Neurocomputing 275 (2018) 924–931

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Learning more distinctive representation by enhanced PCA network

Yang Liu, Shuangshuang Zhao, Qianqian Wang, Quanxue Gao*

State Key Laboratory of Integrated Services Network, Xidian University, China

ARTICLE INFO

Article history:
Received 16 January 2017
Revised 23 June 2017
Accepted 13 September 2017
Available online 21 September 2017

Communicated by Dr. Nianyin Zeng

Keywords:
PCA
Deep learning
Face recognition
Convolutional neural network (CNN)

ABSTRACT

Subspace learning approaches extract features by a simple linear transformation, which can viewed as a shallow network, and they cannot reveal the deep structure embedded in pixels of image. To solve this problem, a deep principal component analysis (PCA) network, namely enhanced PCA Network (EPCANet), is proposed to explore more distinctive representation for face images. EPCANet adds a spatial pooling layer between the first layer and second layer in the PCANet. The spatial pooling layer reveals more spatial and distinctive information by down-sampling or pixel offset for the first layer output and original images. Extensive experimental results in several databases illustrate the efficiency of our proposed methods.

1. Introduction

Feature extraction is one of the important and fundamental problems in pattern recognition. Principal Component Analysis (PCA) [1] and Linear Discriminant Analysis (LDA) [2] are two of the most representative unsupervised and supervised techniques for feature extraction. PCA has been widely investigated and become one of the successful approaches in pattern analysis especially image recognition. It is an unsupervised feature extraction method and aims to find a set of projection vectors so that the projected data can provide an efficient representation for data. LDA is a supervised method and seeks to find a set of projection vectors that maximize the between-class distance and minimize the within-class distance simultaneously, however, it cannot achieve good performance when the number of the labeled data is small [1–3].

To well uncover local intrinsic geometric structure, many manifold learning based subspace methods have been developed [4–7], among which locality preserving projection (LPP) [5] and neighborhood preserving projection (NPE) [6] are two of the most representative linear methods. However, manifold learning methods heavily depend on adjacent graph, which is constructed by hand. This reduces the flexibility of methods. To handle this problem, sparse representation is a powerful tool [8] and has been widely used in pattern recognition and machine learning. Apart from the aforementioned methods, some hand-crafted low-level features, such as local binary patterns (LBP), SIFT and HOG features [9,10], have achieved great success for some special data in pattern analysis and machine learning, however, they cannot achieve good performance for some pattern analysis.

All of the aforementioned methods directly extract features from image pixels by a simple linear transform, which can be viewed as shallow network model, and cannot provide enough efficient representation for image analysis. Recently, breakthroughs on deep learning have been achieved on speech and language processing, object detection, image retrieval and image recognition especially face recognition since Hinton developed deep belief network (DBN). Deep learning aims to learn the most efficient and discriminant features by different deep architectures. Restricted Boltzmann machines (RBMs) [11–13], deep belief networks (DBNs) [14–18], autoencoder (AE) [19–21] and convolutional neural networks (CNNs) [22–25] are four of the most representative deep architectures. RBMs was proposed by Smolensky and has been widely used in pattern recognition since Hinton published his work [15] in 2006, which updates the weights by maximizing the likelihood. The DBNs consists of multiple layers of stochastic and latent variables and are more effective to analyze unlabeled data. It has been widely applied into speech recognition, breast cancer classification, and image retrieve [16,17]. An autoencoder (AE) aims to efficiently encode data for the purpose of dimensionality reduction, and has been widely used to learn generative models of data [21]. CNN is a multi-layer neural network that is composed of two different layers convolution layers and sub-sampling (pooling) layers. It has been successfully applied to object detection, face recognition, behavior recognition, speech recognition, and image classification [23–25]. Compared with subspace learning methods, deep learning is time-consuming for parameter tuning.

* Corresponding author.
E-mail addresses: 610887187@qq.com, xd_ste_pr@163.com (Q. Gao).

https://doi.org/10.1016/j.neucom.2017.09.041
0925-2312/

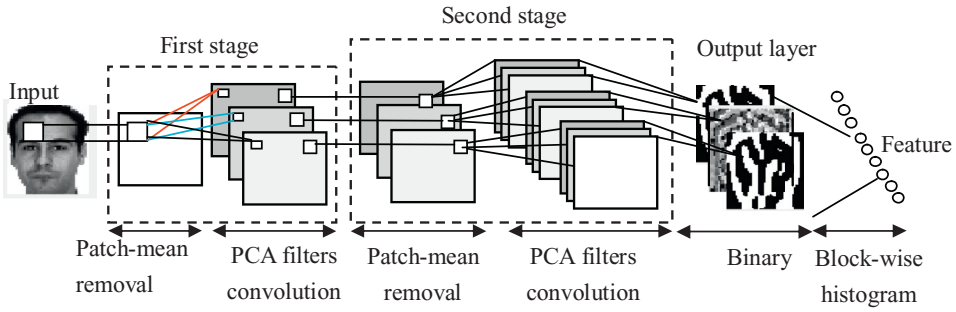


Fig. 1. The architecture of PCANet model.

