Pattern Recognition 79 (2018) 279–289

Contents lists available at ScienceDirect

Pattern Recognition

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

Self-learning for face clustering

Xiaoshuang Shi[a], Zhenhua Guo[b], Fuyong Xing[c], Jinzheng Cai[a], Lin Yang[a,*]

[a] J. Crayton Pruitt Family Department of Biomedical Engineering, University of Florida, United States
[b] Graduate School at Shenzhen, Tsinghua University, China
[c] Department of Biostatistics and Informatics, University of Colorado Denver, United States

ARTICLE INFO

Article history:
Received 23 October 2017
Revised 27 January 2018
Accepted 10 February 2018
Available online 15 February 2018

Keywords:
Face clustering
Patch-based two-dimensional reconstruction
Self-paced learning

ABSTRACT

In this paper, we simulate the learning way of human to propose a self-learning framework for face clustering. Specifically, we first perform a decorrelation operation on face images through patch-based two-dimensional reconstruction, which has a similar function to the retina. Then we group the semantically similar faces by using a novel self-paced learning model, which is inspired by three major observations: (i) The learning process of human gradually proceeds from easy to complex tasks; (ii) The prior knowledge of human might change with the increase of learned experience; (iii) More prior knowledge usually leads to better prediction accuracy. Experiments on benchmark face databases demonstrate the effectiveness and efficiency of the proposed framework.

1. Introduction

Face recognition is a key research topic in machine learning, pattern recognition and computer vision areas because of its potential value for practical applications and theoretical challenges [1,2]. It has attracted considerable attention over the past two decades, and many unsupervised [3,4] and supervised methods [5–11] have been proposed for face recognition. Generally, supervised learning methods require numerous semantic labels to extract good high-level features [5,6]. Nevertheless, manually labeling data is time consuming and expensive. In addition, during the process of recording, face images usually encounter some kinds of noise, like lighting variability [12] and occlusions [13]. Although the visual effect of images is different, essentially the number of dimensions is small [14]. Unsupervised learning aims to explore the intrinsic low-dimensional features [15–18] or structures with removing the noise effect [4] and thus might improve the performance of recognition algorithms. To make better use of the unlabeled data and explore the inherent characteristics of face images, in this paper, we focus on the problem of unsupervised learning.

When human eyes see an image, the retina in eyes usually performs a decorrelation operation on the image to reduce its redundancy [19,20] before image recognition. Inspired by this observation, many image decorrelation methods are proposed to reduce redundancy [3,21,22] in order to boost the accuracy of face recognition [23]. Recently, two-dimensional whitening reconstruction (TWR) [24] has been proposed to reduce the redundancy of internal face images. It preserves the significant intrinsic features and approximates each generated whitening face as a Gaussian signal for face recognition. However, *TWR can reduce the global redundancy and preserve global intrinsic features of a 2D face image matrix*. A considerable amount of literature [1,25–27] has demonstrated the significance of local features on face recognition. It suggests that reducing the redundancy of local pixels and preserving local important intrinsic features might be more conductive to face recognition. Moreover, the method in [24] does not quantitatively analyze the Gaussian distribution of the whitening face, although the whitening face has been shown to be close to a Gaussian signal.

After reducing the redundancy of face images, it is important to extract the significant and useful knowledge from the preprocessed image for recognition. Recent literature [28–31] illustrates that learning knowledge from samples in an organized and meaningful order often performs better than using randomly selected samples. Based on the learning way of human who learns knowledge gradually from easy to complex samples [32], curriculum learning (CL) [28] learns a model by gradually increasing the difficulty level of samples so that the entropy of training samples is increased. CL and its variants have achieved promising performance on many applications, such as action/event detection [29], dictionary learning [33] and tracking [34]. Since the curriculum in CL [28] is predetermined by prior knowledge and cannot be adjusted accordingly, it might generate the inconsistency between predetermined knowledge and dynamically learned models. To address this problem, self-paced learning (SPL) [35], has been proposed to dy-

* Corresponding author.
E-mail address: lin.yang@bme.ufl.edu (L. Yang).

https://doi.org/10.1016/j.patcog.2018.02.008
0031-3203/

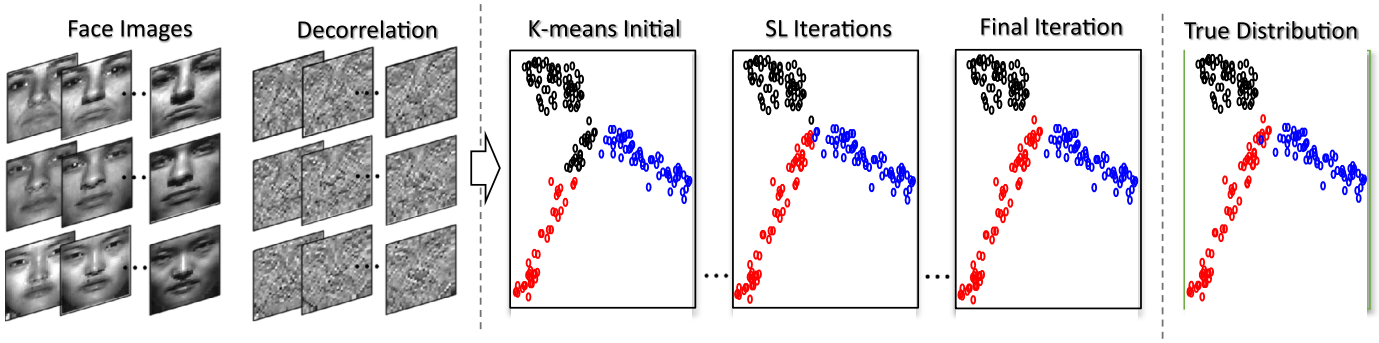


Fig. 1. The flowchart of face clustering by self-learning (SL iterations denote the self-paced learning process).

namically determine the curriculum that can adapt to the learning pace of the learner. Furthermore, self-paced curriculum learning (SPCL) [30], has been presented to incorporate the prior knowledge before training as well as the paced learning process during training to avoid overfitting. However, both CL and SPCL are suitable for supervised learning scenarios, since they assume that the prior knowledge is fixed during the learning process. To learn curriculum in the unsupervised process, prior knowledge might be flexible, because in practice the prior views of human sometimes change based on important scientific discoveries [36,37]. It implies that *the human prior knowledge might change with the increase of learned knowledge*. In addition, *more prior knowledge often leads to better prediction*, which has been demonstrated by numerous supervised learning methods [5,7,8,38].

Based on the aforementioned observations, in this paper we propose a novel self-learning framework (please see Fig. 1), which consists of two major stages: image decorrelation and self-paced learning, for face clustering without using any label information. Our main contributions are listed as follows:

- We extend TWR to handle local image patches in order to reduce image redundancy while preserve local significant features, and provide a quantitative analysis on the Gaussian distribution of the whitening face and its patch-based version.
- We propose a novel self-paced learning model for face clustering, taking into consideration the three observations about the process of human learning: (i) The learning process of human and animals gradually proceeds from easy to complex tasks; (ii) The prior knowledge might be revised and become more accurate with the growth of learned knowledge; (iii) More prior knowledge usually produces better accurate prediction results.
- Experiments on three benchmark databases illustrate the effectiveness and efficiency of the patch-based TWR and the self-paced learning model on faces under various environments.

The rest of this paper is structured as follows. Section 2 briefly reviews the related work. Section 3 presents patch-based TWR and provides a quantitative analysis on the distribution of the whitening face. Section 4 introduces a self-paced learning model and shows a self-learning framework for face clustering. Section 5 displays and analyzes experimental results on benchmark face databases. Section 6 concludes this paper and points out the future work.

2. Preliminaries

2.1. Definitions and notations

Throughout this paper, matrices and vectors are denoted as boldface uppercase and lowercase letters, respectively. For a matrix $\mathbf{X} \in \mathbb{R}^{p\times n}$, its ith row and jth column are denoted by $\mathbf{x}^i$ and $\mathbf{x}_j$, respectively, and x_j^i is its one entry at the ith row and jth column.

2.2. Two-dimensional whitening reconstruction

TWR [24] mainly consists of two stages: (i) a whitening process on a 2D face image matrix; (ii) 2D face image matrix reconstruction. Given a 2D face image $\mathbf{X} \in \mathbb{R}^{p\times n}$, where p and n denote the image height and width, respectively, TWR first calculates the mean vector $\bar{\mathbf{x}} = \frac{1}{p}\sum_{i=1}^{p}\mathbf{x}^i$ and then subtracts it from $\mathbf{X}$ to obtain a matrix $\mathbf{X}_v$. Next, $\mathbf{X}_v$ is decomposed by using singular value decomposition (SVD) to obtain $\mathbf{X}_v = \mathbf{UDV}^T$, where $\mathbf{U}^T\mathbf{U} = \mathbf{I}_p$ and $\mathbf{V}^T\mathbf{V} = \mathbf{I}_n$. Finally, the whitening face is achieved by $\tilde{\mathbf{X}} = \mathbf{U}(:, 1:m)\mathbf{V}(:, 1:m)^T$, where $m \le min(p, n)$. The detailed procedure of TWR is shown in Algorithm 1. Note that the whitening face $\tilde{\mathbf{X}}$ discards the coefficient $\sqrt{n}$ because all images in one set usually have the identical image size. Through SVD and removing eigenvalues, TWR can reduce the pixel redundancy of the internal face image and meanwhile preserve the significant global intrinsic features.

Algorithm 1 TWR [24].

Input: An image matrix $\mathbf{X} = \left[\mathbf{x}^1; \mathbf{x}^2; \cdots; \mathbf{x}^p\right] \in \mathbb{R}^{p\times n}$, the rank m.
1. Calculate the mean vector $\bar{\mathbf{x}} = \frac{1}{p}\sum_{i=1}^{p}\mathbf{x}^i$,
2. Calculate the matrix $\mathbf{X}_v = \left[\mathbf{x}^1 - \bar{\mathbf{x}}; \mathbf{x}^2 - \bar{\mathbf{x}}; \cdots; \mathbf{x}^p - \bar{\mathbf{x}}\right]$;
3. Calculate the SVD of $\mathbf{X}_v = \mathbf{UDV}^T$;
4. Calculate the matrix $\tilde{\mathbf{X}}_v = \sqrt{n}\mathbf{U}(:, 1:m)\mathbf{V}(:, 1:m)^T$.

