \tag{21}$$

Further, we define the matrix that collects all the patches without the patch mean of the $k$'th output $O^k_i$ being removed as

$$Y^p_x = [Y^m_{\mathrm{x},1}, Y^m_{\mathrm{x},2},\ldots,Y^m_{x,N}] \in \mathbb{R}^{k\times Nkmn}, \tag{22}$$

$$Y^p_y = [Y^p_{\mathrm{y},1}, Y^p_{\mathrm{y},2},\ldots,Y^p_{y,N}] \in \mathbb{R}^{k\times Nkmn}. \tag{23}$$

Finally, the input of the second stage is obtained by concatenating $Y^p_x$ and $Y^p_y$ for all $L_1$ filters:

$$Y_x = [Y^1_x, Y^2_x,\ldots,Y^{L_1}_x] \in \mathbb{R}^{k\times L_1 Nkmn}, \tag{24}$$

$$Y_y = [Y^1_y, Y^2_y,\ldots,Y^{L_1}_y] \in \mathbb{R}^{k\times L_1 Nkmn}. \tag{25}$$

We take $Y_x$ and $Y_y$ as the input data of L1-2DPCA. Assuming that the number of filters in stage 2 is $L_2$, we design the second stage filters $\{\mathbf{w}^*_{\mathrm{x},1},\ldots,\mathbf{w}^*_{x,L_2}\}$ and $\{\mathbf{w}^*_{y,1},\ldots,\mathbf{w}^*_{y,L_2}\}$ by repeatedly calling Algorithm 2. The L1-2DPCA filters of stage 2 are expressed as follows:

$$W^2_q = \mathbf{w}^*_{x,q} \times \mathbf{w}^{*T}_{y,q} \in \mathbb{R}^{k\times k}, \tag{26}$$

where $q = 1,2,\ldots,L_2$.

Therefore, we have $L_2$ outputs for each output $O^p_i$ of stage 1:

$$B^q_i = \{O^p_i * W^2_q\}, \quad l = 1,2,\ldots,L_2. \tag{27}$$

Note that the number of outputs of stage 2 is $L_1L_2$.

*Pooling stage.* First, we use a Heaviside-like step function to binarize the output of stage 2. The function $H(\cdot)$ can be expressed as follows:

$$H(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases}. \tag{28}$$

Each pixel is encoded by the following function:

$$T^m_i = \sum_l^{L_2} 2^{l-1} H(B^q_i), \tag{29}$$

where $T^m_i$ is an integer of range $[0, 2^{L_2-1}]$.

Second, we divide $T^m_i$ into $B$ blocks. Then, we make a histogram of all blocks of $T^m_i$ with $2^{L_2}$ values and concatenate all the histogram of $B$ blocks into one vector hist$(T^m_i)$. In this way, we obtain $L_1$ histograms and we put them into a vector:

$$f_i = [\mathrm{hist}(T^1_i),\ldots,\mathrm{hist}(T^{L_2}_i)] \in \mathbb{R}^{2^{L_2}L_1B\times 1}. \tag{30}$$

Using the L1-2DPCA model described above, we can transform an input image into a feature vector as the output of L1-2D$^2$PCANet.

## 3 Experiments

In this section, we evaluate the performance of L1-PCANet and L1-2D$^2$PCANet with PCANet and 2DPCANet as baselines on Yale, AR, extended Yale B, and FERET databases, respectively, which are shown in Fig. 3. SVM[29] implementation from the libsvm is used as the classifier with default settings. We repeat some experiments 10 times and calculate the average recognition accuracy and root mean square error (RMSE). In all experiments, we create all PCANet and its different variations instances on MATLAB and other CNNs on Tensorflow.

### 3.1 *Extended Yale B*

Extended Yale B consists of 2414 images of 38 individuals captured with different lighting conditions. These pictures are preprocessed to have the same size $48 \times 42$ and alignment. The parameters are set as $k = 5$, $B = 3$, $L_1 = L_2 = 4$.