Inspired by the advantages of deep structure and efficiency of traditional subspace learning methods for parameter learning, recently, many deep subspace learning methods have been widely studied for feature extraction. For example, Liong et al. [26] proposed a two-layer PCA network for image recognition. In each layer, ZPCA plus PCA is employed to learn parameters for learning hierarchical features. ZPCA aims to learn parameters such that the obtained features have uncorrelated components and unity variances. It can be viewed as a pooling layer and cannot well exploit spatial information embedded in pixels. To well exploit spatial information, Chan et al. [27] employed PCA to learn weights of CNNs and proposed PCANet which comprises cascaded PCA, binary hashing, and block-wise histograms. Tian et al. [28,29] proposed stacked PCA Network (SPCANet) and MS-PCANet which obtain the final features by stacking multiple output features of PCANet. It is easy to see that the learned features not only contain large redundancy but also have high-dimensionality. Moreover, these deep subspace architectures ignore the pooling layer.

Most existing works have demonstrated that pooling layer in deep learning helps improve the robustness of algorithms and different sematic features are important for image classification [30–34]. Moreover, pixels offset can help exploit more spatial information among nearby pixels in image that is important for classification [35]. Inspired by the aforementioned facts, we proposed a deep learning framework named Improved PCA Network (EPCANet). Each stage comprises convolutional filter layer and spatial pooling layer. Different from traditional CNN, the convolutional filter layer of EPCANet uses PCA to learn the filter kernels, therefore, tuning parameter process are avoided and training time is remarkably reduced. Subsampling or pixel staggered [35] is used in spatial pooling layer before dividing images into paches. It helps to improve stables of representation. After that, it is followed by simple binary hashing and block histograms for indexing and pooling. Experiments on several databases illustrate the efficiency of our method.

The rest of this paper is organized as follows: Section 2 introduces PCANet and SPCANet. In Section 3 we describe the framework of EPCANet. Section 4 reports the experimental results. Section 5 concludes this paper.

2. PCANet and SPCANet

2.1. PCA network (PCANet)

PCANet is a simple learning network based on CNN for image classification. Fig. 1 shows the diagram of PCANet [27]. PCANet is extremely simple and efficient which only have three parameters: layer number, filter number, and filter size. In PCANet, filter kernels can be efficiently learned by traditional PCA method. It avoid learning filter kernels iteratively as in traditional CNNs. After that Binary hashing and block-wise histograms are applied to output layer of the last stage.

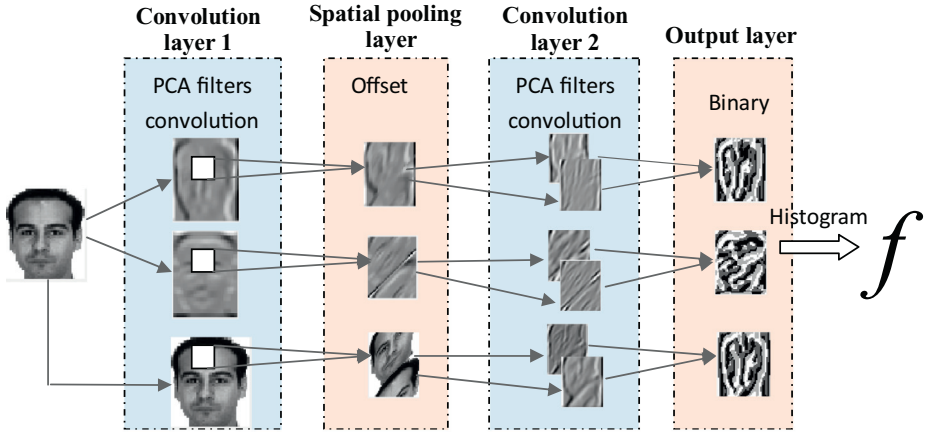
Extensive experiments illustrate that PCANet is remarkably superior to traditional subspace methods for image analysis. Moreover, compared with hand-crafted features such as LBP and LQP, the learned features can be widely used in many applications. Motivated by PCANet, some variations of PCANet are proposed, such as RandNet and LDANet which employ random filter and LDA filter to learn filter kernels in CNN [27], respectively. Although PCANet is simple and effective for image classification, it cannot well encode more efficient discriminant information due to the fact that the learned features is only composed of output of the last layer and do not well characterize spatial information.

2.2. Stacked PCA network (SPCANet)

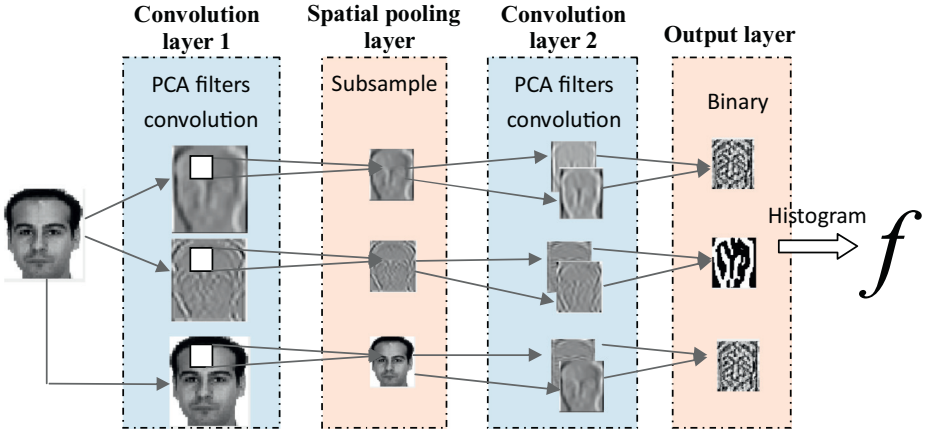
Stacked PCA Network (SPCANet) [28] is an effective improve method of PCANet, which also has similar processing components with PCANet. The main difference between SPCANet and PCANet is that the feature of each stage is extracted through the binary hashing and block-wise histograms in SPCANet, and the final feature is obtained by stacking multiple output features which reveal different discriminant information. So, compared with PCANet, SPCANet well encodes more discriminant information. However, by concatenating multiple output features, the final features contain much redundancy, which may make performance degrade. Moreover, neither SPCANet nor PCANet contain pooling layer which helps improve robustness of algorithm [30]. This reduces the flexibility of algorithms. To solve this problem, we propose an enhanced PCANet (EPCANet) for image recognition in Section 3.