Output: Reconstructed matrix $\tilde{\mathbf{X}}_v \in \mathbb{R}^{p\times n}$.

2.3. Self-paced Curriculum Learning

CL [28] illustrates that learning knowledge from samples in an organized and meaningful order often performs better than using randomly selected samples, while it keeps the curriculum (predetermined by prior knowledge) fixed and does not consider the feedback about learners. SPL [35] can dynamically learn the curriculum, but it does not incorporate prior knowledge into the learning process and the learning is completely dominated by the training loss. In order to dynamically determine the curriculum and meanwhile take into account the prior knowledge before training and the information learned during training, we briefly introduce SPCL in the following.

Given training data $\mathbf{Z} \in \mathbb{R}^{d\times N}$, where d and N are the number of dimensions and samples, respectively, $\mathbf{y} \in \mathbb{R}^N$ denotes their corresponding label vector. SPCL [30] proposes a formulation as follows:

$$\begin{aligned} &\min_{\mathbf{w},\mathbf{v}\in[0,1]^N} E(\mathbf{w}, \mathbf{v}; \lambda, \Phi) \\ &= \sum_{i=1}^{N} v_i L(y_i, g(\mathbf{z}_i, \mathbf{w})) + f(\mathbf{v}; \lambda), \\ &s.t.\ \mathbf{v} \in \Phi. \end{aligned} \tag{1}$$

In Eq. (1), $g(\mathbf{z}_i, \mathbf{w})$ represents the estimated label of the sample $\mathbf{z}_i$, $L(y_i, g(\mathbf{z}_i, \mathbf{w}))$ is a loss function that calculates the cost between the true label y_i and the estimated label $g(\mathbf{z}_i, \mathbf{w})$. $\mathbf{w}$ denotes model parameters inside the decision function g. For example, with a simple least square model, $L(y_i, g(\mathbf{z}_i, \mathbf{w})) = \|y_i - \mathbf{z}_i\mathbf{w}\|_2^2$, where $\mathbf{w} \in \mathbb{R}^d$ is a column vector and $g(\mathbf{z}_i, \mathbf{w}) = \mathbf{z}_i^T\mathbf{w}$ is a decision function.

$f(\mathbf{v}; \lambda)$ is the self-paced function, in which $\mathbf{v} = [v_1, v_2, \cdots, v_N]$ are weight variables reflecting the significance of each sample and λ is a parameter to control the learning pace through changing the value of $\mathbf{v}$. To set $f(\mathbf{v}; \lambda)$, one popular scheme first introduced in [35] is: $f(\mathbf{v}; \lambda) = -\lambda\|\mathbf{v}\|_1 = -\lambda\sum_{i=1}^{N} v_i$, where $v \in \{0, 1\}^N$.

Φ denotes a feasible region encoded by the predetermined curriculum, which is determined by the prior knowledge before training. In [30], Φ is defined as:

Definition 1. *Given a curriculum γ on training data $\mathbf{Z}$, $\mathbf{a} \in \mathbb{R}^N$ is a column vector and c is a constant, $\Phi = \{\mathbf{v}|\mathbf{a}^T\mathbf{v} < c\}$ is a feasible region of γ when it holds: 1) $\Phi \wedge \mathbf{v} \in [0, 1]^n$ is nonempty; 2) $a_i < a_j$ for $\gamma(\mathbf{z}_i) < \gamma(\mathbf{z}_j)$, $a_i = a_j$ for $\gamma(\mathbf{z}_i) = \gamma(\mathbf{z}_j)$.*

To solve the problem in Eq. (1), SPCL employs the same iteration strategy as SPL that divides the optimization problem into two subproblems: $\mathbf{w}$-subproblem and $\mathbf{v}$-subproblem. Firstly, fixing $\mathbf{v}$ and updating $\mathbf{w}$; secondly, fixing $\mathbf{w}$ and updating $\mathbf{v}$. In SPL, with $\mathbf{v}$ fixed, it adopts a latent structural SVM (latent SSVM) [39] to learn model parameters $\mathbf{w}$; with $\mathbf{w}$ fixed and $f(\mathbf{v}; \lambda) = -\lambda\sum_{i=1}^{N} v_i$, it updates $\mathbf{v}$ by:

$$\mathbf{v}^* = \begin{cases} 1 & L(y_i, g(\mathbf{z}_i, \mathbf{w})) < \lambda, \\ 0 & otherwise, \end{cases} \tag{2}$$

where $g(\mathbf{z}_i, \mathbf{w})$ is the decision function of the latent SSVM.

3. Patch-based two-dimensional whitening reconstruction and analysis

3.1. Patch-based TWR

TWR can reduce the global redundancy and preserve global intrinsic features of a holistic 2D face image matrix. However, local face features might perform better than global ones [1,25]. Therefore, we present a patch-based TWR (PTWR) to reduce the redundancy of local pixels and meanwhile preserve local intrinsic features.

Given an image $\mathbf{X} \in \mathbb{R}^{p\times n}$ that can be divided into a set of non-overlapped patches $\mathbf{X} = \{\mathbf{X}_1; \mathbf{X}_2; \cdots; \mathbf{X}_q\}$, where $\mathbf{X}_i \in \mathbb{R}^{h\times w}$ $(1 \le i \le q)$ and $pn = hwq$. For each patch $\mathbf{X}_i$, we apply TWR to $\mathbf{X}_i$ and then obtain the reconstructed whitening patch $\tilde{\mathbf{X}}_i$ (Similar to [24], the coefficient $\sqrt{w}$ is discarded), which can be combined to form the final reconstructed whitening face $\tilde{\mathbf{X}} = \{\tilde{\mathbf{X}}_1; \tilde{\mathbf{X}}_2; \cdots; \tilde{\mathbf{X}}_q\}$. For clarity, we present the detailed procedure of PTWR in Algorithm 2. Note that TWR can be viewed as one special case of PTWR, because when $h = p$ and $w = n$, PTWR degenerates into TWR. Since PTWR reduces the redundancy of local pixels and preserves local intrinsic features of a 2D face image, in this paper we name the face image processed by PTWR as local whitening face. Based on [24], using TWR to process each patch requires $\mathcal{O}(min(h^2w, hw^2))$ operations. Thus, the time complexity of PTWR is $\mathcal{O}(min(hpn, wpn))$, where $pn = hwq$.

Algorithm 2 PTWR.

Input: An image matrix $\mathbf{X} \in \mathbb{R}^{p\times n}$, path size $h \times w$, the patch rank r.

1. Dividing an image into a set of non-overlapped patches: $\mathbf{X} = \{\mathbf{X}_1; \mathbf{X}_2; \cdots; \mathbf{X}_q\}$.
2. For $i = 1$ to q do
   $\tilde{\mathbf{X}}_i \leftarrow \text{TWR}(\mathbf{X}_i, r)$,
   end;
3. Reconstructed image: $\tilde{\mathbf{X}} = \{\tilde{\mathbf{X}}_1; \tilde{\mathbf{X}}_2; \cdots; \tilde{\mathbf{X}}_q\}$.

Output: Reconstructed image matrix $\tilde{\mathbf{X}} \in \mathbb{R}^{p\times n}$.

To better illustrate the effect of PTWR on face representation, we present local whitening faces and their reconstructions using principle component analysis (PCA) in Fig. 2. We can see that local whitening faces have more white points, which represent large pixel values, than whitening faces, because they emphasize local texture information and preserve local significant intrinsic features.

3.2. Projection for separation

To illustrate the performance of PTWR on linear projection, we select three human faces with lighting variability from the Extended Yale-B database [40]. Fig. 3 shows an example of original, whitening and local whitening faces using PCA and PCA II [41] to project them into a 2D plane (Note that to show the superior performance of PTWR, Fig. 3 adopts images from the three individuals with labels 22, 32 and 34 rather than 7, 23 and 33, on which local whitening faces are also separable). Here, PCA is to keep the two largest eigenvalues of the data matrix to preserve the maximum variance among data points, while PCA II is to reduce the redundancy among data points by setting the two largest eigenvalues to be ones. As we can see, local whitening faces are more separable in a 2D space than original and whitening faces. This might rely on two major reasons. First, PTWR reduces local face redundancy, emphasizes local texture information and preserves the local significant intrinsic features, different with TWR that preserves the global rough profile and texture information of original faces (please refer to Fig. 2). Second, the distribution of local whitening faces can also be viewed as an approximative Gaussian (please refer to Fig. 4), with a larger variance than TWR ($\frac{rq}{hw} \ge \frac{m}{pn}$). The detailed distribution of global and local whitening faces is analyzed in Section 3.3.

3.3. Distribution analysis

Since we have observed that the distribution of global and local faces is approximate to Gaussian, in this subsection, we provide the quantitative analysis of the distribution. For simplicity, we temporarily drop the subscript of the whitening image patch and represent it as $\tilde{\mathbf{X}}$ in this subsection. Given a whitening face image $\tilde{\mathbf{X}}$ that contains pn pixels, assume that these pixels are pn statistically independent samples of a random variable $\mathbf{R}$. To calculate the mean and variance of $\mathbf{R}$, we provide two theorems as follows.

