Automatic Subspace Learning via Principal Coefficients Embedding

Xi Peng, Jiwen Lu, *Member, IEEE,* Rui Yan, *Member, IEEE,* and Zhang Yi, *Senior Member, IEEE*

***Abstract*—In this paper, we address two problems in unsupervised subspace learning: 1) how to automatically identify the feature dimension of the learned subspace, and 2) how to learn the underlying subspace in the presence of gross corruptions such as Gaussian noise. We show that these two problems are two sides of one coin, *i.e.*, they can be solved by removing possible errors from training data $\mathbf{D} \in \mathbb{R}^{m\times n}$. To achieve this, we propose a new method (called Principal Coefficients Embedding, PCE) that can simultaneously learn a clean data set $\mathbf{D}_0 \in \mathbb{R}^{m\times n}$ and a linear representation (denoted by $\mathbf{C}$) from $\mathbf{D}$. By embedding $\mathbf{C}$ into a k-dimensional space, PCE obtains a projection matrix that preserves some desirable properties of inputs, where $k \ll m$ is exactly the rank of $\mathbf{C}$. PCE has three advantages: 1) it can automatically determine the feature dimension even though data are sampled from a union of multiple linear subspaces; 2) it is robust to various noises and real disguises; 3) it has a closed-form solution and can be calculated very fast. Extensive experimental results show the superiority of PCE on a range of databases with respect to classification accuracy, robustness and efficiency.**

***Index Terms*—Subspace dimension determination, metric learning, graph embedding, corrupted data, robustness.**

I. Introduction

Subspace learning or metric learning aims to find a projection matrix $\mathbf{\Theta} \in \mathbb{R}^{m\times m'}$ from the training data $\mathbf{D}^{m\times n}$, so that the high-dimensional datum $\mathbf{y} \in \mathbb{R}^m$ can be transformed into a low-dimensional space via $\mathbf{z} = \mathbf{\Theta}^T\mathbf{y}$. Existing subspace learning methods can be roughly divided into three categories: supervised, semi-supervised, and unsupervised. Supervised method incorporates the class label information of $\mathbf{D}$ to obtain discriminative features. The well-known works include but not limit to linear discriminant analysis [1], neighbourhood components analysis [2], and their variants such as [3], [4], [5], [6]. Semi-supervised methods [7], [8], [9] utilize limited labeled training data as well as unlabeled ones for better performance. Unsupervised methods seek a low-dimensional subspace without using any label information of training samples. Typical methods in this category include Eigenfaces [10], Neighbourhood Preserving Embedding (NPE) [11], Locality Preserving Projections (LPP) [12], Sparsity Preserving Projections (SPP) [13] or known as L1-graph [14]. For these various subspace learning methods, Yan et al. [15] have shown that most of them can be unified into the framework of graph embedding, *i.e.*, low dimensional features can be achieved by embedding some desirable properties (described by a similarity graph) from a high-dimensional space into a low-dimensional one. By following this framework, this paper focuses on unsupervised subspace learning, *i.e.*, label information is unavailable in training data.

Although a large number of subspace learning methods have been proposed, less works have discussed two challenging problems: 1) how to automatically determine the dimension of the feature space, referred to as *automatic subspace learning*, and 2) how to immune the affect of corruptions, referred to as *robust subspace learning*.

Automatic subspace learning involves the technique of dimension estimation which aims at identifying the number of features necessary for the learned low-dimensional subspace to describe a data set. In previous studies, most existing methods experimentally set the feature dimension by exploring all possible values based on the classification accuracy. Clearly, such a strategy is time-consuming and easily overfits to the specific data set. In the literature of manifold learning, some dimension estimation methods have been proposed, *e.g.*, spectrum analysis based methods [16], [17], box-counting based methods [18], fractal-based methods [19], [20], tensor voting [21], and neighbourhood smoothing [22]. Although these methods have achieved impressive results, this problem is still far from solved due to the following limitations: 1) these methods may work only when data are sampled in a uniform way and data are free to corruptions, as pointed out by Saul and Roweis [23]; 2) most of these methods can accurately estimate the intrinsic dimension of a single subspace and fail to work well for the scenarios of multiple subspaces, especially, when the subspaces are dependent or disjoint; 3) although some dimension estimation methods can be used prior to the final step to set the number of embedding coordinates of previous subspace learning algorithms, it is preferable to design a subspace learning method that can automatically determine the dimension of feature space and reduce the dimension of data at the same time.

