Pattern Analysis and Applications (2020) 23:673–682
https://doi.org/10.1007/s10044-019-00818-y

THEORETICAL ADVANCES

PCAPooL: unsupervised feature learning for face recognition using PCA, LBP, and pyramid pooling

Amani Alahmadi[1] · Muhammad Hussain[1] · Hatim A. Aboalsamh[1] · Mansour Zuair[2]

Received: 14 March 2017 / Accepted: 25 March 2019 / Published online: 10 May 2019

Abstract
Human face is a widely used biometric modality for verification and revealing the identity of a person. In spite of a great deal of research on face recognition, it still is a challenging issue. Recently, the outstanding performance of deep learning has attracted a good deal of research interest for face recognition. In comparison with hand-engineered features, learning-based face features have proven their superiority in encoding discriminative information. Inspired by deep learning, we introduce a simple and efficient unsupervised feature learning scheme for face recognition. This scheme employs principle component analysis (PCA), local binary pattern (LBP), and pyramid pooling. Following the architecture of a convolutional neural network, the proposed scheme contains three types of layers: convolutional, nonlinear, and pooling layers. PCA is used to learn a filter bank for the convolutional layer. This is followed by LBP operator that encodes the local texture and adds nonlinearity to the feature maps of convolutional layer, which are then pooled using spatial pyramid pooling. To corroborate the effectiveness of the scheme (which we call as PCAPool), extensive experiments were performed on challenging benchmark databases: FERET, Yale, Extended Yale B, AR, and multi-PIE. The comparison reveals that PCAPool performs better than the state-of-the-art methods.

Keywords Face recognition · Convolutional neural network · Local binary pattern · Pyramid pooling · Principal component analysis

1 Introduction

Face recognition still is an active research area due to its wide range of applications and the challenging nature of face images. The appearance of the same face is highly affected by different expressions, poses, illumination conditions, and occlusions, which increase intra-class variability. The key factor to overcome these variations is to find robust and discriminant facial representations.

The earlier efforts in the literature are focused on extracting holistic features using linear discriminant analysis (LDA) [1] and principle component analysis (PCA) [2]. The sensitivity of the holistic techniques to different types of variations, such as illumination, pose, and expression variation, motivated the researchers toward local face descriptors. Numerous works have been proposed on designing local features like Gabor features [3, 4], local binary pattern (LBP) [5]. These hand-crafted techniques can perform well for a certain data but cannot be generalized for new conditions.

Sparse representation classification (SRC) has improved the performance in face recognition. In SRC, the classification decision is made based on the least reconstruction error of a probe image. After the success of its first application [6], different variations of SRC have been proposed [7]. Deng et al. [8] introduced an adaptive technique, superposed sparse representation-based classification (SSRC), in which the recognition is based on computing the sparse representation of a probe image as a superposition of the class centroids. SRC with Gabor-based occlusion dictionary was presented by Yang et al. [9] to enhance the robustness against occlusions.

The spatial pyramid matching has shown a significant improvement in image classification tasks [10, 11]. Shen

✉ Muhammad Hussain
mhussain@ksu.edu.sa

[1] Department of Computer Science, College of Computer and Information Sciences, King Saud University, Riyadh 51178, Saudi Arabia

[2] Department of Computer Engineering, College of Computer and Information Sciences, King Saud University, Riyadh 51178, Saudi Arabia

et al. [12] developed an efficient method for face recognition, which is based on multi-level spatial pooling of multi-scale local patches. This technique is very simple, and it achieved a high performance on challenging benchmark databases.

Recently, feature learning has received significant attention to cope with the limitations of hand-crafted techniques. With the advances in computational power and resources, deep learning has become an active research field due to its ability to represent abstract semantics of data. A popular deep learning-based technique is convolutional neural network (CNN) [13–15], which involves many layers stacked on top of each other; commonly used layers are convolutional, nonlinear, and pooling layers. Zhu et al. [16] designed a neural network scheme to reduce the intra-class variation and preserve the discriminative information. Sun et al. [17] learned an ensemble of convolutional networks, which generalizes well to face recognition task. The performance of CNN depends on its depth in addition to the tuning of its learnable parameters. In other words, when the architecture of CNN is not deep enough, its performance will not be as good as those of hand-engineered features.

A simple network PCANet proposed by Chan et al. [18] achieved remarkable success for many classification tasks including face recognition. PCANet is a kind of CNN that learns filters for convolutional layers in an unsupervised manner using principal component analysis (PCA). It consists of two convolutional layers, which are followed by binary hashing and block wise histogram computation.

Motivated by the success of PCANet and spatial pyramid pooling, we introduce an effective and simple unsupervised feature learning scheme, where the structure is not as deep as PCANet and CNN, but it is effective in learning discriminative features. It includes one convolutional layer, one nonlinear layer, and a pooling layer. The filters of the convolutional layer are learned in an unsupervised manner using PCA like PCANet, and LBP is used to introduce nonlinearity and extract micro-texture patterns. Finally, spatial pyramid pooling is used for pooling layer. We call this scheme as PCAPooL. Its performance was evaluated on five challenging benchmark face databases: FERET, Yale, Extended Yale B, AR, and multi-PIE.

The comparison indicates that PCAPool performs better than the state-of-the-art methods. Particularly, it outperforms the method based on spatial pyramid pooling (SPP) [12] by 8.36% and 5.61% on DupII and DupI sets of FERRET, respectively. It achieved an accuracy of 99.60% on AR with approximately 1% improvement over SPP (98.7% in [12]). As compared to single-stage PCANet (PCANet-1), PCAPool has improvement of 7.11% and 4.82% on DupII and DupI, respectively, and it achieved almost similar results to those by PCANet-2 (2-stages) on DupII and DupI and better results on Fb. Furthermore, it is important to note that PCAPooL involves the computation of filters using PCA for only one layer, as such it is computationally efficient as compared to PCANet, and other state-of-the-art multi-stage deep learning methods.