3. Enhanced PCA network (EPCANet)

Many studies have shown that pooling layer not only helps to improve robustness against noise of algorithm but also make the representation invariant to the translation of the input [30–34]. Moreover, different sematic features reveal different discriminant information that is important for image classification [12,35]. Finally, pixels offset can help exploit more spatial information among nearby pixels in image that is important for classification [12]. Motivated by these facts, we propose a deep subspace learning framework named Enhanced PCA Network (EPCANet). The structure of EPCANet consists of two convolution layers, a spatial pooling layer and an output layer. We show the model with pixel offset staggered pooling and subsample pooling respectively in Fig. 2. Assume that we have N input training images $\{\mathbf{X}_i\}_{i=1}^{N}$ of size $m \times n$, and suppose that the number of filters in layer i is L_i and the patch size is $k_1 \times k_2$ at all stages.



(a) Our model with pixel offset staggered pooling.



(b) Our model with subsample pooling.

Fig. 2. The architecture of EPCANet model.

3.1. The convolutional filter layer

Around each pixel, we take a $k_1 \times k_2$ patch, and we collect all (overlapping) patches of the ith image and vectorize them, i.e., $x_{i,1}, x_{i,2}, ..., x_{i,mn} \in R^{k_1 k_2}$. Then we subtract patch mean from each patch and obtain $\bar{\mathbf{X}}_i = [\bar{x}_{i,1}, \bar{x}_{i,2}, \cdots, \bar{x}_{i,mn}]$. By constructing the same matrix for all input images and putting them together, we get

$$\mathbf{X} = [\bar{\mathbf{X}}_1, \bar{\mathbf{X}}_2, \ldots, \bar{\mathbf{X}}_N] \in R^{k_1 k_2 \times mnN} \tag{1}$$

We conduct PCA on $\mathbf{X}$. PCA aims to find a set of projection vectors by minimizing the reconstruction error, i.e.,

$$\min_{\mathbf{V} \in R^{k_1 k_2 \times L_1}} \left\| \mathbf{X} - \mathbf{V}\mathbf{V}^T\mathbf{X} \right\|_2^2, s.t. \mathbf{V}^T\mathbf{V} = \mathbf{I}_{L_1} \tag{2}$$

where $\mathbf{I}_{L_1}$ is identity matrix of size $L_1 \times L_1$ and $\mathbf{V} = [\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_{L_1}] \in R^{k_1 k_2 \times L_1}$. The solution of Eq (2) is known as the L_1 eigenvectors corresponding to the first L_1 largest eigenvalues of $\mathbf{X}\mathbf{X}^T$. Then we set the solution as the corresponding convolutional filters.

For the first stage, the filter kernels of CNN are:

$$\mathbf{W}_p^1 = \mathrm{mat}_{k_1,k_2}(\boldsymbol{v}_p) \in R^{k_1 \times k_2}, p = 1, 2, \cdots, L_1 \tag{3}$$

where $\mathrm{mat}_{k_1,k_2}(\boldsymbol{v})$ is a function that maps $\boldsymbol{v} \in R^{k_1 k_2}$ to a matrix $\mathbf{W} \in R^{k_1 \times k_2}$, and $\boldsymbol{v}_p$ means the pth vector in matrix $\mathbf{V}$.

The pth convolution output of the image $\mathbf{X}_i$ on the first stage can be represented as

$$\mathbf{A}_i^p = \mathbf{X}_i * \mathbf{W}_p^1, i = 1, 2, \cdots, N \tag{4}$$

where $*$ denotes 2D convolution, in order to make $\mathbf{A}_i^p$ have the same size of $\mathbf{X}_i$, the boundary of $\mathbf{X}_i$ is zero-padded before convolving with $\mathbf{W}_p^1$.

3.2. Spatial pooling layer

Subsample has been widely used as a pooling layer in CNN that is one of the most popular architectures in dep learning, and may provide different scale features that help to improve stableness of algorithms [30–34]. Moreover, pixels offset can help exploit more spatial information among nearby pixels in image that is important for classification [35]. Motivated by these facts, we respectively employ subsample and pixel offset to get different features in pooling layer in our architecture. Moreover, in order to well exploit more nonlinear features embedded in images, input images and output features on the first stage are put together as the input data of spatial pooling layer in our model. In the following section, we first introduce pixel offset, and then simple describe subsample.