In experiment 1, we compare L1-PCANet and L1-2D$^2$PCANet with PCANet and 2DPCANet. We randomly select $i = 2,3,4,5,6,7$ images per individual for training and use the rest for testing. We also create AlexNet[30] and GoogleNet[11] instances for comparison, which are trained on 1024 images randomly selected from extended Yale B for 20 epochs. The architecture of AlexNet is the same as in Ref. 30 and the architecture of GoogleNet is the same as in Ref. 11. The parameters of two CNNs are set as learning rate = 0.0001, batch size = 128, drop keep prob. = 0.8. The results are shown in Table 1.

In experiment 2, to evaluate the robustness of L1-PCANet and L1-2D$^2$PCANet to outliers, we randomly add blockwise noise to the test images to generate test images with outliers. Within each block, the pixel value is randomly set to be 0 or 255. These blocks occupy 10%, 20%, 30%, and 50% of the images and they are added to the random position of the image, respectively, which can be seen in Fig. 4. The results are shown in Table 2.

To demonstrate the superiority of the proposed method, we compare L1-PCANet and L1-2D$^2$PCANet with the traditional L1-PCA and L1-2DPCA in experiment 3. We create L1-PCA and L1-2DPCA instances based on Refs. 23 and 24. The parameters of L1-PCA and L1-2DPCA are set as $w = 100$. We randomly select $i = 2,3,4,5,6,7$ images per

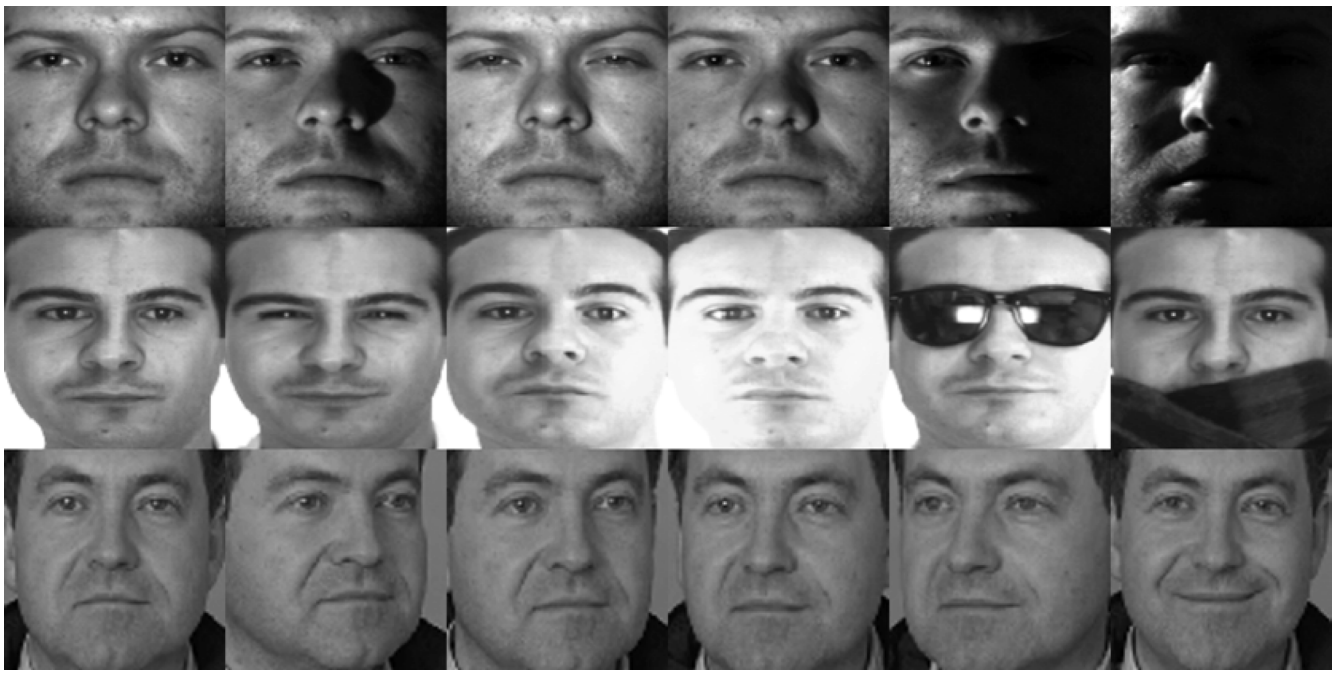
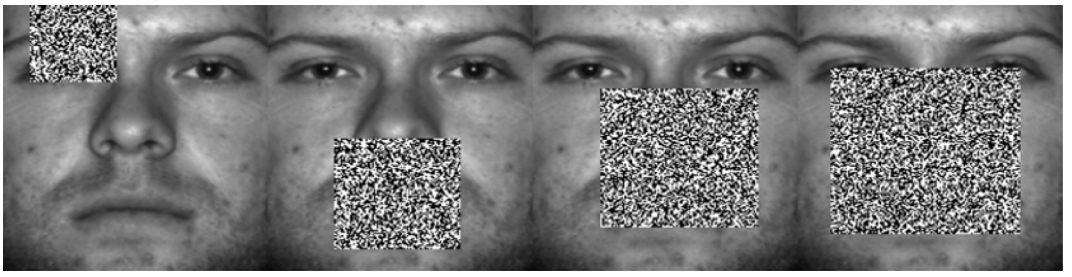