The main contributions of the paper are summarized below:

- The architecture of PCAPool is very simple and does not require a huge volume of data for learning. Instead of having many convolutional, pooling, and fully connected layers, PCAPooL has only one convolutional layer and one nonlinear layer, followed by a deep spatial pyramid pooling layer. It constructs a simple but effective unsupervised feature learning architecture that mimics the framework of the deep CNN architecture.
- PCAPooL introduces nonlinearity with LBP layer and multi-scale coding of features through spatial pyramid pooling and as such being simpler in architecture than PCANet, it gives comparable performance. Also, it consumes less memory than PCANet and can be managed on ordinary computer.
- To the best of our knowledge, this is the first attempt to employ LBP as a nonlinear layer, which enhances the classification performance, and it is empirically proven.
- For pooling layer, PCAPooL does not use simple pooling operation. Instead, it uses the spatial pyramid pooling which exploits the hierarchical information and makes the features rich and discriminative.
- The PCAPool performs better than the state-of-the-art methods on five challenging benchmark face recognition datasets.

The organization of the paper is as follows. Section 2 describes the proposed method. The detail of the experiments conducted for parameter selection is discussed in Sect. 3, while the results of experiments on five databases and discussion are elaborated in Sect. 4. Section 5 concludes the paper and highlights the future research directions.

2 Unsupervised feature learning: PCAPooL

The architecture of the proposed unsupervised feature learning technique is shown in Fig. 1. It consists of three layers: convolutional layer, nonlinear layer, and pooling layer. The effect of the convolutional layer is to decorrelate the image features into different feature maps. Nonlinear layer introduces nonlinearity and extracts the micropatterns from each feature map of the convolutional layer. Pooling layer reveals the hierarchy of the features. The detail of each layer is given in the following subsections. Our main objective in PCAPooL is to introduce a simple architecture that mimics the CNN architecture but with less complexity that needs a

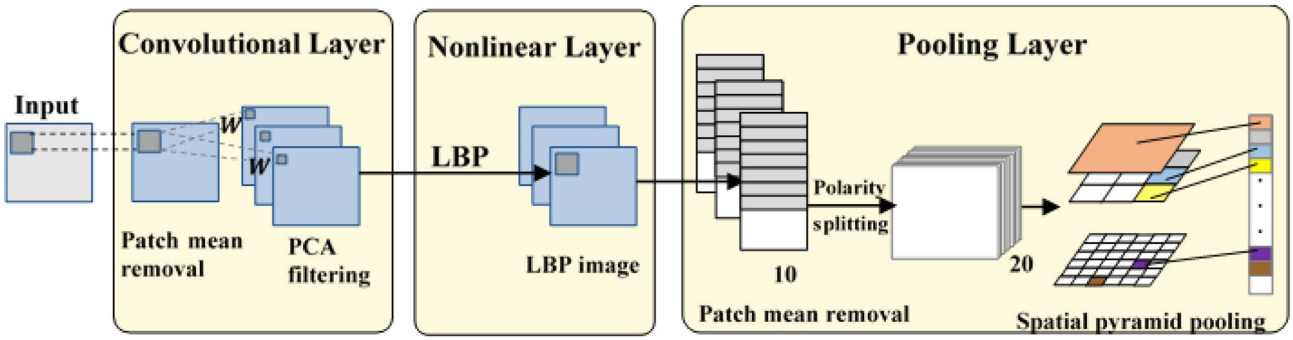

Fig. 1 Architecture of the proposed unsupervised feature earning method

small amount of data for its learning because usually in face recognition the huge volume of data is not available.

2.1 PCA convolutional layer

This layer consists of L feature maps, which are computed by convolving the input image with L filters $\boldsymbol{W}_l$ ($l=1, 2, \ldots, L$) calculated in an unsupervised manner using PCA. For calculating filters $\boldsymbol{W}_l$, PACPool adopted the approach used in PCANet [18]. An overview of this approach is given below.

Assume that N training images $\{\boldsymbol{I}_i\}_{i=1}^{N}$ (each of size $m \times n$) are given and the size of each filter $\boldsymbol{W}_l$ is $k_1 \times k_2$. Around each pixel of $\boldsymbol{I}_i$, extract a patch of size $k_1 \times k_2$ and vectorize it to $\boldsymbol{x}_{i,j}$ and collect vectors corresponding to all (overlapping) patches of $\boldsymbol{I}_i$, i.e.$\boldsymbol{x}_{i,1}, \boldsymbol{x}_{i,2}, \ldots, \boldsymbol{x}_{i,\check{m}\check{n}} \in \mathbb{R}^{\boldsymbol{k_1 k_2}}$, where $\check{m} = m - \lceil k_1/2 \rceil$, $\check{n} = n - \lceil k_2/2 \rceil$. Then subtract the mean vector from each vector and obtain $\bar{\boldsymbol{X}}_i = [\bar{\boldsymbol{x}}_{i,1}, \bar{\boldsymbol{x}}_{i,2}, \ldots, \bar{\boldsymbol{x}}_{i\check{m}\check{n}}]$, where $\bar{\boldsymbol{x}}_{i,j}$ is mean subtracted vector.

Similarly, compute $\bar{\boldsymbol{X}}_i$ for each training image and combine them to form:

$$\boldsymbol{X} = [\bar{\boldsymbol{X}}_1, \bar{\boldsymbol{X}}_2, \ldots, \bar{\boldsymbol{X}}_N] \in \mathbb{R}^{k_1 k_2 \times N\check{m}\check{n}} \tag{1}$$