Theorem 1. *For any column vector $\mathbf{u} \in \mathbb{R}^p$ and $\mathbf{v} \in \mathbb{R}^n$, if $\mathbf{u}^T\mathbf{u} = 1$ and $\mathbf{v}^T\mathbf{v} = 1$, assume that $\mathbf{u}\mathbf{v}^T \in \mathbb{R}^{p\times n}$ contains pn statistically independent samples of a random variable $\mathbf{R}$, then its mean is $E(\mathbf{R}) = \frac{ab}{pn}$ and the variance is $D(\mathbf{R}) = \frac{1}{pn} - \frac{a^2b^2}{p^2n^2}$, where $a = \mathbf{1}_p^T\mathbf{u}$, $b = \mathbf{1}_n^T\mathbf{v}$, and $\mathbf{1}_p \in \mathbb{R}^p$ as well as $\mathbf{1}_n \in \mathbb{R}^n$ is a column vector with all elements being ones.*

Proof. First, calculating the mean with the following equation:

$$\begin{aligned} E(\mathbf{R}) &= \frac{1}{pn}\sum_{i=1}^{p}\sum_{j=1}^{n} u_i v_j \\ &= \frac{1}{pn}\mathbf{1}_p^T\mathbf{u}\mathbf{v}^T\mathbf{1}_n = \frac{1}{pn}(\mathbf{1}_p^T\mathbf{u})(\mathbf{v}^T\mathbf{1}_n) = \frac{ab}{pn}. \end{aligned} \tag{3}$$

Next, we calculate the variance of $\mathbf{R}$. According to the definition of the variance, we can get $D(\mathbf{R}) = E(\mathbf{R}^2) - (E(\mathbf{R}))^2$. $E(\mathbf{R}^2) = \frac{1}{pn}\sum_{i=1}^{p}(u_i^2\sum_{j=1}^{n} v_j^2) = \frac{1}{pn}\sum_{i=1}^{p} u_i^2 = \frac{1}{pn}$, due to $E(\mathbf{R}) = \frac{ab}{pn}$, $D(\mathbf{R}) = \frac{1}{pn} - \frac{a^2b^2}{p^2n^2}$. □

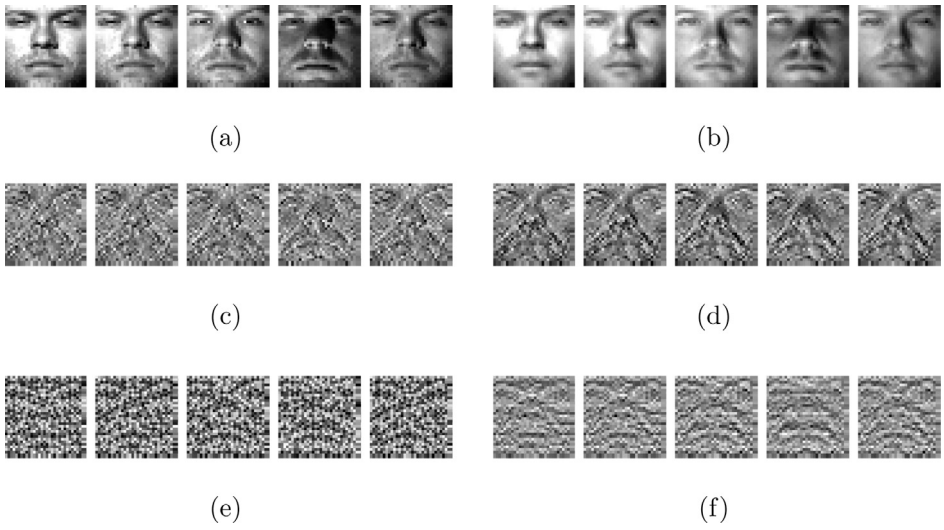


Fig. 2. Face representation with different preprocessing methods. (a)-(b) Original faces and their reconstructions by PCA, (c)-(d) whitening faces and their reconstructions by PCA, (e)-(f) local whitening faces and their reconstructions by PCA. (Original faces are from the Extended Yale-B database, and white points with large pixel values are emphasized texture information of each face.).

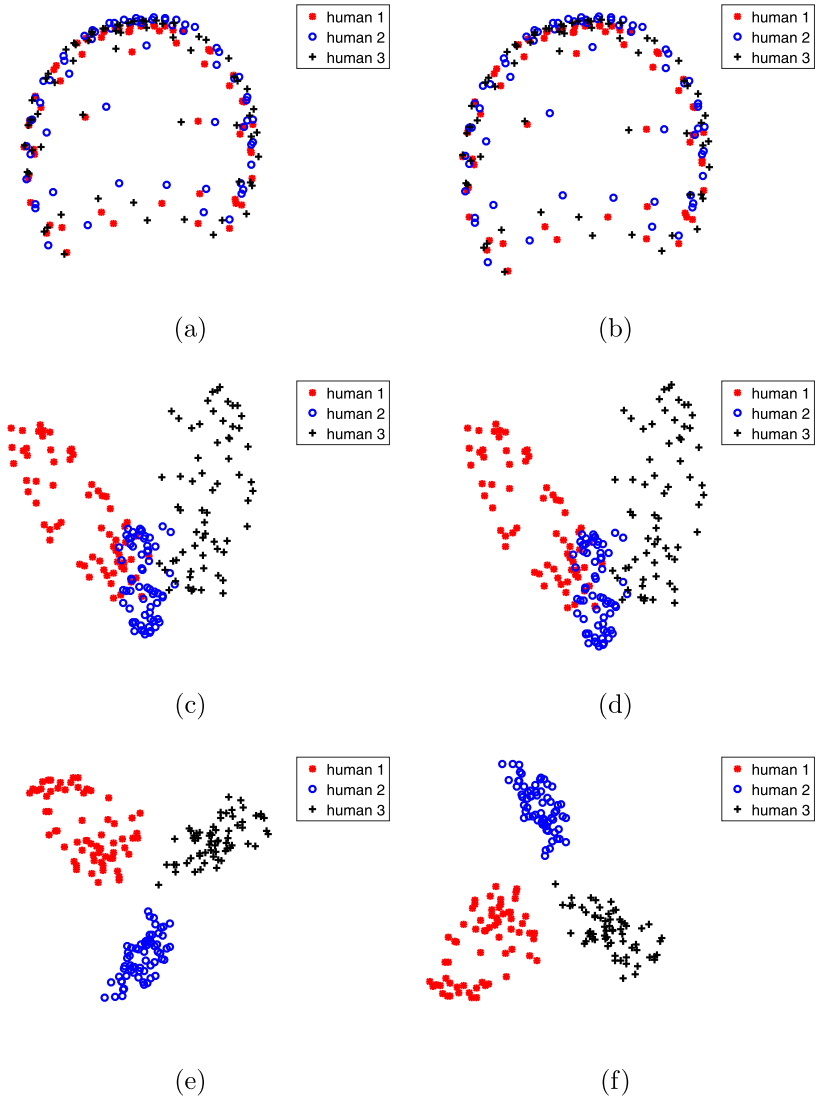


Fig. 3. A comparison of original faces, whitening faces and local whitening faces using PCA and PCA II for a three-class problem (The original faces are from three individuals with labels 22, 32 and 34 in the Extended Yale-B database). (a)-(b) PCA and PCA II with original faces, (c)-(d) PCA and PCA II with whitening faces, (e)-(f) PCA and PCA II with local whitening faces.

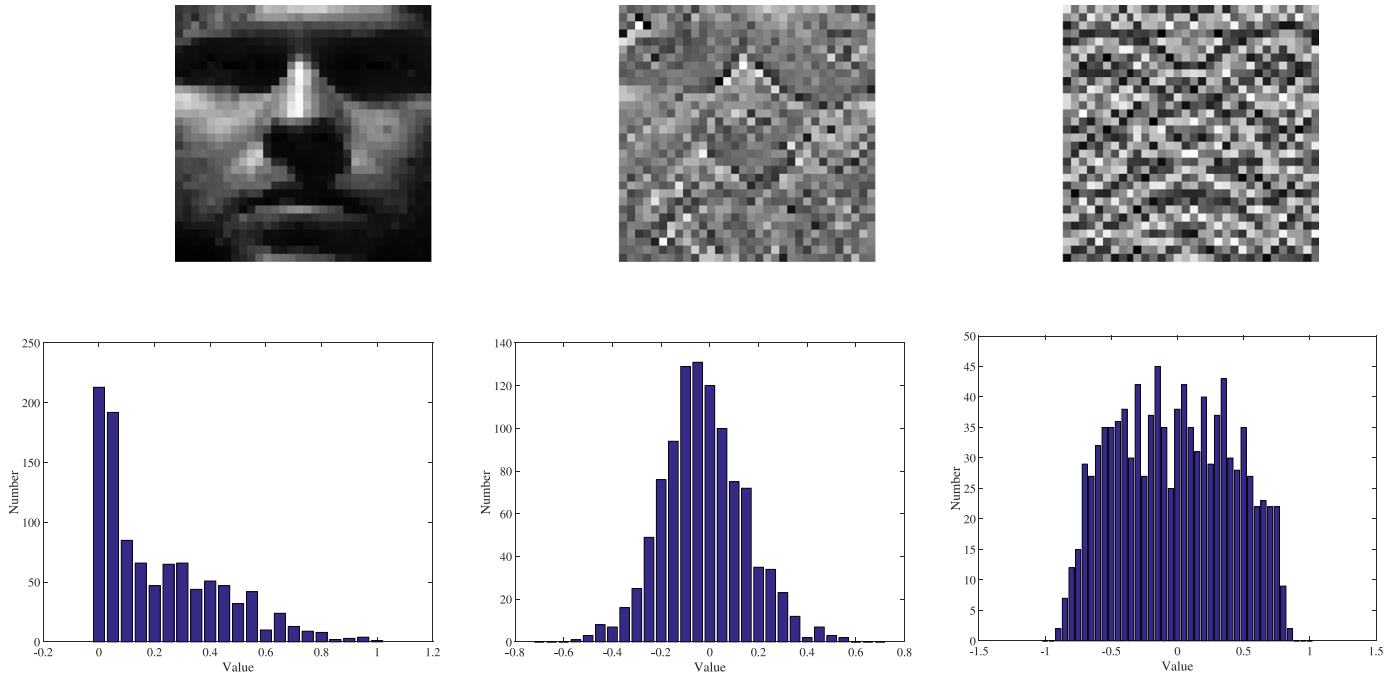


Fig. 4. Pixel value distributions of local whitening face. From left to right, the first row: Original face from the Extended Yale-B database, its global and local whitening face, respectively; the second row are their corresponding distributions.