3.2.1. Pixel offset staggered

Pixel offset is a method for image transformation and it can be expressed as [35]

$$\tilde{\mathbf{A}} = \mathrm{offset}(\mathbf{A}) \tag{5}$$

where the function offset$(\cdot)$ means pixel offset staggered function, i.e. for an image, we move the bottom of the pixel of each column

(a) Pixel offset staggered (b) Image-Subsampling

Fig. 3. Results of pixel offset staggered to one image and image-subsampling to one image.

upward and use the top pixels to fill the vacancy at the bottom of the column according to some certain rules. It disrupts the distribution of the original pixels so that pixels on different rows now lie on the same row. Pixel offset also changes the relationship between the pixels and makes some of the spatial information hidden in the original image exposed. Fig. 3(a) shows a kind of rule of pixel offset staggered. If this method is applied in EPCANet for spatial pooling, the network is denoted as **EPCANet -offset.**

3.2.2. Image-subsampling

Subsampling performs a local averaging on images and it can be defined as:

$$\tilde{\mathbf{A}} = \text{down}_{n\times n}(\mathbf{A}) \tag{6}$$

where $\text{down}_{n\times n}(\cdot)$ denotes the subsample function. Computing the average value of every non-overlapping $n \times n$ region of one image is a typical operation of subsample. In this paper, we set $n = 2$, then the output image $\tilde{\mathbf{A}}$ will have half the number of rows and columns as the original image $\mathbf{A}$. Fig. 3(b) shows a kind of rule of image subsampling. Subsampling can reduce the sensitivity of the output to shift and distortion. If EPCANet uses this method for spatial pooling, we denote the network as **EPCANet-subsample.**

3.3. Output Layer: binary hashing and histogram

Suppose $\tilde{\mathbf{A}}_i^p$ denotes the result of $\mathbf{A}_i^p$ after spatial pooling layer. Similar to the first stage, we can compute the PCA filters $\mathbf{W}_q^2 (q = 1, 2, \cdots, L_2)$ of the second stage, then each input image $\tilde{\mathbf{A}}_i^p$ of the second stage will have L_2 outputs by convolving with $\mathbf{W}_q^2$ for $q = 1, 2, \cdots, L_2$:

$$\mathbf{O}_i^p = \{\tilde{\mathbf{A}}_i^p * \mathbf{W}_q^2\}_{q=1}^{L_2} \tag{7}$$

For each training sample, the number of outputs on the first stage and second stage are L_1 and L_1L_2, respectively. Similar to CNN, we can stack multiple stages of PCA filters to extract higher level features.

3.3.1. Binary hashing

The outputs of the second stage are binarized by Heaviside step function $H(\cdot)$ whose value is one for positive entries and zero otherwise, thus we have $\{H(\tilde{\mathbf{A}}_i^p * \mathbf{W}_q^2)\}_{q=1}^{L_2}$. Then we sum L_2 binary outputs with their bits weighted, which converts the L_2 outputs in the second stage of $\tilde{\mathbf{A}}_i^p$ back into a decimal-valued image $\mathbf{D}_i^p$:

$$\mathbf{D}_i^p = \sum_{q=1}^{L_2} 2^{q-1} H(\tilde{\mathbf{A}}_i^q * \mathbf{W}_q^2) \tag{8}$$

whose every pixel is an integer in the range $[0, 2^{L_2} - 1]$.

3.3.2. Histogram

For each $\mathbf{D}_i^p$, we partition it into B blocks whose size is $h_1 \times h_2$, and compute the histogram of the decimal values in each block, and concatenate all the B histograms into one vector denoted as $f_i^p = BlkHist(\mathbf{D}_i^p)$. After this encoding process, the feature of the input image $\mathbf{X}_i$ is then defined to be the set of block-wise histograms, i.e.,

$$\boldsymbol{f}_i = [BlkHist(\mathbf{D}_i^1), \cdots, BlkHist(\mathbf{D}_i^{L_1})]^T \tag{9}$$

The algorithm of the EPCANet is concluded as follows:

Input: training data $\{\mathbf{X}_i\}_{i=1}^N$, $\mathbf{X}_i \in R^{m\times n}$, L_1, L_2, The patch size is$k_1 \times k_2$.
Convolution layer 1:
- •Patch mean-removal:$\bar{\mathbf{X}} = [\bar{\mathbf{X}}_1, \bar{\mathbf{X}}_2, \cdots, \bar{\mathbf{X}}_N] \in R^{k_1k_2\times mnN}$
- •Compute convolution kernels use PCA:
  $\mathbf{V} = \min_{\mathbf{V}\in R^{k_1k_2\times L_1}} \|\mathbf{X} - \mathbf{V}\mathbf{V}^T\mathbf{X}\|_2^2, s.t. \mathbf{V}^T\mathbf{V} = \mathbf{I}_{L_1}$
  $\mathbf{W}_p^1 = \text{mat}_{k_1,k_2}(\boldsymbol{v}_p) \in R^{k_1\times k_2}, p = 1, 2, \cdots, L_1$
- •Output: $\mathbf{A}_i^p = \mathbf{X}_i * \mathbf{W}_p^1, i = 1, 2, \cdots, N$

Spatial pooling layer: $\tilde{\mathbf{A}}_i^p = \text{offset}(\mathbf{A}_i^p)$ or $\tilde{\mathbf{A}}_i^p = \text{down}_{n\times n}(\mathbf{A}_i^p)$
Convolution layer 2:
- •Compute convolution kernels:$\mathbf{W}_q^2, q = 1, 2, \cdots, L_2$
- •Output: $\mathbf{O}_i^p = \{\tilde{\mathbf{A}}_i^p * \mathbf{W}_q^2\}_{q=1}^{L_2}$

Output layer:
- •Binary Hashing:compute the decimal-valued image
  $\mathbf{D}_i^p = \sum_{q=1}^{L_2} 2^{q-1} H(\tilde{\mathbf{A}}_i^p * \mathbf{W}_q^2)$
- •Histogram: $\boldsymbol{f}_i = [BlkHist(\mathbf{D}_i^1), \cdots, BlkHist(\mathbf{D}_i^{L_1})]^T$
- •Output: $\mathbf{F} = [\boldsymbol{f}_1, \boldsymbol{f}_2, \cdots, \boldsymbol{f}_N]$

---

4. Experiments

In this section, we validate the performance of EPCANet with SVM classifier, and compare it with PCANet and SPCANet. In the experiments, we perform these algorithms on the CMU PIE, AR, Extended Yale B, ORL and LFW for face recognition.