The L filters $\boldsymbol{W}_l$ for the convolutional layer are calculated using PCA, which solves the following optimization problem:

$$\min_{\boldsymbol{V} \in \mathbb{R}^{k_1 k_2 \times L}} \left\| \boldsymbol{X} - \boldsymbol{V}\boldsymbol{V}^{\boldsymbol{T}}\boldsymbol{X} \right\|_F^2, \quad \text{s.t.}\ \boldsymbol{V}\boldsymbol{V}^{\boldsymbol{T}} = \boldsymbol{I}_L, \tag{2}$$

where $\boldsymbol{V} \in \mathbb{R}^{k_1 k_2 \times L}$ is a matrix of size $k_1 k_2 \times L$ and $\boldsymbol{I}_L$ is the identity matrix of size $L \times L$. The solution of this problem consists of L eigenvectors $\boldsymbol{q}_l$ of the covariance matrix $\boldsymbol{X}\boldsymbol{X}^{\boldsymbol{T}}$. The filters $\boldsymbol{W}_l$ are obtained by converting $\boldsymbol{q}_l$ into 2D filters as follows:

$$\boldsymbol{W}_l = \text{mat}_{k_1,k_2}\left(\boldsymbol{q}_l\left(\boldsymbol{X}\boldsymbol{X}^{\boldsymbol{T}}\right)\right) \in \mathbb{R}^{k_1 \times k_2}, \quad l = 1, 2, \ldots, L, \tag{3}$$

where $\text{mat}_{k_1,k_2}(\boldsymbol{v})$ is a mapping that maps $\boldsymbol{v} \in \boldsymbol{R}^{k_1 k_2}$ to a matrix $\boldsymbol{W} \in \boldsymbol{R}^{k_1 \times k_2}$ and $\boldsymbol{q}_l\left(\boldsymbol{X}\boldsymbol{X}^{\boldsymbol{T}}\right)$ represents the lth principal eigenvector of $\boldsymbol{X}\boldsymbol{X}^{\boldsymbol{T}}$. The largest principle eigenvectors encode the major variation of all the mean-removed training patches.

Finally, an input image $\boldsymbol{I}$ is convolved with $\boldsymbol{W}_l$ filters to get the convolutional layer

$$\boldsymbol{I}_l = \boldsymbol{I} * \boldsymbol{W}_l, \quad l = 1, 2, \ldots, L, \tag{3}$$

where * represents 2D convolution, and the image $\boldsymbol{I}$ is zero-padded before convolution so that $\boldsymbol{I}_l$ have the same size as $\boldsymbol{I}$. Figure 2 shows six filters and the corresponding feature maps in the convolutional layer.

2.2 Nonlinear local binary pattern (LBP) layer

This layer introduces nonlinearity. Instead of using activation functions like sigmoid or ReLU, we employ LBP operator [19, 20]. LBP is a well-known nonlinear operator that is commonly used for texture description, so LBP operator not only introduces nonlinearity but also extract micro-texture patterns and encodes the local structure, and thus it renders the features to be more discriminative. The effectiveness of LBP operator is discussed in Sect. 4.6. The feature maps of convolution layer are processed with LBP operator.

LBP is a local feature that captures various types of textures. Figure 3 shows the computation process of the LBP code of a pixel.

The simple LBP code, denoted by $\text{LBP}_{P,R}$, is calculated as follows:

$$\text{LBP}_{P,R} = \sum_{i=1}^{p-1} S(p_i - p_c)2^i \tag{4}$$

where P denotes the number of pixels in the neighborhood of the central pixel p_c and R is the radius of the neighborhood and $S()$ is the thresholding function, which is defined as follows,

$$S(p_i - p_c) = \begin{cases} 1 & p_i - p_c \geq 0 \\ 0 & p_i - p_c < 0 \end{cases} \tag{5}$$

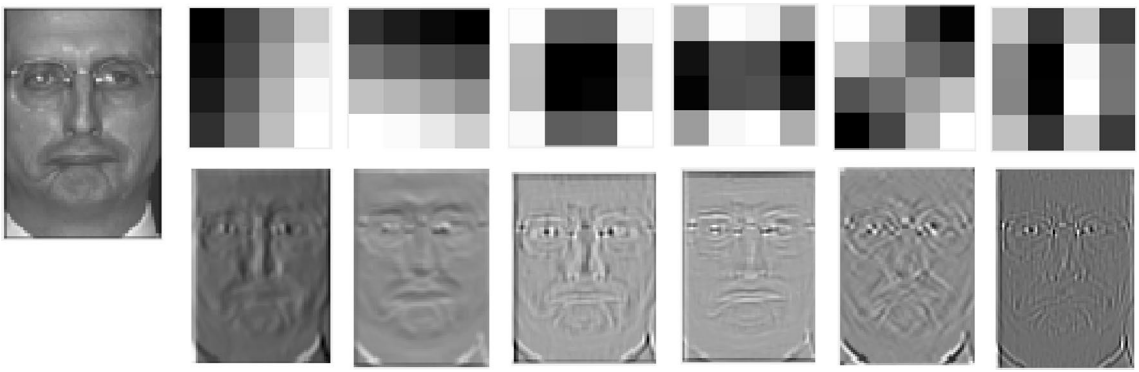

Fig. 2 A face image from FERET database, 6 filters (top row) and the corresponding feature maps (bottom row) in the convolutional layer

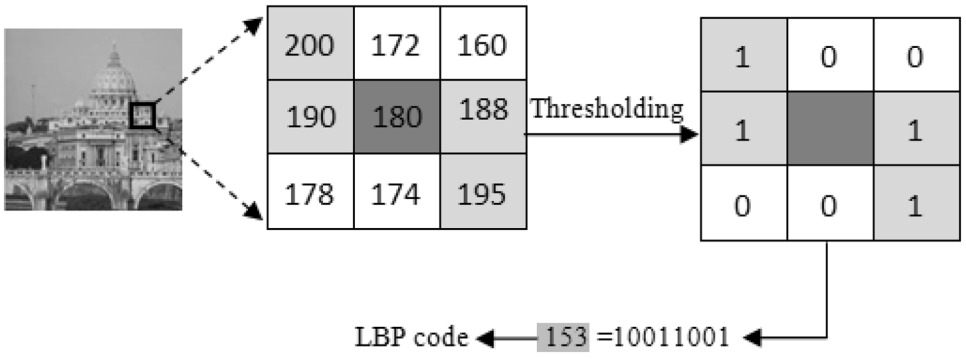

Fig. 3 LBP code computation process

After applying LBP operator on each feature map of the convolutional layer, we get nonlinear layer.