**Fig. 3** Images in three datasets. Top line: Extended Yale B,[26] middle line: AR,[25] bottom line: FERET.[28]

**Table 1** Experiment 1 on extended Yale B.[26]

| | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|
| AlexNet | | | 85.56 ± 0.53 | | | |
| GoogleNet | | | 95.18 ± 0.42 | | | |
| PCANet | 83.41 ± 5.31 | 84.51 ± 5.70 | 84.42 ± 5.37 | 82.48 ± 7.18 | 84.06 ± 6.22 | 89.56 ± 5.48 |
| 2DPCANet | 97.48 ± 1.03 | 97.34 ± 1.81 | 97.01 ± 1.64 | 96.71 ± 2.48 | 95.16 ± 2.93 | 97.22 ± 2.02 |
| L1-PCANet | 97.88 ± 0.22 | 97.98 ± 0.22 | 97.88 ± 0.18 | 97.86 ± 0.17 | 97.94 ± 0.19 | 97.90 ± 0.16 |
| L1-2D$^2$PCANet | 99.67 ± 0.09 | 99.71 ± 0.07 | 99.73 ± 0.09 | 99.73 ± 0.06 | 99.75 ± 0.06 | 99.77 ± 0.07 |



(a) (b) (c) (d)

**Fig. 4** Some generalized face images with outliers of extended Yale B:[26] (a) 10%; (b) 20%; (c) 30%; and (d) 50%.

individual for gallery images and seven images per individual for training. The results are shown in Table 3.

In experiment 4, we examine the impact of the block size B for L1-PCANet and L1-2D$^2$PCANet. The block size changes from $2 \times 2$ to $8 \times 8$. The results are shown in Fig. 5(a).

### 3.2 *AR*

AR face database contains 2600 color images corresponding to 100 people's faces (50 men and 50 women). It has two session data from two different days and each person in each session has 13 images, including 7 images with only illumination and expression change, 3 images wearing sunglasses,

**Table 2** Experiment 2 on extended Yale B.[26]

| | 10% | 20% | 30% | 50% |
|---|---|---|---|---|
| PCANet | $92.68 \pm 0.42$ | $88.51 \pm 0.40$ | $74.63 \pm 0.48$ | $44.10 \pm 0.76$ |
| 2DPCANet | $94.26 \pm 0.25$ | $88.71 \pm 0.57$ | $79.54 \pm 0.89$ | $55.34 \pm 0.70$ |
| L1-PCANet | $94.34 \pm 0.40$ | $91.50 \pm 0.51$ | $83.58 \pm 0.60$ | $65.01 \pm 0.61$ |
| L1-2D$^2$PCANet | $99.00 \pm 0.15$ | $98.28 \pm 0.18$ | $95.73 \pm 0.20$ | $84.01 \pm 0.74$ |