4.1. Experiments on CMU PIE database

The CMU PIE database [36] contains 41,368 face images belonging to 68 subjects. The face images were captured by 13 synchronized cameras and 21 flashes, under varying pose, illumination and expression. Fig. 4 shows some sample images of one subject. We selected C05 images as gallery that includes 49 samples per person, and each image was manually cropped and resized to 64×64 pixels. We select the first 5 images from each class as training samples and the remaining for test.

In the experiments, we set the patch size of network as $k_1 = k_2 = 5$ and the size of non-overlapping block as 7×7. For the sake of representation, we fix $L_2 = 8$ and vary the filter number L_1 of the first stage from 2 to 16. We compare EPCANet with PCANet and SPCANet [27]. The results are shown in Fig. 5.

4.2. Experiments on AR database

The AR face database [38] consists of 3120 frontal-face pictures of 120 individuals with facial expressions, lighting conditions, and occlusions (sun glasses and scarves) changes. The size of the images is 50×40. For each individual, 26 pictures were taken in two separate sessions. We do three group experiments in this database. In the first group experiment, 4 different expressions images of the second session are selected as the testing samples. In the second

Fig. 4. Some sample images of one subject on CMU PIE database.

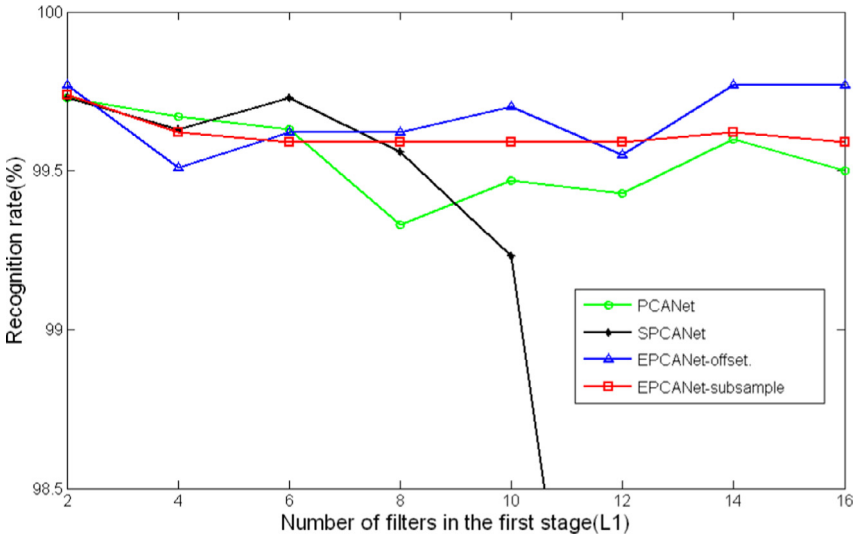


Fig. 5. Recognition rate vs. different value of L_1 on CMU PIE database.

Table 1
Recognition rates (%) on AR database.

Experiment	1st	2nd	3rd
PCANet	93.33	94.44	93.81
SPCANet	93.54	94.17	94.17
EPCANet-offset	93.54	93.33	93.45
EPCANet-subsample	**94.17**	**94.72**	**94.40**

Table 2
Recognition rates (%) on the Extended Yale B database.

Percent occlusion	0%	10%	20%	30%	40%
PCANet	90.91	83.14	77.20	65.32	55.65
SPCANet	90.32	83.36	78.45	67.16	58.14
EPCANet-offset	**92.30**	**85.41**	**78.81**	**67.96**	**58.50**
EPCANet-subsample	**93.91**	**88.42**	**81.60**	**68.62**	56.60

group experiment, 3 illumination images of the second session are selected as the testing samples. In the third group experiment, 4 different expressions and 3 illumination images of the second session are selected as the testing samples. For all of the aforementioned experiments, we select 4 different expressions and 3 illumination conditions of the first session as training samples. In this experiment, we set $L_1 = L_2 = 8$, size of the patch is $k_1 = k_2 = 5$ and the size of non-overlapping block is 7×7. Table 1 shows the experimental results.

4.3. Experiments on the extended Yale B database

The Extended Yale B dataset [39] consists of 2414 frontal-face images of 38 individuals with the resolution 32×32 and illumination changes. There are 64 images for each object except 60 for 11th and 13th, 59 for 12th, 62 for 15th and 63 for 14th, 16th and 17th. In our experiment, the first 20 images of each class are selected as training samples, the remaining images for test. In order to further verify the robustness, we also simulate various levels of contiguous occlusion, from 0 percent to 40 percent, by replacing a random region located square block of each test image with an unrelated image as described in Fig. 6.

In this experiment, we set $L_1 = L_2 = 8$, the size of patch is $k_1 = k_2 = 5$ and the size of non-overlapping block is 7×7. Table 2 shows the experimental results.