It is worth noting that in Theorem 1, we suppose the mean of $\mathbf{R}$ to be known when calculating the variance. If the mean is unknown, the variance $D(\mathbf{R}) = \frac{1}{pn-1} - \frac{a^2b^2}{pn(pn-1)}$. Additionally, $a \le \sqrt{p}$, because, $(u_1 + u_2 + \cdots + u_p)^2 \le p(u_1^2 + u_2^2 + \cdots + u_p^2) \Rightarrow |u_1 + u_2 + \cdots + u_p| \le \sqrt{p}$. Similarly, we can attain $b \le \sqrt{n}$.

Theorem 2. *For two orthogonal matrices $\tilde{\mathbf{U}} \in \mathbb{R}^{p\times m}$ and $\tilde{\mathbf{V}} \in \mathbb{R}^{n\times m}$, $\tilde{\mathbf{U}}^T\tilde{\mathbf{U}} = \mathbf{I}_m$ and $\tilde{\mathbf{V}}^T\tilde{\mathbf{V}} = \mathbf{I}_m$, assume that $\tilde{\mathbf{u}}_i\tilde{\mathbf{v}}_i^T \in \mathbb{R}^{p\times n}$ contains pn statistically independent samples of a random variable $\mathbf{R}_i$, and $\mathbf{R} = \sum_{i=1}^m \mathbf{R}_i$. Then $E(\mathbf{R}) = \frac{\sum_{i=1}^m a_ib_i}{pn}$ and $D(\mathbf{R}) = \frac{m}{pn} - \frac{(\sum_{i=1}^m a_ib_i)^2}{p^2n^2}$, where $a_i = \mathbf{1}_p^T\tilde{\mathbf{u}}_i$ and $b_i = \mathbf{1}_n^T\tilde{\mathbf{v}}_i$.*

Proof. The mean of $\mathbf{R}$ is equivalent to:

$$E(\mathbf{R}) = E\left(\sum_{i=1}^m \mathbf{R}_i\right) = \sum_{i=1}^m E(\mathbf{R}_i) = \frac{\sum_{i=1}^m a_ib_i}{pn}. \tag{4}$$

The variance of $\mathbf{R}$ can be obtained by:

$$\begin{aligned} D(\mathbf{R}) &= D\left(\sum_{i=1}^m \mathbf{R}_i\right) = \sum_{i=1}^m D(\mathbf{R}_i) \\ &= \sum_{i=1}^m E(\mathbf{R}_i^2) - (E(\mathbf{R}_i))^2 \\ &= \frac{m}{pn} - \frac{\sum_{i=1}^m (a_ib_i)^2}{p^2n^2}. \end{aligned} \tag{5}$$

where the second equality is based on $E(\mathbf{R}_i\mathbf{R}_j) = 0$ $(i \ne j)$, because $\tilde{\mathbf{u}}_i$ and $\tilde{\mathbf{u}}_j$, $\tilde{\mathbf{v}}_i$ and $\tilde{\mathbf{v}}_j$ are orthogonal, respectively. The third equality is on the basis of Theorem 1. □

In Theorem 2, when $a_i \to 0$ or $b_i \to 0$ $(1 \le i \le m)$, we have $E(\mathbf{R}) \to 0$ and $D(\mathbf{R}) \to \frac{m}{pn}$. For each whitening face in [24], if $rank(\tilde{\mathbf{X}}) = m$, we can get $\mathbf{U} = \mathbf{X}_v\mathbf{V}(:, 1:m)(\mathbf{D}(1:m, 1:m))^{-1}$. Because $\mathbf{1}_p^T\mathbf{X}_v = 0$, $\mathbf{1}_p^T\mathbf{u}_i$ $(1 \le i \le m)$ would be 0 and then $E(\mathbf{R}) = 0$ and $D(\mathbf{R}) = \frac{m}{pn}$. In practice, although $rank(\mathbf{X}_v)$ might be smaller than m, it is very close to m. Thus $\mathbf{1}_p^T\mathbf{u}_i \to 0$ $(1 \le i \le m)$. As a result, the distribution of each whitening face is approximate to $\mathcal{N}(0, \frac{m}{pn})$.

Since each patch in PTWR is obtained by TWR, the distribution of each whitening patch $\tilde{\mathbf{X}}_i$ $(1 \le i \le q)$ is close to $\mathcal{N}(0, \frac{r}{hw})$, where $r \le min(h, w)$ is the rank of the whitening patch. Assume that $\tilde{\mathbf{X}}_i$ contains hw statistically independent samples of a random variable $\mathbf{R}_i$. $\tilde{\mathbf{X}} = \bigcup_{i=1}^q \tilde{\mathbf{X}}_i$, $E(\mathbf{R}) = \frac{1}{q}\sum_{i=1}^q \mathbf{R}_i$ and $D(\mathbf{R}) = \sum_{i=1}^q D(\mathbf{R}_i)$ when $\mathbf{R}_1$, $\mathbf{R}_2$, $\cdots$, $\mathbf{R}_q$ are mutually independent. Hence the distribution of local whitening face is approximate to $\mathcal{N}(0, \frac{rq}{hw})$, where $q = \frac{pn}{hw}$.

4. Self-learning for face clustering

After obtaining the preprocessed face images, it is necessary to extract significant and useful knowledge to recognize faces. SPCL [30] suggests that (i) using the prior knowledge and the learning scheme of "from easy to complex" in training can significantly improve the prediction performance. Meanwhile, we also observe that (ii) in real world the prior knowledge of humans might change with increasing learned knowledge (The prior views of human often change because of important scientific discoveries [36,37]), and (iii) more prior knowledge usually produces better prediction results. Motivated by these observations, we propose a novel self-paced learning framework without any label information. Its detailed procedure is shown in Algorithm 3, namely unsupervised self-paced learning (USL). Specifically, the optimization model of the proposed framework is:

$$\begin{aligned} &\min_{\mathbf{W}^t, \mathbf{v}^t \in \{0,1\}^N} E(\mathbf{W}, \mathbf{v}; \mathbf{C}) \\ &= \sum_{i=1}^N v_i^t \min_{1\le k\le K} L(\mathbf{c}_k^t, g(\mathbf{z}_i, \mathbf{W}^t)) - \lambda \sum_{i=1}^N v_i^t, \end{aligned} \tag{6a}$$

$$\mathbf{C}^{t+1} = c(\mathbf{C}^t, \mathbf{Z}, \mathbf{W}^t)), \tag{6b}$$

$$\mathbf{Y}^{t+1} = s(\mathbf{C}^{t+1}, \mathbf{Z}, \mathbf{W}^t), \tag{6c}$$

$$\mathbf{W}^{t+1} = h(\mathbf{Y}^{t+1}), \tag{6d}$$

Algorithm 3 Unsupervised self-paced learning (USL).

Input: Matrix $\mathbf{Z}$, prior knowledge $\mathbf{C}_0$, stepsize μ, maximum number of iterations T.

1. Initialize $t=0$, $\mathbf{Y}_0=s(\mathbf{Z},\mathbf{C}_0)$ and $\mathbf{W}_0=h(\mathbf{Y}^0)$;
2. **while not converge or reach the maximum iterations**
3. **Repeat**
4. $\mathbf{v}^{t*}\leftarrow \min_{\mathbf{v}^t} E(\mathbf{W}^{t*},\mathbf{v}^t;\mathbf{C}^t)$;
5. $\mathbf{W}^{t*}\leftarrow \min_{\mathbf{W}^t} E(\mathbf{W}^t,\mathbf{v}^{t*};\mathbf{C}^t)$;
6. **if** λ is small **then** increases λ based on the stepsize μ.
7. **Until convergence**;
8. $\mathbf{C}^{t+1}\leftarrow c(\mathbf{C}^t,\mathbf{Z},\mathbf{W}^t)$;
9. $\mathbf{Y}^{t+1}\leftarrow s(\mathbf{C}^{t+1},\mathbf{Z},\mathbf{W}^t)$;
10. $\mathbf{W}^{t+1}\leftarrow h(\mathbf{Y}^{t+1})$;
11. **end while**
12. $lb_i\leftarrow \min_{1\le k\le K}\left\|\mathbf{c}_k^*-\mathbf{z}_i^T\mathbf{W}^*\right\|_2^2$, $(1\le i\le N)$, where lb_i is the prediction label of the sample $\mathbf{z}_i$.
12. **Output:** Labels **lb** corresponding to $\mathbf{Z}$.

where t denotes the tth loop iteration, $\mathbf{Z}\in\mathbb{R}^{d\times N}$ represents the total training samples, $\mathbf{v}$ is a weight vector to determine whether the data points are selected to learn the model parameter $\mathbf{W}\in\mathbb{R}^{d\times r}$ $(r<<d)$, λ is an adjustable parameter to control the pace of selecting samples, and $\mathbf{C}\in\mathbb{R}^{r\times K}$ (K is the number of groups (classes)) denotes the prior knowledge (Here, $\mathbf{C}$ represents the class or group center), which determines the prior knowledge base $\mathbf{Y}\in\mathbb{R}^{b\times d}$. $\mathbf{C}$ and $\mathbf{Y}$ are determined by functions $c(\cdot)$ and $s(\cdot)$, respectively.