**Table 3** Experiment 3 on extended Yale B.[26]

| | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|
| L1-PCA | $22.10 \pm 1.69$ | $32.68 \pm 1.66$ | $43.23 \pm 2.00$ | $52.78 \pm 1.70$ | $59.23 \pm 2.11$ | $64.49 \pm 1.42$ |
| L1-2DPCA | $35.72 \pm 2.50$ | $43.26 \pm 1.92$ | $51.72 \pm 2.12$ | $60.75 \pm 1.42$ | $65.44 \pm 1.88$ | $70.60 \pm 1.56$ |
| L1-PCANet | $60.83 \pm 3.81$ | $74.72 \pm 2.07$ | $83.13 \pm 1.84$ | $87.90 \pm 1.23$ | $91.75 \pm 1.62$ | $94.37 \pm 1.04$ |
| L1-2D$^2$PCANet | $76.23 \pm 3.48$ | $85.20 \pm 2.04$ | $90.65 \pm 1.65$ | $93.52 \pm 1.10$ | $95.62 \pm 1.14$ | $96.86 \pm 0.77$ |

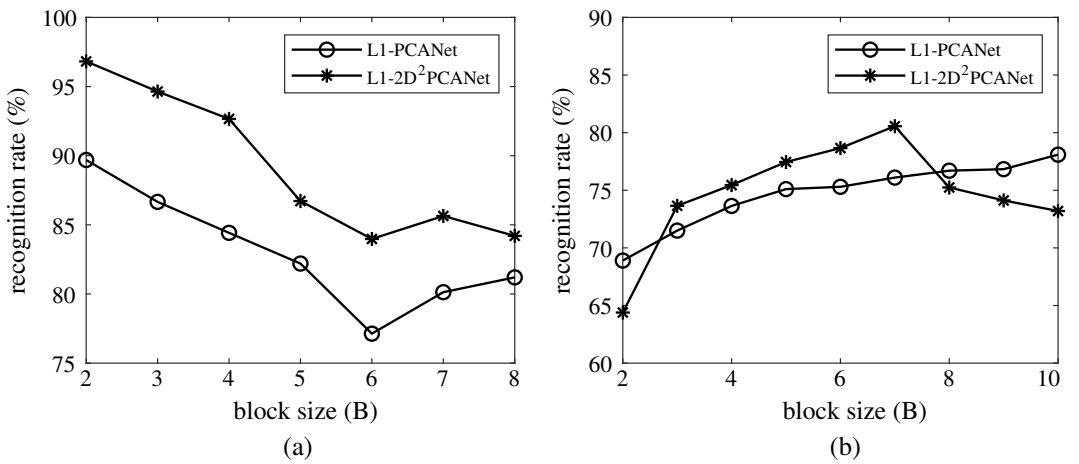



**Fig. 5** Recognition rate of L1-PCANet and L1-2D$^2$PCANet on extended Yale B and FERET dataset for varying number of block size. (a) Extended Yale B and (b) FERET.

and 3 images wearing scarf. Images show frontal faces with different facial expressions, illumination conditions, and occlusions (sunglasses and scarf). These pictures are preprocessed to $40 \times 30$. The parameters are set as $k = 5$, $B = 4$, $L_1 = L_2 = 4$, respectively.

In experiment 5, in order to investigate the impact of the choice of training images, we divide the experiment into four groups: (1) In group 1, we randomly select five images with only illumination and expression change from session 1 per individual as training images; (2) in group 2, we randomly select four images with only illumination and expression change and one image wearing sunglasses from session 1 per individual as training images; (3) in group 3, we randomly select four images with only illumination and expression change and one image wearing scarf from session 1 per individual as training images. The remaining images are test samples; and (4) in group 4, we randomly select three images with only illumination and expression change, one image wearing sunglasses and one image wearing scarf from session 1 per individual as training images. The remaining images in session 1 and all images in session 2 are used as test images. We manually select five images from session 1 as the gallery images and keep gallery images of each group the same. The results are shown in Table 4.