4.4. Experiments on ORL database

The ORL database [37] contains a set of face images taken between April 1992 and April 1994. There are 40 distinct subjects and each subject contains ten different images with illumination changes. The size of the image is 32×32. For some subjects, the images were taken in different time, varying the lighting, facial expressions (open/closed eyes, smiling/not smiling) and facial details (glasses/no glasses). All images were taken against a dark homogeneous background with the subjects in an upright, frontal position (with tolerance for some side movement). Fig. 7 shows some sample images of one subject. In this experiment, the first 5 images of each class are selected as training samples, and the rest 5 images of each class are used for testing. In this database, we also set $L_2 = 8$ and vary the number of filters in the first stage from 2 to 16. The patch size of the network is set $k_1 = k_2 = 5$ and the size of non-overlapping block is 7×7. The result is showed in Fig. 8.

4.5. Experiments on LFWcrop database

Besides the above face databases, we also evaluate the EPCANet on the LFWcrop dataset [40]. LFWcrop dataset is a cropped version of the Labeled Faces in the Wild (LFW) [41] dataset, keeping only the center portion of each image (i.e. the face). In the vast majority of images, almost all of the background is omitted. The extracted

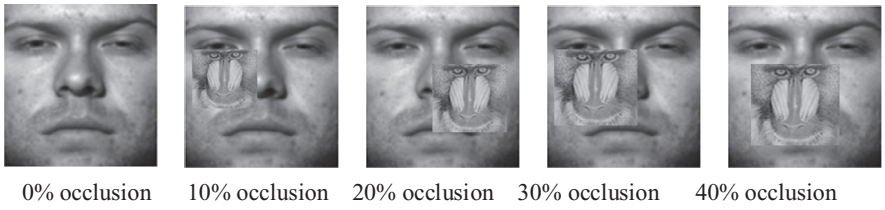


Fig. 6. Various levels (0–40%) of block occlusion on one image of Extended Yale B Database.

Fig. 7. Some sample images of two subjects on ORL database.

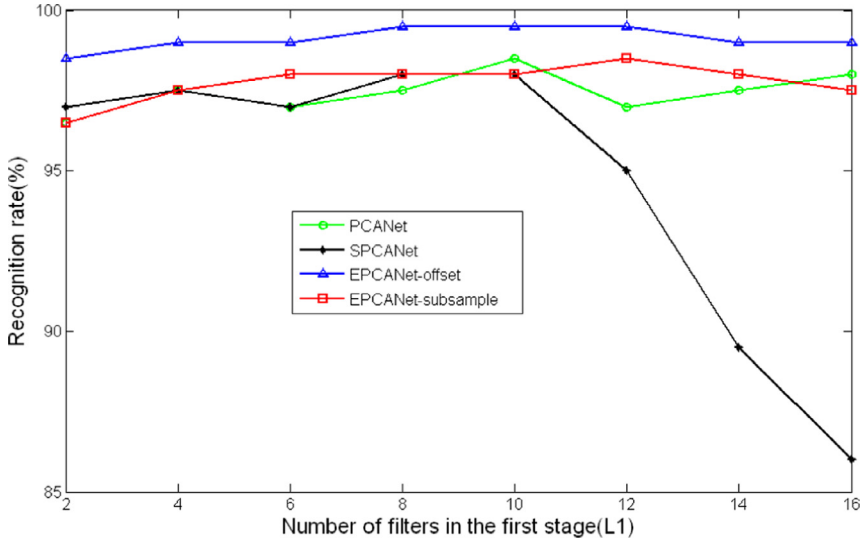


Fig. 8. Recognition rate vs. different value of L_1 on ORL database.

Table 3
Recognition rates (%) on the LFWcrop database.

Number of filters in Layer One	8	10	12	14	16
PCANet	86.24	86.70	87.16	84.40	84.86
SPCANet	**87.61**	87.61	83.94	63.76	49.08
EPCANet-offset	**87.61**	**88.53**	**87.61**	**87.61**	**86.24**
EPCANet-subsample	85.78	84.86	84.40	83.94	83.94

area was then scaled to a size of 64×64 pixels. The selection of the bounding box location was based on the positions of 40 randomly selected LFW faces. As the location and size of faces in LFW was determined through the use of an automatic face locator (detector) [13], the cropped faces in LFWcrop exhibit real-life conditions, including misalignment, scale variations, in-plane as well as out-of-plane rotations. In the experiment, we choose person who has more than 20 photos but less than 100 photos as the subdataset, which contains 57 classes and 1883 images. For each person, we randomly choose ninety percent of images for training, and the remaining images for testing. In this experiment, we set $L_2 = 8$, the size of patch is $k_1 = k_2 = 5$ and the size of non-overlapping block is 7×7. The experimental results are given in Table 3.

Comparing the aforementioned experiments, several interesting observations as follows:

(1) From the above experimental results, we can see EPCANet-offset performs better in CMU PIE, LFWcrop and ORL databases, while EPCANet-subsample performs better in AR and Extended Yale B databases.
(2) One can see that our algorithms (EPCANet-offset and EPCANet-subsample) have better performance on face recognition than PCANet or SPCANet in CMU PIE and ORL database. When the filters number (the value of L_1) in the first stage is more than 8, our algorithms show obvious superiority than PCANet. With the increase of the filters number in the first stage, the performance of SPCANet dropped dramatically. Furthermore, in LFWcrop database, with the number of filters in layer one increase, the recognition rates of SPCANet dropped dramatically. This is probably because that the dimension of features by SPCANet is much higher than other methods especially when layer one has lots of filters, thus extracted features contains redundant information, which is bad for the final classification.
(3) As can be seen the results on CMU PIE, LFWcrop and ORL databases, EPCANet- offset consistently performs

better than EPCANet-subsample. This is probably because that the method of pixel offset staggered exposes the spatial information hidden in original images, which is important for classification.