Eq. (6)a is similar to the objective function of Kumar et al. [35], and both of them are to learn from easy to complex samples. The function $g(\cdot)$ is to reduce the dimensionality of samples $\mathbf{Z}$. We select $g(\mathbf{z}_i,\mathbf{W}))=\mathbf{z}_i^T\mathbf{W}$ that is a decision function used in fisherfaces [5]. The function $min_{1\le k\le K}L(\cdot)$ is to calculate the loss between one sample and its nearest group center. We adopt $L(\mathbf{c}_k,g(\mathbf{z}_i,\mathbf{W}))=\left\|\mathbf{c}_k-\mathbf{z}_i^T\mathbf{W}\right\|_2^2$. The learning process of Eq. (6)a can be divided into two main sub-steps:

1) Fixing $\mathbf{W}^t$ and updating $\mathbf{v}^t$:

$$\mathbf{v}^{t*}=\begin{cases}1 & \min_{1\le k\le K} L(\mathbf{c}_k,g(\mathbf{z}_i,\mathbf{w}))<\lambda,\\ 0 & otherwise.\end{cases} \tag{7}$$

Different with Eq. (2) just selecting confident samples, Eq. (7) not only selects confident samples but also assigns labels to them. This sub-step corresponds to step 4 in Algorithm 3. Since λ is an adjustable parameter to determine the stepsize μ, in practice we adjust it to make each class add one sample for each update, i.e. $\mu=1$, in order to utilize balanced data to smoothly learn model parameters $\mathbf{W}^t$.

2) Fixing $\mathbf{v}^t$ and updating $\mathbf{W}^t$: based on the selected confident samples and their labels, we learn the projection matrix $\mathbf{W}^t$ by using the fisherfaces. This sub-step corresponds to step 5 in Algorithm 3. Empirically, we repeat these two steps until assigning all samples labels.

Eq. (6)b aims to change the prior knowledge $\mathbf{C}^t$ based on the learned model parameter $\mathbf{W}^t$. The function $c(\cdot)$ is to obtain updated group centers $\mathbf{C}^{t+1}$ based on $\mathbf{W}^t$. In $\mathbf{C}^{t+1}=c(\mathbf{C}^t,\mathbf{Z},\mathbf{W}^t)$, $\mathbf{c}_k^{t+1}=\frac{1}{n_k}\sum_{i=1}^{n_k}\mathbf{Z}_k^T\mathbf{W}^t$, where $\mathbf{Z}_k$ represents the samples belonging to kth class and it is obtained by using $min_{1\le k\le K}L(\mathbf{c}_k^t,g(\mathbf{z}_i,\mathbf{W}^t))=min_{1\le k\le K}\left\|\mathbf{c}_k^t-\mathbf{z}_i^T\mathbf{W}^t\right\|_2^2$ to select samples belonging to the kth class. Eq. (6)b corresponds to Step 8 in Algorithm 3.

The goal of Eq. (6)c is to update and increase the prior knowledge base $\mathbf{Y}^t$. $s(\cdot)$ is to select several nearest neighbors of each group center from $\mathbf{Z}$. $s(\cdot)$ contains two sub-steps: 1) obtaining $\mathbf{Z}_k$ by calculating $min_{1\le k\le K}L(\mathbf{c}_k^{t+1},g(\mathbf{z}_i,\mathbf{W}^t))=min_{1\le k\le K}\left\|\mathbf{c}_k^{t+1}-\mathbf{z}_i^T\mathbf{W}^t\right\|_2^2$ for each sample; 2) attaining $\mathbf{Y}_k^{t+1}$, which denotes the samples belonging to the kth class in the prior knowledge base, i.e. $\mathbf{Y}_k^{t+1}\subset\mathbf{Y}^{t+1}$, by solving the following model:

$$\min_{\mathbf{v}_k\in\{0,1\}^{n_k}}\sum_{i=1}^{n_k}v_k\left\|\mathbf{c}_k^{t+1}-\mathbf{z}_{ki}^T\mathbf{W}^t\right\|_2^2,\ s.t.\ \sum_{i=1}^{n_k}v_k=n_k^s, \tag{8}$$

where $\mathbf{z}_{ki}\subset\mathbf{Z}_k$ represents the ith sample belonging to the kth class, and n_k^s is the selected number of samples for the kth class. We repeat above two sub-steps K times to attain the prior knowledge base $\mathbf{Y}^{t+1}$. Eq. (6)c corresponds to Step 9 in Algorithm 3.

Eq. (6)d updates $\mathbf{W}^t$ based on the updated prior knowledge base $\mathbf{Y}^{t+1}$ that contains selected samples and their assigned labels. We set $h(\cdot)$ to be the objective function in fisherfaces, which can utilize the labeled samples in $\mathbf{Y}^{t+1}$ to learn the projection matrix $\mathbf{W}^{t+1}$. Eq. (6)d corresponds to Step 10 in Algorithm 3.

4.1. Difference to related work

We compare two most related work: SPCL [30] and joint unsupervised learning (JULE) [42]. Different with SPCL using a fixed prior knowledge, USL takes into account the changed prior knowledge and a flexible prior knowledge base, and thus it is more suitable for unsupervised learning problems. One major difference between JULE and USL is that USL trains samples in a meaningful order instead of randomly selecting samples. In addition, JULE uses a convolution neural network (CNN) and a recurrent neural network (RNN) to learn image representations. However, CNN and RNN usually require sufficient training data to adjust many parameters to obtain good performance and they are more suitable for large-scale databases.

Based on Algorithms 2 and 3, we present a self-learning framework for face clustering in Algorithm 4. Specifically, PTWR has a

Algorithm 4 Self-learning based face clustering (SLFC).

Input: Face images $\mathbf{X}^1,\mathbf{X}^2,\cdots\mathbf{X}^N$, number of classes K, stepsize μ, maximum number of iterations T.

1. Calculate $\tilde{\mathbf{X}}^i$ $(1\le i\le N)$, where $\tilde{\mathbf{X}}^i$=**PTWR**$(\mathbf{X}^i)$,
2. Calculate $\mathbf{z}_i$ $(1\le i\le N)$ via reducing the dimensionality of $vector(\tilde{\mathbf{X}}^i)$,
3. Calculate $\mathbf{C}_0\leftarrow$**K-means**$(\mathbf{Z},K)$;
4. Calculate **lb** $\leftarrow$ **USL** $(\mathbf{Z},\mathbf{C}_0;\mu)$;

Output: Labels **lb** corresponding to faces $\mathbf{X}^i$ $(1\le i\le N)$.

function similar to the retina of human eyes, which aims to reduce the redundancy of face images. Then, to obtain the data matrix $\mathbf{Z}\in\mathbb{R}^{d\times N}$, we use PCA or PCA II [41] to reduce the dimensionality of $vector(\tilde{\mathbf{X}}^i)\in\mathbb{R}^{pn}$ $(1\le i\le N)$, which represents the vector form of the ith 2D sample $\tilde{\mathbf{X}}^i\in\mathbb{R}^{p\times n}$. Next, the prior knowledge $\mathbf{C}_0$ can be achieved with K-means. Finally, a self-paced learning process (Algorithm 3) is used to gradually recognize faces.

4.2. Time complexity analysis

The time cost of Algorithm 4 is determined by four main steps: PTWR (step 1), dimensionality reduction (step 2), K-means (step 3) and USL (step 4). The time complexity of PTWR is $min(\mathcal{O}(hpn),\mathcal{O}(wpn))$, where h and w are the height and width of image patches, respectively, and p and n are the height and width of images, respectively. Hence step 1 to process N images with PTWR requires $min(\mathcal{O}(hpnN),\mathcal{O}(wpnN))$ operations. In step 2, calculating $\mathbf{z}_i$ $(1\le i\le N)$ via a low-dimensional projection matrix $\mathbf{P}\in\mathbb{R}^{pn\times d}$ requires $\mathcal{O}(dpnN)$ operations, while learning the low-dimensional projection matrix by employing PCA or PCA II usually needs $max(\mathcal{O}(p^3n^3),\mathcal{O}(Np^2n^2))$ operations. Thus the time complexity of step 2 is $max(\mathcal{O}(p^3n^3),\mathcal{O}(Np^2n^2))$. In

step 3 the time complexity of K-means is $\mathcal{O}(NdKl)$, where l is the number of its iterations. The time cost of step 4 is determined by solving its four subproblems Eq. (6)a–d. The time complexity of Eq. (6)a is $max(\mathcal{O}(N^2d^2), \mathcal{O}(Nd^3))$ determined by using the fisherfaces to learn model parameters $\mathbf{W}^t$. Both Eq. (6)b and c spend at most $\mathcal{O}(Ndr)$ operations. Eq. (6)d learns model parameters by employing the fisherfaces and the prior knowledge base costs $max(\mathcal{O}(d^3), \mathcal{O}(bd^2))$ operations. Thus, the time complexity of step 4 is $max(\mathcal{O}(TN^2d^2), \mathcal{O}(TNd^3))$. Therefore, the total time complexity of Algorithm 4 is $max(\mathcal{O}(p^3n^3), \mathcal{O}(Np^2n^2), \mathcal{O}(TN^2d^2), \mathcal{O}(TNd^3))$.

5. Experimental results and analysis

In our experiments, we evaluate the proposed framework SLFC on three benchmark databases: Extended Yale-B [40] and AR [43] databases in controlled environments as well as LFW [44] database in uncontrolled environments. The selected databases are described as follows:

Extended Yale-B database [40] consists of 16,128 face images of 38 human subjects under 9 poses and 64 illumination conditions. We select 2414 face images with frontal pose under different illumination conditions. Each face image is cropped and resized to 32×32 pixels.

AR database [45] contains over 4000 color images of 126 human subjects (70 men and 56 women) under different facial expressions, illumination conditions and occlusions. We choose 3120 face images of 120 individuals (70 men and 50 women), and crop and resize each face image to 50×40 pixels in our experiment.

LFW database [44] consists of face images of 5749 human subjects in unconstrained environments. We select a subset constituted by 30 individuals, and each of them has over 20 images but no more than 30 images. All face images are aligned by a deep learning method [46] and then cropped and resized to 40×40 pixels.

5.1. Experimental setting

We compare local whitening faces produced by PTWR against the global whitening face generated by TWR. PCA and PCA II [41] are used to reduce their dimensionality. To evaluate the performance of clustering, we use the semantic label as ground truth, and adopt the same definition of clustering accuracy as [24]:

$$Cluster\ accuracy = \frac{number\ of\ correct\ classified\ images}{number\ of\ total\ images}. \quad (9)$$

Then we add an unsupervised self-learning process (Algorithm 3) into the global or local whitening face. To better demonstrate the strength of the proposed framework, we show the comparative performance of eight state-of-the-art methods: K-means (KM), PCA, PCA II, low-rank representation (LRR) [47], sparse subspace clustering (SSC) [48], thresholding ridge regression (TRR) [49], OMP+SSC [50], Aℓ_0-SSC [51], on original faces. Additionally, we also show the clustering accuracy of SSC and TRR with adding the preprocessing procedure TWR and PTWR.