In order to investigate the impact of the choice of gallery images, experiment 6 is the same as experiment 5 except that the gallery images and the training images are exchanged. We use the remaining images in session 1 and all images in session 2 as test samples. The results are shown in Table 5.

### 3.3 *FERET*

This database contains a total of 11338 facial images. They were collected by photographing 994 subjects at various facial angles. We gathered a subset from FERET, which is composed by 1400 images recording of 200 individuals, with each seven images exhibit large variations in facial

**Table 4** Experiment 5 on AR.[25]

| | No occlusion | Sunglass | Scarf | Sunglass and scarf |
|---|---|---|---|---|
| PCANet | $78.63 \pm 3.09$ | $78.74 \pm 4.84$ | $79.23 \pm 4.47$ | $80.40 \pm 4.10$ |
| 2DPCANet | $82.94 \pm 4.31$ | $83.85 \pm 4.48$ | $82.21 \pm 2.97$ | $83.44 \pm 4.27$ |
| L1-PCANet | $87.09 \pm 0.50$ | $86.73 \pm 0.31$ | $87.33 \pm 0.12$ | $86.46 \pm 0.22$ |
| L1-2D$^2$PCANet | $89.26 \pm 0.37$ | $88.59 \pm 0.27$ | $88.85 \pm 0.28$ | $88.52 \pm 0.19$ |

**Table 5** Experiment 6 on AR.[25]

| | No occlusion | Sunglass | Scarf | Sunglass and scarf |
|---|---|---|---|---|
| PCANet | $66.71 \pm 0.87$ | $69.62 \pm 0.69$ | $69.59 \pm 0.69$ | $72.66 \pm 0.70$ |
| 2DPCANet | $69.24 \pm 0.70$ | $74.78 \pm 0.70$ | $72.14 \pm 0.99$ | $75.51 \pm 0.61$ |
| L1-PCANet | $68.56 \pm 0.65$ | $75.23 \pm 0.60$ | $72.35 \pm 0.77$ | $79.34 \pm 0.71$ |
| L1-2D$^2$PCANet | $77.08 \pm 0.64$ | $81.10 \pm 0.37$ | $78.34 \pm 0.61$ | $84.17 \pm 0.75$ |

expression, facial angle, and illumination. This subset is available in our GitHub repository. These pictures are preprocessed to have the same size $40 \times 40$ and alignment. The parameters are set as $k = 5$, $B = 10$, $L_1 = L_2 = 4$, respectively.

In experiment 7, we divide the experiment into seven groups. The training images of each group consist of 200 images from the subset with different facial angle, expression, and illumination. We use the remaining images in the subset as test images. The results are shown in Table 6.

In experiment 8, we examine the impact of the block size B for L1-PCANet and L1-2D$^2$PCANet. The block size changes from $2 \times 2$ to $10 \times 10$. The results are shown in Fig. 5(b).

### 3.4 *Yale*

Yale consists of 15 individuals and 11 images for each individual, which shows varying facial expressions and configurations. These pictures are preprocessed to have the same size $32 \times 32$. The parameters are set as $k = 5$, $B = 4$, $L_1 = L_2 = 4$, respectively.

In experiment 9, we randomly select $i = 2,3,4,5,6,7$ images per individual for training and use the rest for testing. The results are shown in Table 7.

**Table 6** Experiment 7 on FERET.[28]

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | Average | RMSE |
|---|---|---|---|---|---|---|---|---|---|
| PCANet | 75.83 | 76.83 | 76.17 | 68.00 | 73.67 | 69.83 | 79.11 | 74.21 | 3.69 |
| 2DPCANet | 73.17 | 76.17 | 76.17 | 73.67 | 78.33 | 73.50 | 74.00 | 75.00 | 1.78 |
| L1-PCANet | 82.83 | 82.17 | 82.00 | 82.50 | 85.00 | 82.50 | 81.83 | 82.69 | 0.99 |
| L1-2D$^2$PCANet | 86.00 | 84.83 | 85.50 | 86.50 | 87.33 | 86.83 | 86.83 | 86.26 | 0.81 |