(4) Different from the aforementioned three databases, the test images in AR and Extended Yale B databases contain occlusion noise. The performance of EPCANet-subsample is superior to EPCANet-offset in these two databases. This is probably because that the method of subsampling can reduce the resolution of original images, which can further weaken the impact of occlusion.

5. Conclusion

In this paper, we have proposed arguably the simplified convolutional deep learning framework— EPCANet to learn filter kernels through PCA instead of SGD in CNN. In addition, there is a pooling layer between the two neighboring stages. The data processing method of pooling layer can be image pixel offset or image-subsample. In the output layer, the binary hashing and histogram are used to process the output of the second stage to obtain final feature. The results of extensive experiments demonstrate that our model is insensitive to illumination and is robust to occlusions. Besides, the results of SPCANet indicate that more redundant and the high dimension of the final feature is not good for classification. EPCANet outperforms the state-of-the-artface recognition methods in most cases.

It is easy to see that, EPCANet combines the idea of deep learning and PCA to learn more efficient sematic features for image analysis. PCA is employed to learn filter kernels. It avoids parameter adjustment. Since apart from PCA, there are many other subspace methods such as locality preserving projection (LPP), Neighborhood preserving projection (NPE), and so on. Thus, we can also employ them to learn filter kernels. We will study them in our future work.

Acknowledgments

The authors would like to thank the anonymous reviewers and AE for their constructive comments and suggestions, which improved the paper substantially. This work is supported by National Natural Science Foundation of China under Grant 61271296, China Postdoctoral Science Foundation (Grant 2012M521747), and the 111 Project of China (B08038).