2.3 Spatial pyramid pooling layer

There is a hierarchy among different facial features, and the features extracted keeping in view this hierarchy are very much effective in discriminating face images [12]. To incorporate the hierarchy among extracted features, we adopt spatial pyramid pooling approach proposed in Ref. [12]. On each feature map of the nonlinear layer, a multi-level spatial pyramid pooling is applied, and the pooled features extracted from each of the L feature maps are concatenated to form the final feature vector.

From each feature map in nonlinear layer, pooled features are extracted as follows, and the detail is given in Ref. [12]. First, the response of each training image is calculated for the LBP plane, then mean-removed patches are extracted from all these images, and PCA is employed for reducing the dimension of these patches. This step does the important task of reducing the dimension of the final pooled features. After that, polarity splitting is applied on the normalized patches, i.e., each patch $\mathbf{x}$ is split into positive and negative components using $\max\{\mathbf{x}, 0\}$ and $\max\{-\mathbf{x},0\}$, which are finally concatenated.

Finally, the patches are spatially pooled using max pooling operation. If N levels are used, then pooling pyramid is set to $\{ l_1, l_2, .., l_N \}$ and the number of cells is $l_i \times l_i = l_i^2$ at ith level and the total number of pooling cells is $\sum_{n=1}^{N} l_n^2$. As the processed patches has very low dimension, it is possible to apply a large number of pooling levels, which is a key factor to attain higher performance (see Sect. 3.4).

2.4 Ridge regression classifier

For classification, we used a simple and efficient method based on ridge regression; this method was employed in [12, 21]. It is computationally efficient. The training of ridge regression based classifier is faster than support vector machine (SVM) because it has a closed-form solution. Despite its simplicity and efficiency, its performance is comparable with linear SVM [10]. The description of a face by PCAPool is quite high dimensional, and high-dimensional descriptions are likely to be linearly separable, and so ridge regression gives good result for face recognition.

3 Parameter tuning

Different layers of PCAPool involve four hyper-parameters (filter size, number of filters, neighborhood of LBP, and levels in pyramid pooling), and tuning of these parameters is essential for the best performance. In this section, we test the effect of these hyper-parameters. The values of the hyper-parameters that give the highest recognition performance are chosen for the next experiments. For this purpose, all the experiments are conducted on DupII, a challenging subset of FERET database. The description of this database is provided in Sect. 4.1.

3.1 Filter size for convolutional layer

The filter size for convolutional layer depends on the size of a local patch. We conducted experiments with local patches of four sizes: 3×3, 4×4, 6×6, and 8×8 pixels. We selected the top 10 principal eigenvectors in each case to define the filters $\boldsymbol{W}_l, 0 \le l \le 10$. It can be observed from Fig. 4a that 4×4 achieved the best performance; as filter size increases beyond 4×4, the accuracy start decreasing. The reason is that the filters with size 4×4 capture small-scale features, which are important for discrimination, whereas the filters with bigger size, for example 8×8, involves the neighborhood of size 8×8, which extract large-scale features and ignore the small detail. Further, we investigated the effect of combining different patches sizes (4×4, 5×5), (4×4, 6×6), (4×4, 6×6, 8×8), but we found the results were not improved and 4×4 provided the best performance with less computational cost. It is due to the reason that features at higher scales are taken care of by spatial pyramid pooling, which using small-scale features creates a hierarchy of multi-scale features and results in rich and discriminative description.

3.2 Number of filters

The number of filters is another hyper-parameter for convolutional layer. Considering 10 choices, we tested the effect of the number of PCA filters on the performance of PCAPooL. Figure 4b shows that using 10 or 11 principle components, which preserve about 90% of the eigenvalues energy, achieved the highest results. It can be observed that 11 PCA filters resulted in the same performance but with increased computational complexity (see Fig. 5), which scales linearly with the number of filters. In case of 11 filters, the image is convolved with 11 filters and 11 feature maps are created, and it follows that 11 filters not only increase the computational cost but also space requirements without any increase in accuracy. Consequently, for best trade-off between accuracy, time, and space complexity, we selected 10 PCA filters for onward experiments.

3.3 Neighborhood size of LBP for nonlinear layer

There are two parameters of LBP: the number of points p and radius r of the neighborhood. We examined the effect of the most widely used neighborhood which are (8, 1), (16, 2) and (24, 3) [39]. Figure 6 indicates that the highest performance was achieved by using $\text{LBP}_{8,1}$. Also, note that the accuracy decreases as the number of the neighboring pixels p and radius r are increased. The reason that higher scales do not give better performance is that when higher scales are selected, small-scale features, which are

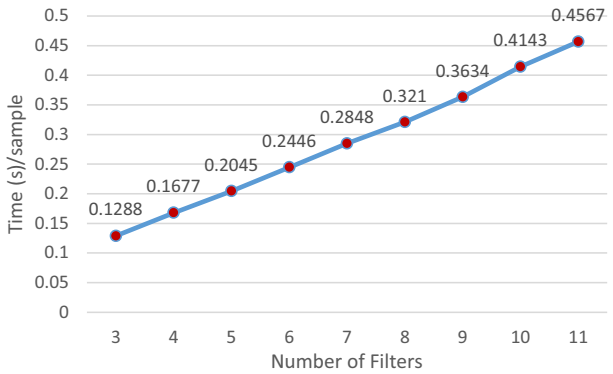

Fig. 5 Running time in seconds required for a test sample with different number of PCA filters

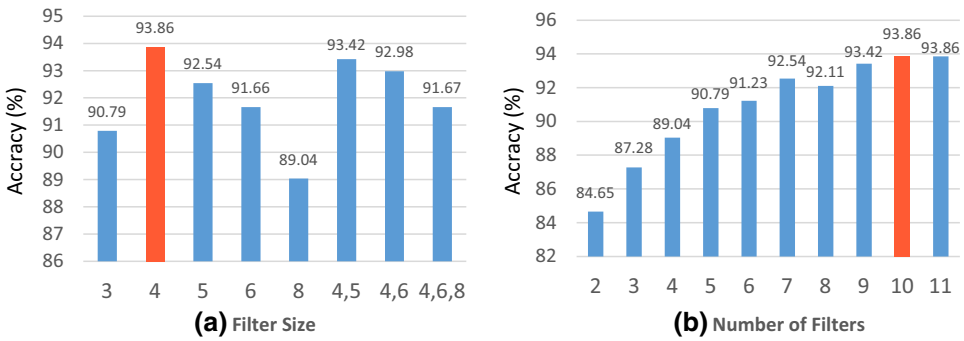

Fig. 4 Recognition accuracy (%) with varying patch sizes (**a**) and number of PCA filters (**b**) on FERET dataset (Training: Fa, Testing: DupII)