**Table 7** Experiment 9 on Yale.[26]

| | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|
| PCANet | $86.33 \pm 1.87$ | $86.75 \pm 2.37$ | $87.50 \pm 1.58$ | $87.25 \pm 2.12$ | $87.25 \pm 2.14$ | $87.29 \pm 2.22$ |
| 2DPCANet | $91.33 \pm 2.80$ | $91.78 \pm 1.94$ | $90.44 \pm 2.59$ | $90.67 \pm 2.34$ | $90.87 \pm 2.90$ | $91.93 \pm 2.13$ |
| L1-PCANet | $91.45 \pm 0.89$ | $92.00 \pm 0.83$ | $91.22 \pm 0.54$ | $91.00 \pm 0.44$ | $91.89 \pm 0.51$ | $92.67 \pm 0.33$ |
| L1-2D$^2$PCANet | $94.03 \pm 0.32$ | $95.10 \pm 0.41$ | $94.95 \pm 0.33$ | $95.25 \pm 0.32$ | $95.16 \pm 0.41$ | $95.66 \pm 0.40$ |

**Table 8** Experiment 10 on LFW-a.[27]

| | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|
| PCANet | $30.07 \pm 4.69$ | $31.86 \pm 5.35$ | $34.35 \pm 5.91$ | $35.71 \pm 6.34$ | $38.56 \pm 6.82$ |
| 2DPCANet | $33.00 \pm 3.52$ | $35.68 \pm 3.64$ | $39.02 \pm 3.74$ | $39.92 \pm 3.98$ | $43.15 \pm 4.12$ |
| L1-PCANet | $34.14 \pm 0.39$ | $36.27 \pm 0.29$ | $39.08 \pm 0.57$ | $40.25 \pm 0.77$ | $44.26 \pm 0.81$ |
| L1-2D$^2$PCANet | $39.35 \pm 0.29$ | $42.20 \pm 0.46$ | $45.91 \pm 0.34$ | $46.99 \pm 0.42$ | $50.12 \pm 0.47$ |

### 3.5 *LFW-a*

LFW-a is a version of LFW after alignment with deep funneling. We gathered the individuals, including more than nine images from LFW-a. The parameters are set as $k = 5$, $B = 3$, $L_1 = L_2 = 4$, respectively.

In experiment 10, we randomly choose $i = 3{,}4{,}5{,}6{,}7$ images per individual for gallery images and keep training images of each group the same. The results are shown in Table 8.

## 4 Results and Analysis

Tables 1 and 3 show the results of experiments 1 and 3 on extended Yale B, Table 4 shows the result of experiment 5 on AR, Table 6 shows the result of experiment 7 on FERET, Table 7 shows the result on Yale, and Table 8 shows the result on LFW-a.

In these experiments, we changed the training images by random selection. From the results, we can see that the L1-2D$^2$PCANet outperforms L1-PCA, L1-2DPCA, PCANet, 2DPCANet, and L1-PCANet in terms of recognition accuracy and RMSE, because we introduce L1-norm into the network. The two L1-norm-based networks we proposed are far superior to the traditional L2-norm-based networks in terms of RMSE, which means the proposed networks are insensitive to changes in training images. That is, the accuracy of the traditional L2-norm-based networks largely depends on the choice of training images while the L1-norm-based networks we proposed can achieve better and stable accuracy under any training images. A possible explanation of this phenomenon is as follows. In fact, the expression, posture, illumination condition, and occlusion in the images can be regarded as interference or noise in face recognition. This noise degrades L2-norm-based networks much more than it degrades L1-norm-based networks. Therefore, the proposed networks exhibit the superiority when the training images contain some changes in expression, posture, illumination condition, and occlusion.

Table 2 shows the result of experiment 2 on extended Yale B. In this experiment, we randomly add blockwise noise to the test images. From the results, we can see that as the blockwise noise increases from 10% of the image size to 50%, the performance of PCANet, 2DPCANet, and L1-PCANet drops rapidly while L1-2D$^2$PCANet still has good performance. Therefore, it can be considered that L1-2D$^2$PCANet has better robustness against outlier and noise than other three networks.