On Extended Yale-B, AR and LFW databases, we randomly select $K = [10, 20, 30, 38]$, $K = [20, 40, 80, 120]$ and $K = [5, 10, 20, 30]$ individuals for clustering, respectively. We empirically set the patch size to be 4×4, 5×4 and 4×4 for PTWR on Extended Yale-B, AR and LFW databases, respectively and $r = 4$ on all three databases. Note that when setting patch size, patches with the same scale as original faces can empirically achieve the best or sub-best performance. Additionally, the patch size should be larger than 2×2, which is too small to capture structural information of faces. For PCA and PCA II on the original, global or local whitening faces, we utilize them to reduce the number of dimensions of each image vector to $2K$, on which PCA and PCA II can achieve their best or sub-best performance. For the maximum number of iterations in USL, we set it to be 15, 10, and 5 on Extended Yale-B, AR and LFW databases, respectively. Additionally, we set $\mu = 1$ in our experiments in order to learn model parameters smoothly. For fair comparison, K-means is ran 10 times for the same samples. We repeat above processes 10 times and calculate the average clustering accuracy.

5.2. Experimental results

Table 1 presents the clustering accuracy of different algorithms on Extended Yale-B, AR and LFW databases. These algorithms are divided into four groups: (1) KM, PCA, PCA II, LRR, SSC, TRR, OMP+SSC and Aℓ_0-SSC; (2) TWR+SSC, TWR+TRR, PTWR+SSC and PTWR+TRR; (3) TWR+PCA, TWR+PCA II, PTWR+PCA and PTWR+PCA II; (4) TWR+PCA+USL, TWR+PCA II+USL, PTWR+PCA+USL and PTWR+PCA II+USL. We bold the best accuracy of each group.

On the Extended Yale-B database, PTWR can greatly improve the performance of SSC, TRR, PCA and PCA II, and it exhibits significantly superior performance to TWR except the case that TWR and PTWR add USL at $k = 10$, e.g. the average clustering accuracy of PTWR+PCA and PTWR+PCA II is 28.5% and 16.2% higher than TWR+PCA and TWR+PCA II, respectively. Moreover, USL can further improve the accuracy of PTWR+PCA and PTWR+PCA II. Specifically, PTWR+PCA+USL and PTWR+PCA II+USL obtain 18.9% and 19.8% higher average accuracy than PTWR+PCA and PTWR+PCA II, respectively. Compared to the algorithms KM, PCA, PCA II, LRR, SSC, TRR, OMP+SSC and Aℓ_0-SSC, PTWR+PCA+USL and PTWR+PCA II+USL also show better performance, i.e. the gain in average accuracy of PTWR+PCA+USL is 9.1% over the best competitor TRR. Additionally, compared to TWR+SSC, TWR+TRR, PTWR+SSC and PTWR+TRR, PTWR+PCA+USL and PTWR+PCA II+USL have slightly higher average accuracy than the best competitor PTWR+SSC.

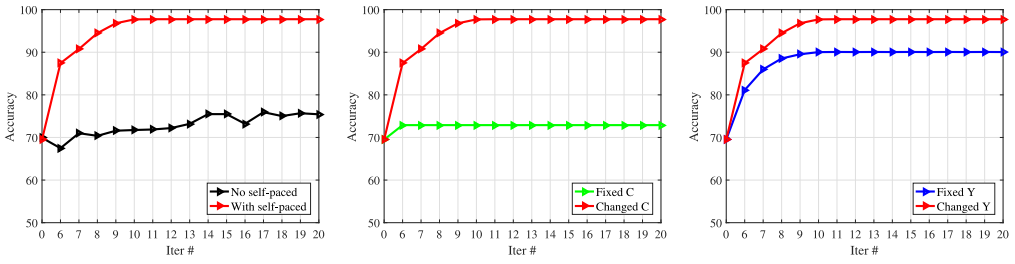
On the AR database, both TWR and PTWR can improve the accuracy of PCA and SSC, while they cannot always boost the accuracy of TRR and PCA II. Additionally, the average performance of TWR and PTWR is similar (within 1.5%) on SSC, TRR, PCA and PCA II. However, with adding USL, PTWR shows slightly better accuracy than TWR. For example, PTWR+PCA+USL attains 4.1% higher average accuracy than TWR+PCA+USL. Furthermore, with USL, the clustering accuracy of all TWR+PCA, TWR+PCA II, PTWR+PCA and PTWR+PCA II are significantly improved, e.g. the gain of PTWR+PCA+USL ranges from 19.6% to 42.6% over PTWR+PCA. Compared to the algorithms KM, PCA, PCAII, LRR, SSC, TRR, OMP+SSC, Aℓ_0-SSC, TWR+SSC, TWR+TRR, PTWR+SCC and PTWR+TRR, PTWR+PCA+USL achieves better average clustering accuracy (75.2%).

On the LFW database, both TWR and PTWR can significantly improve the performance of PCA, PCA II, SSC and TRR. For example, the gain in accuracy of PTWR+TRR ranges from 43.8% to 61.0% over TRR. Additionally, PTWR shows superior performance to TWR except on PCA and PCA II at $K = 5$. USL can further improve the clustering accuracy of TWR+PCA, TWR+PCA II, PTWR+PCA and PTWR+PCA II in most of cases. Among all algorithms, PTWR+TRR and PTWR+PCA+USL achieve very similar average clustering accuracy and outperform the other algorithms.

Based on experimental results on the three databases, we can observe that (i) Both PTWR and USL are effective and efficient on face images under illumination changes; (ii) PTWR and TWR obtain similar accuracy in many cases when faces images under different facial expressions, illumination conditions and occlusions, probably because some occluded patches without containing discriminative information are enhanced by PTWR. However, USL are still effec-

Table 1
Clustering results on three benchmark databases (PCA means PCA+K-means, similar for the other unsupervised algorithms except TWR+PCA+USL, TWR+PCA II+USL, PTWR+PCA+USL and PTWR+PCA II+USL).

Method	Extended Yale-B					AR					LFW				
	10	20	30	38	Avg	20	40	80	120	Avg	5	10	20	30	Avg
KM	16.2	11.2	9.9	9.9	11.8	46.0	39.7	31.3	26.2	35.8	40.4	29.4	20.8	17.3	27.0
PCA	12.4	8.6	7.4	6.6	8.8	48.2	39.4	33.4	30.2	37.8	43.0	25.5	19.2	18.2	26.5
PCA II	27.4	25.5	25.7	24.3	25.7	66.8	60.3	53.4	48.9	57.4	41.3	28.4	23.1	20.8	28.4
LRR	88.3	79.7	72.2	69.4	77.4	37.7	29.4	24.7	24.8	29.2	24.4	13.9	12.2	10.0	15.1
SSC	89.5	77.6	75.4	73.9	79.1	71.3	64.9	61.9	60.7	64.7	**52.9**	**36.5**	**31.8**	**31.4**	**38.2**
TRR	**96.1**	88.8	**86.0**	**85.3**	**89.1**	**78.1**	**74.2**	**69.1**	**66.2**	**71.9**	46.7	34.4	31.0	27.6	34.9
OMP+SSC	82.8	80.1	82.0	81.2	81.5	69.0	49.5	49.9	45.6	53.5	49.5	31.9	25.7	22.6	32.4
Aℓ_0-SSC	72.4	**88.9**	83.8	82.9	82.0	65.8	47.9	53.6	49.9	54.3	48.3	28.0	27.0	23.5	31.7
TWR+SSC	80.9	68.8	59.6	59.8	67.3	75.3	74.4	**69.3**	**68.6**	71.9	71.8	53.9	43.8	34.8	51.1
TWR+TRR	90.8	98.5	**95.0**	89.9	93.6	78.2	**75.8**	69.1	65.7	**72.2**	67.4	50.9	43.4	35.6	49.3
PTWR+SSC	**99.4**	98.5	93.7	**94.8**	**96.6**	**81.1**	74.3	65.6	63.3	71.1	**73.3**	**56.0**	46.5	38.0	53.4
PTWR+TRR	95.9	**99.6**	94.6	93.0	95.8	78.8	74.4	66.2	63.5	70.7	70.1	55.2	**49.9**	**39.7**	**53.7**
TWR+PCA	52.6	50.0	48.5	48.2	49.8	63.6	57.6	55.2	53.3	57.4	**73.2**	46.1	36.1	27.7	**45.8**
TWR+PCAII	59.4	59.1	61.0	63.9	60.9	59.3	56.9	**56.0**	**54.2**	56.6	70.9	45.7	35.6	26.8	44.8
PTWR+PCA	**85.3**	**80.1**	77.0	70.9	**78.3**	**71.4**	**60.1**	53.9	48.4	**58.5**	51.9	50.9	**41.1**	**34.7**	42.2
PTWR+PCAII	77.0	74.6	**79.8**	**77.0**	77.1	58.9	56.8	**56.0**	53.8	56.4	52.0	**51.2**	39.6	31.9	43.7
TWR+PCA+USL	**95.1**	96.1	90.9	87.6	92.4	76.6	73.0	67.3	67.6	71.1	**72.4**	58.6	39.7	33.9	51.2
TWR+PCAII+USL	94.3	92.6	86.7	85.8	89.9	74.6	72.4	65.0	62.5	68.6	70.6	56.0	40.5	33.1	50.1
PTWR+PCA+USL	93.8	97.6	**98.7**	**98.6**	**97.2**	**85.4**	**78.5**	**68.1**	**69.0**	**75.2**	69.0	**60.0**	46.5	**40.2**	**53.9**
PTWR+PCAII+USL	92.8	**97.7**	98.3	**98.6**	96.9	82.0	74.5	64.4	63.5	71.1	66.7	56.3	**47.4**	38.3	52.1



(a) @ $K = 38$ (b) @ $K = 38$ (c) @ $K = 38$

Fig. 5. Influence of three major factors on clustering accuracy. (a) The process 'learning from easy to complex' (Eq. (6)a); (b) The changeable **C** (Eq. (6)b); (c) Flexible **Y** in Eq. (6)c. (Here, we adopt the method PTWR+PCA+USL and select face images from the Extended Yale-B database to test the three major factors.)

tive when face images are under various facial expressions, illumination conditions and occlusions; (iii) When face images are in unconstraint environments, PTWR exhibits promising performance on PCA, PCA II, SSC and TRR. USL can further improve the clustering accuracy of TWR+PCA, TWR+PCA II, PTWR+PCA and PTWR+PCA II in most of cases.