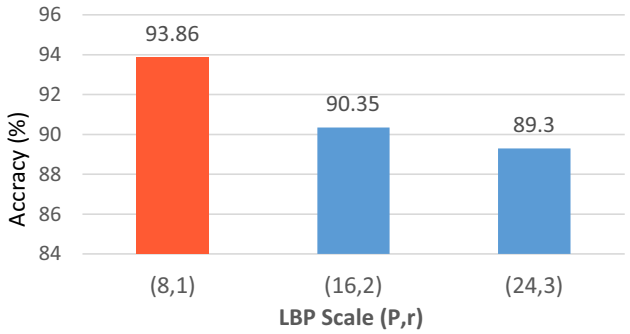

Fig. 6 Effect of LBP parameters (p, r) on the recognition accuracy (%) on FERET dataset (Training: Fa, Testing: DupII)

highly discriminative, are ignored and performance accuracy decreases. Also, the number of LBP codes becomes unmanageable because the number of LBP codes increases exponentially with p; the number of LBP codes is 2^P if (p, r) is used.

3.4 Levels of spatial pyramid for pooling layer

We examined the effect of spatial pyramids with different levels, i.e., 2-level to 8-level pyramids with regular grids of varying scales from 1×1 to 18×18, for example, an 8-level pyramid {1, 2, 4, 6, 8, 10, 12, 15} consists of regular grids of 8 scales 1×1, 2×2, …, 15×15. We tested different levels with different combinations of scales; the dimension of the final feature vector depends on this choice. Figure 7 shows that the three combinations {1, 2, 4, 6, 8, 10, 12, 15}, {1, 3, 5, 7, 10, 16} and {1, 3, 7, 10, 16} achieved the best result (i.e., 93.86%). The last combination results in less number of features, as such it has been selected as the best choice of for onward experiments.

In addition, the experiments were also performed to test the effect of using the two commonly used pooling strategies: average pooling and max pooling. The recognition accuracies obtained using different patch sizes with max and average pooling are depicted in Fig. 8. It is obvious that for all patch sizes, max pooling performed better.

3.5 Time and space complexity

PCAPool consists of three layers: convolutional layer, nonlinear layer and spatial pyramid pooling layer. If the size of the input image is $m\times n$, then the time complexity of both convolution layer and nonlinear layer is O(mn) and that of pyramid pooling layer is O($mn\log mn$) and as such the overall time complexity of PCAPool is O($mn\log mn$).

The space complexity of each of the three layers is $O(mn)$, and consequently, the space complexity of PCAPool is $O(mn)$.

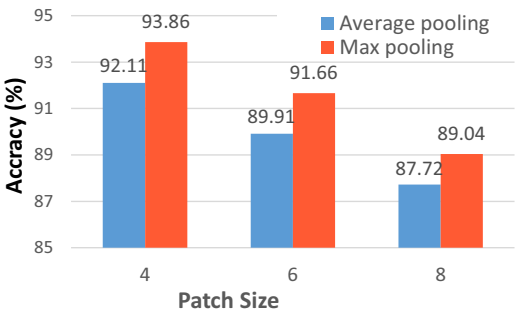

Fig. 8 Classification accuracy of using max pooling and average pooling in different patch sizes using FERET dataset (training: Fa, testing: DupII)

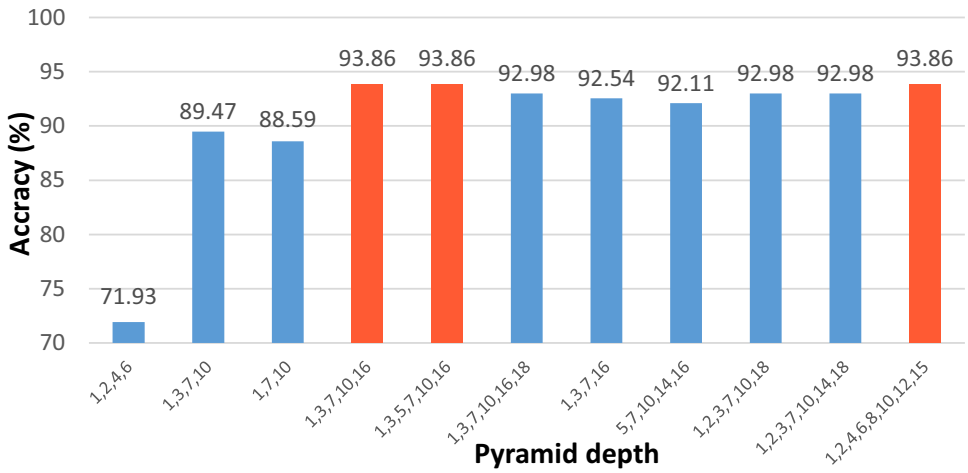

Fig. 7 Effect of different combinations of multi-level pyramid pooling using FERET dataset (training: Fa, testing: DupII)

Table 1 Recognition accuracies (%) on FERET database

Method	Accuracy (%)		
	Fb (expression)	DupI (aging)	DupII (aging)
LBP [5]	93.00	61.00	50.00
HGPP [26]	97.50	79.5	77.8
LLC [27]	99.40	82.3	77.8
POEM [28]	99.60	88.80	85.00
DMMA [29]	98.10	81.60	83.20
Tan's method [30]	98.0	90.0	85.0
Shen's method [12]	99.70	89.10	85.50
PCANet-1 [18]	99.50	89.89	86.75
PCANet-2 [18]	99.58	95.43	94.02
PCAPooL	99.70	94.71	93.86

4 Experimental results and discussion

This section presents the detailed results and discussion on these results. The performance is evaluated on five benchmark databases: Yale face database [1], FERET face database [22], AR database [23], Extended Yale Face database B [24], and multi-PIE database [25]. Also the effectiveness of PCAPool has been demonstrated. For each dataset an evaluation protocol is provided, where training/testing datasets are fixed and the research community uses this protocol for evaluation and comparison. In our experiments, we also used the same evaluation protocols.