We also investigate the impact of the choice of gallery images on AR; see Table 4. From the horizontal comparison of Table 5, the more categories the gallery contains, the higher the accuracy is.

Figure 5 shows the result of experiment 4 on extended Yale B and experiment 8 on FERET. When the block is small, the local information cannot be contained perfectly, and it may get more noise when the block is big.

## 5 Conclusion

In this paper, we have proposed a deep learning network L1-2D$^2$PCANet, which is a simple but robust method. We use the L1-norm-based 2DPCA[14] instead of L2-norm-based 2DPCA[15] for the filter learning because of the advantages of L1-norm. It is more robust to outliers than L2-norm. By introducing L1-norm into 2DPCANet,[16] we hope the network will inherit such advantages.

To verify the performance of L1-2D$^2$PCANet, we evaluate them on the facial datasets, including AR, extended Yale B, Yale, and FERET, respectively. The results show that L1-2D$^2$PCANet has three distinct advantages over traditional L2-norm-based networks: (1) Statistically, the accuracy of L1-2D$^2$PCANet is higher than that of other networks on all test datasets. (2) L1-2D$^2$PCANet has better robustness to changes in training images compared with the other networks. (3) Compared with the other networks, L1-2D$^2$PCANet has better robustness to noise and outliers. Therefore, L1-2D$^2$PCANet is an efficient and robust network for face recognition.

However, L1-2DPCA brings more computational load to the network, which increases the computational cost of L1-2D$^2$PCANet. Despite this, the computational cost of L1-2D$^2$PCANet is far less than those traditional CNNs, which are based on backpropagation.

In the future work, we will work on the improving of L1-2DPCA algorithm to solve the problem of the computational cost of L1-2D$^2$PCANet.

### *Acknowledgments*

The paper is supported by the National Natural Science Foundation of China (Grant Nos. 61672265 and U1836218), the 111 Project of Ministry of Education of China (Grant No. B12018), UK EPSRC under Grant No. EP/N007743/1, and MURI/EPSRC/dstl under Grant No. EP/R018456/1.

### *References*

1. P. A. Devijver and J. Kittler, *Pattern Recognition: A Statistical Approach*, Prentice Hall International, New Jersey (1982).
2. B. D. Ripley, "Pattern recognition and neural networks," *Technometrics* **39**(2), 233–234 (1999).
3. A. K. Jain, R.P.W. Duin, and J. Mao, "Statistical pattern recognition: a review," *IEEE Trans. Pattern Anal. Mach. Intell.* **22**(1), 4–37 (2000).