5.2.1. Influence of three major factors on accuracy

In this subsection, we show the influence of the process 'learning from easy to complex' (Eq. (6)a), the changeable prior knowledge **C** (Eq. (6)b), the different size of prior knowledge base **Y** in Eq. (6)c, on clustering accuracy.

In Fig 5, the method PTWR+PCA+USL is used and 2414 face images of 38 individuals are selected from the Extended Yale-B face database. When the number of iterations is 0, the accuracy in Fig 5a–c is equivalent to that of PTWR+PCA+KM; When the number of iterations is equal to 6, **Y** contains 6 samples of each class in Fig 5a–c, while it has 5 samples of each class as **Y** is fixed in Fig 5c. With the increasing of iterations, the sample number of each class contained in **Y** is equivalent to the number of iterations, while the size of **Y** in 'Fixed **Y**' shown in Fig 5c is unchanged.

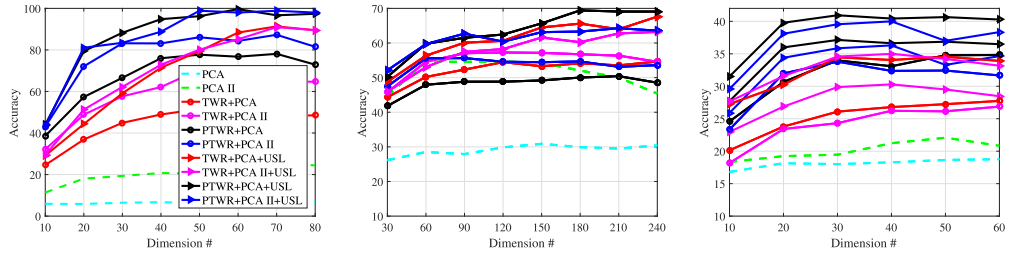
Fig 5a shows that when removing the process Eq. (6)a but preserving Eq. (6)b–d, the clustering accuracy will be greatly decreased. It illustrates the significance of the process to learn from easy to complex samples. Fig 5b, removing Eq. (6)b but preserving Eq. (6)a, c and d, displays the effectiveness of the changeable prior knowledge **C**. When the size of **Y** is fixed, Fig 5c presents that the increasing size of **Y** can boost the clustering accuracy. Note that in Fig 5c, Eq. (6)c is not removed but the size of **Y** is unchanged for 'Fixed **Y**'. For the algorithms TWR+PCA +USL, TWR+PCA II+USL and PTWR+PCA II+USL, and on different databases, similar findings can be observed.

5.2.2. Accuracy versus dimensions

Fig 6 shows the accuracy of PCA, PCA II, TWR+PCA, TWR+PCA II, PTWR+PCA, PTWR+PCA II, TWR+PCA+USL, TWR+PCA II+USL, PTWR+PCA+USL and PTWR+PCA II+USL on three kinds of faces (original, whitening and local whitening) from three databases with respect to different number of dimensions. As we can see, most of algorithms can attain stable performance within $2K$ dimensions. This is the reason for that we reduce the number of dimensions of each image vector to $2K$ in Table 1. Additionally, both TWR and PTWR can significantly boost the performance of PCA and PCA II on Extended Yale-B, AR and LFW databases, and PTWR has superior performance to TWR in most of cases, especially on Extended Yale-B and LFW databases. Moreover, TWR+PCA, TWR+PCA II, PTWR+PCA, PTWR+PCA II with introducing USL can achieve better performance than those without USL in most cases.

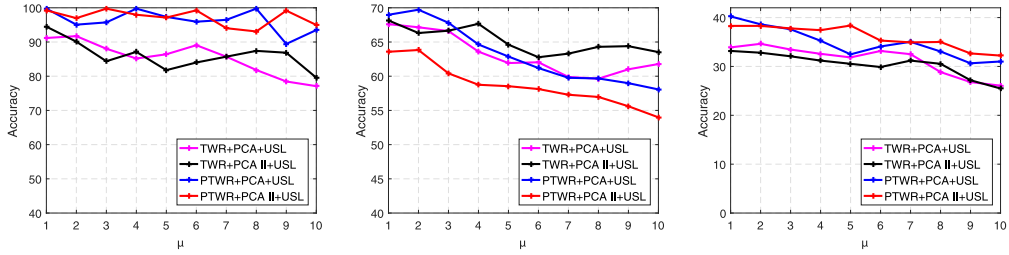
5.2.3. Accuracy versus stepsize

To show the influence of stepsize μ on USL, we display the accuracy of TWR+PCA+USL, TWR+PCA II+USL, PTWR+PCA+USL and PTWR+PCA II+USL on face images for the three databases with respect to different μ in Fig 7. We set the maximum number of iterations in USL to be 15, 10, and 5 on Extended Yale-B, AR and LFW databases, respectively, and repeat the algorithms 10 times to cal-



(a) @ $K = 38$ (b) @ $K = 120$ (c) @ $K = 30$

Fig. 6. Accuracy vs. dimensions reduced by using PCA and PCA II on face images from three databases. (a) Extended Yale-B, (b) AR, (c) LFW.



(a) @ $K = 38$ (b) @ $K = 120$ (c) @ $K = 30$

Fig. 7. Accuracy vs. stepsize on face images for three databases. (a) Extended Yale-B, (b) AR, (c) LFW.

culate the average accuracy. Fig 7 illustrates that USL can attain the best or sub-best accuracy when $\mu = 1$. Probably because a smaller step makes the learning process more smooth.

6. Conclusion

In this paper, we present a self-learning framework without using any label information for face clustering. The proposed framework contains two major steps. First, to simulate the function of the retina of human eyes to reduce the local redundancy of face images and meanwhile preserve local intrinsic significant features, we present a PTWR to obtain local whitening faces, whose distribution has been theoretically analyzed. Then, to simulate the learning way of human to group the faces with semantic similarity, we design and develop a SPL model based on learning knowledge from easy to complex samples, changeable prior knowledge and the enlarged prior knowledge base. Extensive experiments on three benchmark face databases demonstrate the effectiveness and efficiency of the proposed framework. Since deep learning can automatically extract powerful image features, in the future, we will take advantage of both deep learning methods and our SPL model to further improve face recognition without supervision information.

Acknowledgments

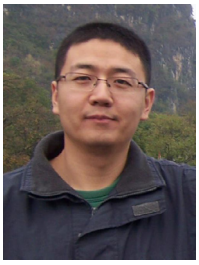
This work was supported by the National Key Research and Development Program of China (No. 2016YFB1001001), the Natural Science Foundation of China (NSFC) (No. 61772296, U1713214).

References

[1] Z. Lei, M.P. Ainen, S.Z. Li, Learning discriminant face descriptor, IEEE Trans. Pattern Anal. Mach. Intell. 36 (2) (2014) 289–302.
[2] M.A.A. Dewan, E. Granger, G.L. Marcialis, R. Sabourin, F. Roli, Adaptive appearance model tracking for still-to-video face recognition, Pattern Recog. 49 (2016) 129–151.
[3] I.T. Jolliffe, Principal component analysis and factor analysis, in: Principal Component Analysis, 1986, pp. 115–128.
[4] M. Turk, A. Pentland, Eigenfaces for recognition, J. Cogn. Neurosci. 3 (1) (1991) 71–86.
[5] P.N. Belhumeur, J.P. Hespanha, D.J. Kriegman, Eigenfaces vs. fisherfaces: recognition using class specific linear projection, IEEE Trans. Pattern Anal. Mach. Intell. 19 (7) (1997) 711–720.
[6] X. He, S. Yan, Y. Hu, P. Niyogi, H. Zhang, Face recognition using laplacianfaces, IEEE Trans. Pattern Anal. Mach. Intell. 27 (3) (2005) 328–340.
[7] J. Wright, A.Y. Yang, A. Ganesh, S.S. Sastry, Y. Ma, Robust face recognition via sparse representation, IEEE Trans. Pattern Anal. Mach. Intell. 31 (2) (2009) 210–227.
[8] Y. Sun, X. Wang, X. Tang, Deep learning face representation from predicting 10,000 classes, in: Proc. Comput. Vision Pattern Recog., 2014, pp. 1891–1898.
[9] Y. Sun, Y. Chen, X. Wang, X. Tang, Deep learning face representation by joint identification-verification, in: Proc. Adv. Neural Info. Process. Syst., 2014, pp. 1988–1996.
[10] X. Shi, Y. Yang, Z. Guo, Z. Lai, Face recognition by sparse discriminant analysis via joint $\ell_{2,1}$-norm minimization, Pattern Recog. 47 (7) (2014) 2447–2453.
[11] X. Shi, Z. Guo, Z. Lai, Y. Yang, Z. Bao, D. Zhang, A framework of joint graph embedding and sparse regression for dimensionality reduction, IEEE Trans. Image Process. 24 (4) (2015) 1341–1355.
[12] D. Kang, H. Han, A.K. Jain, S.W. Lee, Nighttime face recognition at large standoff: cross-distance and cross-spectral matching, Pattern Recog. 47 (12) (2014) 3750–3766.
[13] W. Ou, X. You, D. Tao, P. Zhang, Y. Tang, Z. Zhu, Robust face recognition via occlusion dictionary learning, Pattern Recog. 47 (4) (2014) 1559–1572.
[14] M. Meytlis, L. Sirovich, On the dimensionality of face space, IEEE Trans. Pattern Anal. Mach. Intell. 29 (7) (2007).
[15] L. Sirovich, M. Kirby, Low-dimensional procedure for characterization of human faces, J. Opt. Soc. Am. 4 (3) (1987) 519–524.
[16] G.W. Cottrell, M.K. Fleming, Face recognition using unsupervised feature extraction, in: Proc. Int'l Neural Network Conf., 1990, pp. 322–325.
[17] J. Yu, R. Hong, M. Wang, J. You, Image clustering based on sparse patch alignment framework, Pattern Recog. 47 (11) (2014) 3512–3519.
[18] Z. Zhang, F. Xing, X. Shi, L. Yang, Revisiting graph construction for fast image segmentation, 2017, arXiv:1702.05650
[19] F. Attneave, Some informational aspects of visual perception, Psychol. Rev. 61 (3) (1954) 183.
[20] J.J. Atick, A.N. Redlich, What does the retina know about natural scenes? Neural Comput. 4 (4) (1992) 196–210.
[21] M.S. Bartlett, H.M. Lades, T.J. Sejnowski, Independent component representation for face recognition, in: Proc. SPIE Symposium. Electronic Imaging: Science and Technology, 1998, pp. 528–539.
[22] M.S. Bartlett, J.R. Movellan, T.J. Sejnowski, Face recognition by independent component analysis, IEEE Trans. Neural Netw. 13 (6) (2002) 1450–1464.
[23] M.S. Bartlett, Face Image Analysis by Unsupervised Learning, Springer Science & Business Media, 2012.
[24] X. Shi, Z. Guo, F. Nie, L. Yang, J. You, D. Tao, Two-dimensional whitening reconstruction for enhancing robustness of principal component analysis, IEEE Trans. Pattern Anal. Mach. Intell. 38 (10) (2016) 2130–2136.