According to the evaluation and discussion on the choice of tunable parameters in the previous section, the results presented in this section have been obtained with experiments using 10 PCA filters with 4x4 size for convolutional layer, $LBP_{8,1}$ for nonlinear layer and max pooling with 5-level pooling pyramid {1, 3, 7, 10, 16} for pooling layer. In pooling layer, we used the same patch size as was used in PCA filters (i.e., 4×4), and all patches are transformed to 10D vectors using PCA.

4.1 Experiments on FERET

Currently, the single-sample-per-person (SSPP) face recognition problem is an active research area due to its significance for real-world applications and challenges. Here, we show the results of PCAPool on FERET dataset to reveal its ability to handle SSPP problem.

FERET dataset [22] is a widely employed benchmark face recognition database. For the FERET dataset, the images were captured in multiple sessions during several years; these images have complex intra-class variability. Its subset Fa, which contains one image of 1196 subjects each, is used as gallery set, whereas Fb (having expression variations) and DupI and DupII (having age variations) are used as probe subsets. Each image in the dataset is cropped based on the locations of eyes.

From Table 1, we can see that our approach achieved the highest accuracies over the state-of-the-art results after PCANet-2. Comparing with [12], which used spatial pyramid pooling, PCAPooL achieved an increase of 8.36% and 5.61% on DupII and DupI, respectively. In comparison with PCANet, the results of PCAPooL outperformed the single-stage PCANet (PCANet-1) by 7.11% and 4.82% on DupII and DupI, respectively, and achieved almost similar results to PCANet-2 (2-stages) on DupII and DupI but higher results on Fb. The results also indicate the success of PCAPooL to overcome SSPP problem.

4.2 Experiments on AR

The AR dataset [23] was acquired from 126 subjects (70 men and 56 women) and consists of over 4000 facial images. From each subject, 26 facial images were captured in two different sessions. The captured images involve variations of facial expressions (neutral, smile, anger, and scream) and illuminations (left-light on, right-light on and all side-lights on) and occlusion by sunglasses and scarves. Out of 126 subjects, 100 subjects were selected for our experiments (50 males and 50 females). The results of PCAPooL and other recent methods are given in Table 2, which clearly show the superiority of our method.

4.3 Experiments on Yale

The Yale database [1] contains grayscale images captured from 15 subjects. The images exhibit variations in facial expression (normal, happy, sad, sleepy, surprised, and wink), illumination (left-light, center-light, right-light), and facial accessories (with/without glasses). For our experiments, a random subset with $P = (2, 3, 4, 5, 6, 7, 8)$ images per subject was selected as gallery set, and the remaining images were used to form the probe set. For each P, we averaged the results over 50 random splits. The results are presented in Table 3, which show a significant increase in the performance accuracy of PCAPooL on all training sizes compared with other existing methods.

Table 2 Recognition accuracies (%) on AR database

Method	Accuracy (%)
L2 [31]	95.90
SSRC [8]	98.90
Volterra [32]	87.50
Shen's method [12]	98.70
PCAPooL	99.62

Table 3 Recognition accuracies (%) on Yale database

Method	Accuracy (%)						
	2 Train	3 Train	4 Train	5 Train	6 Train	7 Train	8 Train
Eigenfaces [33]	56.5	51.1	47.8	45.2	–	–	–
Fisherfaces [1]	54.3	35.5	27.3	22.5	–	–	–
Laplacianfaces [34]	43.5	31.5	25.4	21.7	–	–	–
O-Laplacianfaces [35]	44.3	29.9	22.7	17.9	–	–	–
PCAPooL	91.38	95.83	97.35	98.36	98.67	98.63	98.93

Table 4 Recognition accuracy (%) of PCAPooL on Extended Yale B database

Train	20	30	40	50	Average
Accuracy (%)	99.73	99.76	99.70	99.63	99.70

Table 5 Recognition accuracies (%) of recent methods on Extended Yale B database

Method	Accuracy (%)
LBP [5]	75.76
P-LBP [36]	96.13
PCANet-1 [18]	97.77
PCANet-2 [18]	99.58
PCAPooL	99.70

4.4 Experiments on Extended Yale B

The Extended Yale B dataset [24] was acquired from 38 subjects and consists of 2414 frontal-face images. Under various laboratory-controlled lighting conditions, 64 images were captured from each subject. The images were cropped and normalized to 192×168 pixels, whereas the images were resized to 32×32 pixels for our experiments.

A random subset with P = (20, 30, 40, 50) images per subject was selected as gallery set, whereas the remaining images were used to form the probe set. For each given P, we averaged the results over 10 random splits. The recognition accuracy of PCAPooL on different training set sizes is shown in Table 4. It can be seen that the PCAPooL is consistent in performance and is not affected by the size of training examples. A comparison with recent methods in Table 5 shows that PCAPooL achieved the best result.

4.5 Experiments on multi-PIE

The PIE database [25] was acquired at CMU to support the face recognition research across illumination and pose variations. The PIE database has some drawbacks. It was captured from 68 subjects only in a single session and images exhibit small variation in expressions (neutral, smile, blink, and talk). To overcome these shortcomings, the multi-PIE database was acquired from a larger number of subjects (337) in up to four sessions. It was captured using the improved recording environment. The training set consists of 1855 images, while the testing has 4244 images; we used setting similar to the one that was used in [38]. Table 6 shows that PCAPooL significantly outperformed other recent techniques.