4. C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*, Springer-Verlag, New York, p. 049901 (2006).
5. X.-J. Wu et al., "A new direct LDA (D-LDA) algorithm for feature extraction in face recognition," in *Int. Conf. Pattern Recognit.*, IEEE Xplore (2004).
6. Y. Yi et al., "Face recognition using spatially smoothed discriminant structure-preserved projections," *J. Electron. Imaging* **23**(2), 023012 (2014).
7. Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature* **521**(7553), 436–444 (2015).
8. D. Silver et al., "Mastering the game of Go without human knowledge," *Nature* **550**(7676), 354–359 (2017).
9. S. Lawrence et al., "Face recognition: a convolutional neural-network approach," *IEEE Trans. Neural Networks* **8**(1), 98–113 (1997).
10. N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," 1 Eprint Arxiv (2014).
11. C. Szegedy et al., "Going deeper with convolutions," in *IEEE Comput. Vision and Pattern Recognit.* (2015).
12. T. H. Chan et al., "PCANet: a simple deep learning baseline for image classification?" *IEEE Trans. Image Process.* **24**(12), 5017–5032 (2015).
13. X. J. Wu et al., "A new algorithm for generalized optimal discriminant vectors," *J. Comput. Sci. Technol.* **17**(3), 324–330 (2002).
14. L. Xuelong, P. Yanwei, and Y. Yuan, "L1-norm-based 2DPCA," *IEEE Trans. Syst. Man Cybern. Part B Cybern.* **40**(4), 1170–1175 (2010).
15. J. Yang et al., "Two-dimensional PCA: a new approach to appearance-based face representation and recognition," *IEEE Trans. Pattern Anal. Mach. Intell.* **26**(1), 131–137 (2004).
16. D. Yu and X. J. Wu, "2DPCANet: a deep leaning network for face recognition," *Multimedia Tools Appl.* **77**(10), 12919–12934 (2018).
17. M. Hirose et al., "Principal component analysis for surface reflection components and structure in the facial image and synthesis of the facial image in various ages," *Proc. SPIE* **9398**, 939809 (2015).
18. Z. Jia, B. Han, and X. Gao, "2DPCANet: dayside aurora classification based on deep learning," in *CCF Chin. Conf. Comput. Vision.*, Springer, Berlin, Heidelberg, pp. 323–334 (2015).
19. Q. R. Zhang, "Two-dimensional parameter principal component analysis for face recognition," *Adv. Mater. Res.* **971**–**973**, 1838–1842 (2014).
20. L. Tian, C. Fan, and Y. Ming, "Multiple scales combined principle component analysis deep learning network for face recognition," *J. Electron. Imaging* **25**(2), 023025 (2016).
21. C. Ding, "R1-PCA: rotational invariant L1-norm principal component analysis for robust subspace factorization," in *Int. Conf. Mach. Learn.*, ACM (2006).
22. A. Baccini, P. Besse, and A. D. Falguerolles, "A l1-norm PCA and a heuristic approach," in *Ordinal and Symbolic Data Analysis*, E. Diday, Y. Lechevalier, and P. Opitz, Eds., Springer, New York, pp. 359–368 (1996).
23. N. Kwak, "Principal component analysis based on L1-norm maximization," *IEEE Trans. Pattern Anal. Mach. Intell.* **30**(9), 1672–1680 (2008).
24. X. Li, Y. Pang, and Y. Yuan, "L1-norm-based 2DPCA," *IEEE Trans. Syst. Man Cybern. Part B* **40**(4), 1170–1175 (2010).
25. A. M. Martinez, "The AR face database," CVC Technical Report, p. 24 (1998).
26. A. S. Geo et al., "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 (2001).
27. P. Zhu et al., "Multi-scale patch based collaborative representation for face recognition with margin distribution optimization," in *Eur. Conf. Comput. Vision*, Springer, Berlin, Heidelberg, pp. 822–835 (2012).
28. P. J. Phillips et al., "The FERET September 1996 database and evaluation procedure," *Lect. Notes Comput. Sci.* **1206**, 395–402 (1997).
29. C. A. Burges, "Tutorial on support vector machines for pattern recognition," *Data Mining Knowl. Discovery* **2**, 121–167 (1998).
30. A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," *Adv. Neural Inf. Process. Syst.* **25**(2) (2012).



**Yun-Kun Li** received his BS degree in microelectronics from the School of Internet of Things Engineering, Jiangnan University, in 2017. He is currently a postgraduate in the Jiangsu Provincial Engineering Laboratory of Pattern Recognition and Computational Intelligence, Jiangnan University. His research interests include pattern recognition and deep learning.

**Xiao-Jun Wu** received his BS degree in mathematics from Nanjing Normal University, Nanjing, in 1991, and his MS and PhD degrees in pattern recognition and intelligent system from Nanjing University of Science and Technology, Nanjing, in 1996 and 2002, respectively. He has published more than 200 papers in his fields of research. His current research interests include pattern recognition, computer vision, and computational intelligence.

**Josef Kittler** received his BA, PhD, and DSc degrees from the University of Cambridge, in 1971, 1974, and 1991, respectively. He is currently a professor of machine intelligence with the Centre for Vision, Speech, and Signal Processing, University of Surrey, Guildford, United Kingdom. He has authored the textbook "*Pattern Recognition: A Statistical Approach*" and over 600 scientific papers. His current research interests include biometrics, video and image database retrieval, medical image analysis, and cognitive vision.