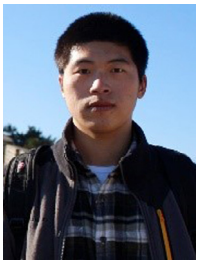
[25] T. Ahonen, A. Hadid, M. Pietikainen, Face description with local binary patterns: application to face recognition, IEEE Trans. Pattern Anal. Mach. Intell. 28 (12) (2006) 2037–2041.
[26] X. Peng, L. Zhang, Z. Yi, K.K. Tan, Learning locality-constrained collaborative representation for robust face recognition, Pattern Recog. 47 (9) (2014) 2794–2806.
[27] X. Tang, B. Triggs, Enhanced local texture feature sets for face recognition under difficult lighting conditions, IEEE Trans. Image Process. 19 (6) (2010) 168–182.
[28] Y. Bengio, J. Louradour, R. Collobert, J. Weston, Curriculum learning, in: Proc Int. Conf. Mach. Learn., 2009, pp. 41–48.
[29] L. Jiang, D. Meng, S.I. Yu, Z. Lan, S. Shan, A.G. Hauptmann, Self-paced learning with diversity, in: Proc. Adv. Neural Info. Process. Syst., 2014, pp. 2078–2086.
[30] L. Jiang, D. Meng, Q. Zhao, S. Shan, A.G. Hauptmann, Self-paced curriculum learning, in: Proc. AAAI Conf. Artificial Intell., vol. 2, 2015, p. 6.
[31] J. Yu, X. Yang, F. Gao, D. Tao, Deep multimodal distance metric learning using click constraints for image ranking, IEEE Trans. Cybern. 47 (12) (2017) 4014–4024.
[32] J.L. Elman, Learning and development in neural networks: the importance of starting small, Cognition 48 (1) (1993) 71–99.
[33] Y. Tang, Y. Yang, Y. Gao, Self-paced dictionary learning for image classification, in: Proc. ACM Int. Conf. Multimedia, 2012, pp. 833–836.
[34] J.S. Supancic, D. Ramanan, Self-paced learning for long-term tracking, in: Proc. Comput. Vision Pattern Recog., 2013, pp. 2379–2386.
[35] M.P. Kumar, B. Packer, D. Koller, Self-paced learning for latent variable models, in: Proc. Adv. Neural Info. Process. Syst., 2010, pp. 1189–1197.
[36] E.J. Dijksterhuis, The Principal Works of Simon Stevin: Vol. 1-3 (i 4 bd), CV Swets & Zeitlinger, 1955.
[37] M. Arndt, O. Nairz, J. Vos-Andreae, C. Keller, G.V.d. Zouw, A. Zeilinger, Wave—particle duality of c60 molecules, Nature 401 (6754) (1999) 680–682.
[38] C. Hong, J. Yu, J. Wan, D. Tao, M. Wang, Multimodal deep autoencoder for human pose recovery, IEEE Trans. Image Process. 24 (12) (2015) 5659–5670.
[39] C.J. Yu, T. Joachims, Learning structural SVMS with latent variables, in: Proc. Int. Conf. Mach. Learn., 2009, pp. 1169–1176.
[40] K.C. Lee, J. Ho, D. Driegman, Acquiring linear subspaces for face recognition under variable lighting, IEEE Trans. Pattern Anal. Mach. Intell. 27 (5) (2005) 684–698.
[41] J. Yang, D. Zhang, J.Y. Yang, Is ICA significantly better than PCA for face recognition? in: Proc. Int. Conf. Comput. Vision., 2005, pp. 198–203.
[42] J. Yang, D. Parikh, D. Batra, Joint unsupervised learning of deep representations and image clusters, in: Proc. Compt. Vision Pattern Recog., 2016, pp. 5147–5156.
[43] T. Sim, S. Baker, M. Bsat, The CMU pose, illumination, and expression database, in: Proc. Int. Conf. Auto. Face Gesture Recog., 2002, pp. 46–51.
[44] G.B. Huang, M. Ramesh, T. Berg, E. Learned-Miller, Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments, University of Massachusetts, Amherst, October, 2007.
[45] A.M. Martinez, R. Benavente, The AR face database, 2003, http://rvl1.Ecn.purdue.edu/aleix_face_DB.Html.
[46] G.B. Huang, M. Mattar, H. Lee, E. Learned-Miller, Learning to align from scratch, in: Proc. Adv. Neural Info. Process. Syst., 2012.
[47] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, Y. Ma, Robust recovery of subspace structures by low-rank representation, IEEE Trans. Pattern Anal. Mach. Intell. 35 (1) (2013) 171–184.
[48] E. Elhamifar, R. Vidal, Sparse subspace clustering: algorithm, theory, and applications, IEEE Trans. Pattern Anal. Mach. Intell. 35 (11) (2013) 2765–2781.
[49] X. Peng, Z. Yi, H. Tang, Robust subspace clustering via thresholding ridge regression, in: Proc. AAAI Conf. Artificial Intell., 2015, pp. 3827–3833.
[50] E.L. Dyer, A.C. Sankaranarayanan, R.G. Baraniuk, Greedy feature selection for subspace clustering, J. Mach. Learn. Res. 14 (1) (2013) 2487–2517.
[51] Y. Yang, J. Feng, N. Jojic, J. Yang, T.S. Huang, Proc. Euro. Conf. Comput. Vision, 2016, pp. 459–468.

Xiaoshuang Shi received the B.S. degree in automation from Northwestern Polytechnical University, China, and M.S. degree in automation from Tsinghua University, China, in 2009 and 2013, respectively. From September 2013 to April 2015, he was a Research Assistant in Shenzhen Key Laboratory of Broadband Network & Multimedia, Graduate School at Shenzhen, Tsinghua University, China. Now, he is pursuing a Ph.D. degree in the J. Crayton Pruitt Family Department of Biomedical Engineering at University of Florida, Gainesville, USA. His current research interests include large-scale image retrieval, pattern recognition and medical image analysis.

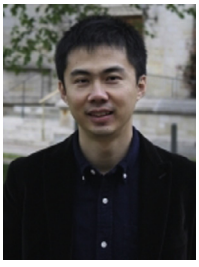


Zhenhua Guo received the M.S. and Ph.D degree in computer science from Harbin Institute of Technology and the Hong Kong Polytechnic University in 2004 and 2010 respectively. Since April 2010, he has worked in Graduate School at Shenzhen, Tsinghua University. His research interests include pattern recognition, texture classification, biometrics, video surveillance, etc.

Fuyong Xing received the bachelors degree from Xian Jiaotong University, Xian, China, the M.S. degree from Rutgers University, New Brunswick, NJ, USA, and the Ph.D. degree in electrical and computer engineering from the University of Florida, Gainesville, FL, USA, in 2017. He is currently an Assistant Professor with the Department of Biostatistics and Informatics, Colorado School of Public Health, University of Colorado Denver, Denver, CO, USA. His current research interests include biomedical image computing, imaging informatics, computer vision, machine learning, and deep learning.



Jinzheng Cai received the B.E. degree in biomedical engineering from the Fudan University, Shanghai, China, in 2013. Now, he is a Ph.D. candidate in the J. Crayton Pruitt Family Department of Biomedical Engineering at University of Florida, Gainesville, USA. His research interests include large-scale image retrieval, pattern recognition and medical image analysis.



Lin Yang is an associate professor in the J. Crayton Pruitt Family Department of Biomedical Engineering, the Department of Electrical and Computer Engineering, and the Department of Computer and Information Science and Engineering at University of Florida. He was an assistant professor in the Department of Radiology and Pathology, and graduate faculty in the Department of Biomedical Engineering at Rutgers University from 2009 to 2011. He was an assistant professor in the Department of Biomedical Informatics and the Department of Computer Science at University of Kentucky from 2011 to 2014. His major research interest is focused on biomedical image analysis, imaging informatics, computer vision, biomedical informatics, and machine learning. He is also working on high performance computing and computed aided health care and information technologies. He leads the Biomedical Image Computing and Imaging Informatics (BICI2) Lab: http://www.bme.ufl.edu/labs/yang/.