Table 6 Recognition accuracies (%) on multi-PIE database

Method	Accuracy (%)
Method in [3]	66.43
RSC [37]	64.85
Improved RSC by [38]	71.26
Methods in [38]	77.43
PCAPooL	82.94

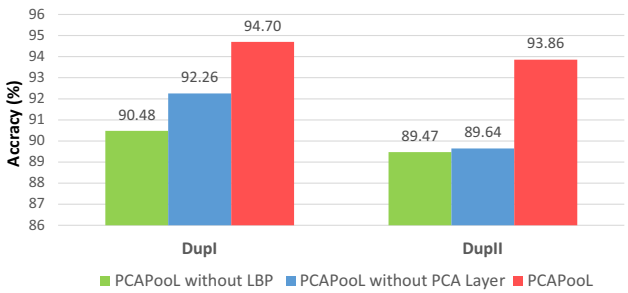

Fig. 9 Effectiveness of nonlinear LBP and convolutional layers

4.6 Effectiveness of nonlinear LBP layer

The question is whether nonlinear LBP layer has impact on the recognition performance of PCAPool. To answer this question, we implemented the method without nonlinear LBP layer. The results of PCAPooL with and without LBP layer on DupI and DupII sets of FERET database are depicted in Fig. 9. It can be observed that LBP layer increases the accuracy by 4.22%, and 4.39% on DupI and DupII, respectively.

Moreover, we investigated the effect of LBP on the method proposed in Ref. [12], and instead of using the original image, we gave LBP image as input. Figure 10 shows

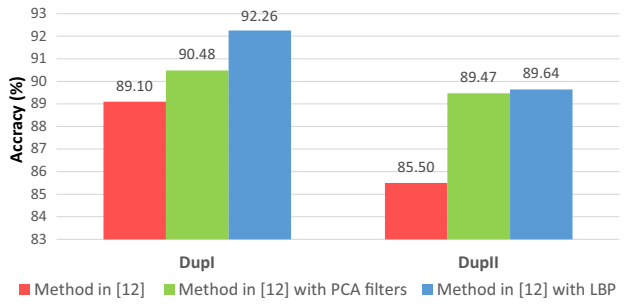

Fig. 10 Effectiveness of LBP and PCA convolutional layers in Shen's method [12]

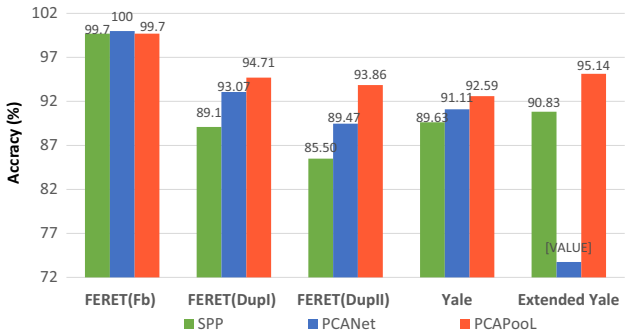

Fig. 11 Effectiveness of PCAPooL in comparison with PCANet and Shen's method [12]

that the introduction of LBP enhanced the results by 3.16% and 4.14% on DupI and DupII, respectively.

4.7 Effectiveness of PCA convolution layer

Another question arises about the effectiveness of PCA convolutional layer. To address this question, we tested the method with and without PCA convolutional layer (same as applying LBP in Shen's method [12]). It can be observed from Fig. 9 that PCA convolutional layer increases the accuracy by 2.44% and 4.22% on DupI and DupII, respectively.

Comparing the effect of PCA layer with that of LBP Layer (see Figs. 9 and 10), we observe that LBP has stronger effect.

4.8 Comparative evaluation of PCAPooL

In this section, we evaluate PCAPooL in comparison with PCANet and Shen's method [12]. To ensure fairness in comparison, for both methods, the features were extracted using the optimal configurations as reported in their original works. Moreover, all the experiments were performed using the same classifier (i.e., ridge regression classifier).

Experiments conducted on FERET, Yale, and Extended Yale B datasets demonstrate the effectiveness of PCAPool over PCANet and Shen's method [12], which can be clearly noticed from Fig. 11. Particularly, PCAPooL outperformed PCANet by 1.64%, 4.39%, 1.48%, and 21.4% on DupI, DupII, Yale, and Extended Yale B, respectively. In comparison with Shen's method [12], PCAPooL achieved higher performance by 5.61%, 8.36%, 2.96%, and 4.31% on DupI, DupII, Yale, and Extended Yale B, respectively.

5 Conclusion

In this paper, we have introduced a novel unsupervised feature learning method—PCAPooL—for face recognition. It consists of three layers: convolutional layer, nonlinear layer, and pooling layer. PCA is used to learn filters for the convolution layer. For nonlinear layer, LBP operator is employed on the feature maps of the activation of the convolutional layer. Then, using multi-level spatial pyramid pooling, discriminative features from feature maps of the nonlinear layer are extracted. A thorough evaluation reveals that despite the simplicity, PCAPooL performs better than many state-of-the-art methods on five benchmark databases: FERET, AR, multi-PIE, Yale, and Extended Yale B. PCAPool handles SSPP problem, variations in pose and illumination, and occlusions in a nice way. Our next plane is to apply PCAPooL on other recognition tasks such as image retrieval and object classification.

Acknowledgements The authors are thankful to the Deanship of Scientific Research, King Saud University, Riyadh, Saudi Arabia for funding this work through the Research Group No. RGP-1439-067.

References

1. 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
2. Turk MA, Pentland AP (1991) Face recognition using eigenfaces. In IEEE Computer Society conference on computer vision and pattern recognition, 1991. Proceedings CVPR'91, pp 586–591
3. Liu C, Wechsler H (2002) Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition. IEEE Trans Image Process Publ IEEE Signal Process Soc 11(4):467–476
4. Lei Z, Li SZ, Chu R, Zhu X (2007) Face recognition with local gabor textons. In: Lee S-W, Li SZ (eds) Advances in biometrics: international conference, ICB 2007, Seoul, Korea, August 27–29, 2007. Proceedings. Springer, Berlin, pp 49–57
5. Ahonen T, Hadid A, Pietikäinen M (2006) Face description with local binary patterns: application to face recognition. IEEE Trans Pattern Anal Mach Intell 28(12):2037–2041
6. Wright J, Yang A, Ganesh A, Sastry S, Ma Y (2009) Robust face recognition via sparse representation. IEEE Trans Pattern Anal Mach Intell 31(2):210–227