References

[1] P.N. Belhumeur, J.P. Hespanha, D.J. Kriegman, Eigenfaces vs. fisherfaces: Recognition using class specific linear projection, IEEE Trans. Pattern Anal. Mach. Intel. 19 (7) (1997) 711–720.
[2] Y. Hua, Y. Jie, A direct LDA algorithm for high-dimensional data — with application to face recognition, Pattern Recog. 34 (10) (2001) 2067–2070.
[3] M.A. Turk, A.P. Pentland, Face recognition using eigenfaces, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1991, 1991, pp. 586–591.
[4] X. He, D. Cai, S. Yan, H.J. Zhang, Neighborhood preserving embedding, in: Proceedings of the 10th IEEE International Conference on Computer Vision, 2005, pp. 1208–1213.
[5] Q. Gao, H. Zhang, J. Liu, Two-dimensional margin, similarity and variation embedding, Neurocomputing 86 (2012) 179–183.
[6] Q. Wang, F. Chen, Q. Gao, X. Gao, F. Nie, On the schatten norm for matrix based subspace learning and classification, Neurocomputing 216 (2016) 192–199.
[7] J. Wang, D. Shi, D. Cheng, Y. Zhang, J. Gao, LRSR: Low-rank-sparse representation for subspace clustering, Neurocomputing 214 (2016) 1–12.
[8] P. Gruber, K. Stadlthanner, M. Hm, F.J. Theis, E.W. Lang, Denoising using local projective subspace methods, Neurocomputing 69 (13–15) (2006) 1485–1501.
[9] T. Ahonen, A. Hadid, M. Pietikäinen, Face description with local binary patterns: application to face recognition, 28(12)(2006), 2037–2041.
[10] D.G. Lowe, Distinctive image features from scale-invariant keypoints, Int. J. Comput. Vis. 60 (2) (2004) 91–110.
[11] A. Fischer, C. Igel, Training RBMs based on the signs of the CD approximation of the log-likelihood derivatives, in: Proceedings of the ESANN, 2011.
[12] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, F.E. Alsaadi, A survey of deep neural network architectures and their applications, Neurocomputing 234 (2017) 11–26.
[13] G.E. Hinton, A practical guide to training restricted Boltzmann machines, in: Neural Network: Tricks Trade (2012) 599–619.
[14] N. Zeng, Z. Wang, H. Zhang, W. Liu, F.E. Alsaadi, Deep belief networks for quantitative analysis of a gold immunochromatographic strip, Cognitive Comput. 8 (4) (2016) 684–692.
[15] N. Zeng, H. Zhang, W. Liu, J. Liang, F.E. Alsaadi, A switching delayed PSO optimized extreme learning machine for short-term load forecasting, Neurocomputing 240 (2017) 175–182.
[16] G.E. Dahl, D. Yu, L. Deng, A. Acero, Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition, IEEE Trans. Audio, Speech, Lang. Process. 20 (1) (2012) 30–42.
[17] A.M. Abdel Zaher, A.M. Eldeib, Breast cancer classification using deep belief networks, Exp. Syst. Appl. 46 (2016) 139–144.
[18] N. Zeng, Z. Wang, H. Zhang, Inferring nonlinear lateral flow immunoassay state-space models via an unscented Kalman filter, Sci. China Inform. Sci. 59 (11) (2016) 112204.
[19] H. Bourlard, Y. Kamp, Auto-association by multilayer perceptrons and singular value decomposition, Biol. Cybern. 59 (4–5) (1988) 291–294.
[20] Y. Bengio, Learning deep architectures for AI, Found. Trends Mach. Learn. 2 (1) (2009) 1–127.
[21] L. Deng, Three classes of deep learning architectures and their applications: a tutorial survey, APSIPA Trans. Signal Inf. Process. (2012).
[22] I.J. Kim, X. Xie, Handwritten hangul recognition using deep convolutional neural networks, Int. J. Doc. Anal. Recognit. 18 (1) (2015) 1–13.
[23] Y. Kim, T. Moon, Human detection and activity classification based on microdoppler signatures using deep convolutional neural networks, IEEE Geosci. Remote Sens. Lett. 13 (1) (2015) 8–12.
[24] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, Adv. Neural Inf. Process. Syst. 25 (2012) 1097–1105.
[25] T.N. Sainath, B. Kingsbury, G. Saon, H. Soltau, A.r. Mohamed, G. Dahl, B. Ramabhadran, Deep convolutional neural networks for large-scale speech tasks, Neural Netw. 64 (2015) 39–48.
[26] V.E. Liong, J. Lu, G. Wang, Face recognition using Deep PCA„ in: Proceedings of the 2013 9th International Conference on Information Communications and Signal Processing (ICICS), IEEE, 2013, pp. 1–5.
[27] T.H. Chan, K Jia, S. Gao, J. Lu, Z. Zeng, Y. Ma, PCANet: A simple deep learning baseline for image classification? IEEE Trans. Image Process. 24 (12) (2015) 5017–5032.
[28] L. Tian, C. Fan, Y. Ming, Y. Jin, Stacked PCA Network (SPCANet): An effective deep learning for face recognition, in: Proceedings of the IEEE International Conference on Digital Signal Processing (DSP)„ 2015, pp. 1039–1043.
[29] L. Tian, C. Fan, Y. Ming, Multiple scales combined principle component analysis deep learning network for face recognition, J. Electronic Imaging 25 (2) (2016) 023025.
[30] Y.L. Boureau, J. Ponce, Y. LeCun, A theoretical analysis of feature pooling in visual recognition, in: Proceedings of the 27th International Conference on Machine Learning (ICML-10), 2010, pp. 111–118.
[31] Y.L. Boureau, N.L. Roux, F. Bach, J. Ponce, Y. LeCun, Ask the locals: multi-way local pooling for image recognition, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2011, 2011, pp. 2651–2658.
[32] M.D. Zeiler, R. Fergus, Stochastic pooling for regularization of deep convolutional neural networks, arXiv:1301.3557, 2013.
[33] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324.
[34] Y. Jia, C. Huang, T. Darrell, Beyond spatial pyramids: Receptive field learning for pooled image features, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012, pp. 3370–3377.
[35] L. Zhang, Q. Gao, D. Zhang, Directional independent component analysis with tensor representation, in: Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2008, pp. 1–7.
[36] T. Sim, S. Baker, M. Bsat, The CMU Pose, Illumination, and Expression (PIE) database, in: Proceedings of the IEEE Conference on Automatic Face and Gesture Recognition, 2002, pp. 46–51.
[37] B. Anton, J. Fein, T. To, X. Li, L. Silberstein, C.J. Evans, Immunohistochemical localization of ORL in the central nervous system of the rat, J. Compar. Neurol. 368 (2) (1996) 229–251.
[38] A.M. Martinez, The AR face database, CVC Technical Report, 1998, 24.
[39] S. Georghiades, P.N. Belhumeur, D. Kriegman, From few to many: Illumination cone models for face recognition under variable lighting and pose, IEEE Trans. Pattern Anal. Mach. Intel. 23 (6) (2001) 643–660.
[40] C. Sanderson, B.C. Lovell, Multi-region probabilistic histograms for robust and scalable identity inference, Advances in Biometrics, Springer, Berlin Heidelberg, 2009, pp. 199–208.
[41] G.B. Huang, M. Mattar, T. Berg, E. Learned-Miller, Labeled faces in the wild: a database for studying face recognition in unconstrained environments, Month, 2008.

Yang Liu received the B.Eng. degree in communication engineering from Xidian University, China, in 2013. He is currently working toward the Ph.D. degree in communication & information system from Xidian University, China. His research interests include pattern recognition, dimensionality reduction, sparse representation, and face recognition.

Shuangshuang Zhao received the B. Eng. degree in electronic information science and technology from Hebei University, China, in 2014, and She is currently working toward the M.S. degree in Traffic Information Engineering & Control from Xidian University, China. Her research interests include pattern recognition, and dimensionality reduction, and face recognition.

Qianqian Wang received the B.Eng. degree in communication engineering from Lanzhou University of Technology, China, in 2014. She is currently working toward the Ph.D. degree in communication & information system from Xidian University, China. Her research interests include pattern recognition, dimensionality reduction, sparse representation, and face recognition.

Quanxue Gao received the B.Eng.degree from xi'an Highway University, Xi'an, China, in 1998, the M.S. degree from the Gansu University of Technology, Lanzhou, China,in 2001, and the Ph.D. degree from Northwestern Polytechnical University, X'ian China, in 2005.He was a associate research with the Biometrics Center, The Hong Kong Polytechnic University, Hong Kong from 2006 to 2007. He is currently a professor with the School of Telecommunications Engineering, Xidian University, and also a key member of State Key Laboratory of Integrated Services Networks. His current research interests include pattern recognition and machine learning.