7. Wright J, Ma Y, Mairal J, Sapiro G, Huang TS, Yan S (2010) Sparse representation for computer vision and pattern recognition. Proc IEEE 98(6):1031–1044
8. Deng W, Hu J, Guo J (2013) In defense of sparsity based face recognition. In: 2013 IEEE conference on computer vision and pattern recognition (CVPR), pp 399–406
9. Yang M, Zhang L, Shiu SCK, Zhang D (2013) Gabor feature based robust representation and classification for face recognition with Gabor occlusion dictionary. Pattern Recognit 46(7):1865–1878
10. Lazebnik S, Schmid C, Ponce J (2006) Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2169–2178
11. Yang J, Yu K, Gong Y, Huang T (2009) Linear spatial pyramid matching using sparse coding for image classification. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1794–1801
12. Shen F, Shen C, Zhou X, Yang Y, Shen HT (2016) Face image classification by pooling raw features. Pattern Recognit 54:94–103
13. Jarrett K, Kavukcuoglu K, Ranzato M, LeCun Y (2009) What is the best multi-stage architecture for object recognition? In: 2009 IEEE 12th international conference on computer vision, pp 2146–2153
14. Kavukcuoglu K, Sermanet P, Boureau Y, Gregor K, Mathieu M, Cun YL (2010) Learning convolutional feature hierarchies for visual recognition. Presented at the Advances in Neural Information Processing Systems, 2010, pp 1090–1098
15. Zeiler MD, Fergus R (2014) Visualizing and understanding convolutional networks. In: Fleet D, Pajdla T, Schiele B, Tuytelaars T (eds) Computer vision—ECCV 2014. Springer, Berlin, pp 818–833
16. Zhu Z, Luo P, Wang X, Tang X (2013) Deep learning identity-preserving face space. In: Proceedings of the 2013 IEEE international conference on computer vision, Washington, DC, USA, pp 113–120
17. Sun Y, Wang X, Tang X (2014) Deep learning face representation from predicting 10,000 classes. In: Proceedings of the 2014 IEEE conference on computer vision and pattern recognition, Washington, DC, USA, pp 1891–1898
18. Chan TH, Jia K, Gao S, Lu J, Zeng Z, Ma Y (2015) PCANet: a simple deep learning baseline for image classification? IEEE Trans Image Process 24(12):5017–5032
19. Zhang G, Huang X, Li SZ, Wang Y, Wu X (2004) Boosting local binary pattern (LBP)-based face recognition. In: Li SZ, Lai J, Tan T, Feng G, Wang Y (eds) Advances in biometric person authentication. Springer, Berlin, pp 179–186
20. Jabid T, Kabir MH, Chae O (2010) Gender classification using local directional pattern (LDP). In: Proceedings of the 2010 20th international conference on pattern recognition, Washington, DC, USA, pp 2162–2165
21. Gong Y, Lazebnik S (2011) Comparing data-dependent and data-independent embeddings for classification and ranking of internet images. In: Proceedings of the 2011 IEEE conference on computer vision and pattern recognition, Washington, DC, USA, pp 2633–2640
22. Phillips PJ, Wechsler H, Huang J, Rauss PJ (1998) The FERET database and evaluation procedure for face-recognition algorithms. Image Vis Comput 16(5):295–306
23. Martinez A, Benavente R. The AR face database. CVC, Technical Report
24. Georghiades AS, Belhumeur PN, Kriegman DJ (2001) From few to many: illumination cone models for face recognition under variable lighting and pose. IEEE TPAMI 23(6):643–660
25. Gross R, Matthews I, Cohn J, Kanade T, Baker S (2008) Multi-PIE. In: 8th IEEE international conference on automatic face gesture recognition, 2008. FG'08, pp 1–8
26. Zhang B, Shan S, Chen X, Gao W (2007) Histogram of Gabor phase patterns (HGPP): a novel object representation approach for face recognition. IEEE Trans. Image Process. 16(1):57–68
27. Wang J, Yang J, Yu K, Lv F, Huang T, Gong Y (2010) Locality-constrained linear coding for image classification. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3360–3367
28. Vu N-S, Caplier A (2012) Enhanced patterns of oriented edge magnitudes for face recognition and image matching. IEEE TIP 21(3):1352–1368
29. Lu J, Tan Y-P, Wang G (2013) Discriminative multi-manifold analysis for face recognition from a single training sample per person. IEEE TPAMI 35(1):39–51
30. Tan X, Triggs B (2007) Fusing Gabor and LBP feature sets for kernel-based face recognition. In: International workshop on analysis and modeling of faces and gestures, pp 235–249
31. Shi Q, Eriksson A, vanden Hengel A, Shen C (2011) Is face recognition really a compressive sensing problem? In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 553–560
32. Kumar R, Banerjee A, Vemuri BC, Pfister H (2012) Trainable convolution filters and their application to face recognition. IEEE Trans Pattern Anal Mach Intell 34(7):1423–1436
33. Turk M, Pentland A (1991) Eigenfaces for recognition. J Cogn Neurosci 3(1):71–86
34. He X, Yan S, Hu Y, Niyogi P, Zhang H-J (2005) Face recognition using laplacianfaces. IEEE Trans Pattern Anal Mach Intell 27(3):328–340
35. Cai D, He X, Han J, Zhang H-J (2006) Orthogonal Laplacian faces for face recognition. IEEE Trans Image Process 15(11):3608–3614
36. Tan X, Triggs B (2010) Enhanced local texture feature sets for face recognition under difficult lighting conditions. IEEE TIP 19(6):1635–1650
37. Yang M, Zhang L, Yang J, Zhang (2011) Robust sparse coding for face recognition. In: Proceedings of IEEE international conference on computer vision and pattern recognition, pp 625–632
38. Tao Q-C, Liu Z-M, Bebis G, Hussain M (2015) Face recognition using a novel image representation scheme and multi-scale local features. Int J Biom 7(3):191–212
39. Ojala T (2002) Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Trans Pattern Anal Mach Intell 24(7):971–987