Robust subspace learning aims at identifying underlying subspaces even though the training data $\mathbf{D}$ contains gross corruptions. Since $\mathbf{D}$ is corrupted by itself, accurate prior knowledge about the desired geometric properties is hard to be learned from $\mathbf{D}$. Furthermore, grossly corruptions will make dimension estimation more difficult. This robust learning problem, to the best of our knowledge, is seldom studied before. The well-known Principal Component Analysis (PCA) achieves robust results by removing the bottom eigenvectors

Xi Peng and Rui Yan are with Institute for Infocomm Research, Agency for Science, Technology and Research (A*STAR), Singapore 138632. e-mail: pangsaai@gmail.com;ryan@i2r.a-star.edu.sg.

Jiwen Lu is with the Advanced Digital Sciences Center, Singapore 138632. e-mail: jiwen.lu@adsc.com.sg.

Zhang Yi is with Machine Intelligence Laboratory, College of Computer Science, Sichuan University, Chengdu, 610065, China. e-mail: zhangyi@scu.edu.cn.

corresponding to the smallest eigenvalues[1]. However, PCA can achieve a good result only when data are sampled from a single subspace and are contaminated by small Gaussian noise. Moreover, PCA needs specifying a parameter (*e.g.*, 98% energy) to distinct the principal components from the minor ones. To improve the robustness of PCA, Candes et al. recently proposed robust PCA (RPCA) [24] which can handle the sparse corruption and has achieved a lot of success [25], [26], [27]. However, RPCA directly removes the errors from the input space, which cannot obtain the low-dimensional features of inputs. Moreover, the computational complexity of RPCA is too high to handle large scale data set with very high dimensionality. Bao et al. [28] proposed an algorithm which can handle the gross corruption. However, the class label information plays an important role in their model and they did not explore the possibility to automatically determine feature dimension. Tzimiropoulos et al. [29] proposed a subspace learning method from image gradient orientations by replacing pixel intensities of images with gradient orientations. Their method outperforms a lot of popular methods such as Gabor features in illumination- and occlusion-robust face recognition. Besides the above robust subspace learning works, recent development [30], [31], [32], [33] in subspace clustering have also motivated this work a lot.

Based on the above observations, we present a parameter-free method for robust unsupervised subspace learning. The proposed method, referred to as Principal Coefficients Embedding (PCE), formulates the possible corruptions into an objective function so that a clean data set $\mathbf{D}_0$ and the corresponding reconstruction coefficients $\mathbf{C}$ can be simultaneously learned from the training data $\mathbf{D}$. By embedding $\mathbf{C}$ into a m'-dimensional space, PCE obtains a projection matrix $\mathbf{\Theta}^{m\times m'}$, where m' is determined by the rank of $\mathbf{C}$. Our dimension determination method is motivated by classic PCA and the well known Locally Linear Embedding [34]. PCA suggests that the key of dimension reduction is to identify 'important' (*i.e.*, principal) components and eliminate 'unimportant' (*i.e.*, minor) ones. Thus, the dimension of feature space naturally equals to the number of principal components. Such conclusion motivates us to treat robust subspace learning and automatic subspace learning as two sides of one coin. Furthermore, LLE encodes each data point as the linear combination of its neighbourhood and assumes such reconstruction relationship is variant to different mapping spaces. The method implies that the subspace dimension equals to the size of neighbourhood for each data point. By extending this local representation into global case, PCE provides a dimension estimation for the entire data set based on the rank of the coefficient matrix $\mathbf{C}$. The contributions of this work are summarized as follows:

- We propose a robust subspace learning method (*i.e.*, PCE) to handle the gross corruptions that probably exist into training data. Different from the existing methods such as LLE and NPE, PCE formulates the corruptions into its objective function and calculates the reconstruction coefficients using a clean data set.
- Unlike previous subspace learning methods, the proposed method can automatically determine the feature dimension of the learned low-dimensional subspace. Automatic dimension determination largely reduces the efforts for finding an optimal dimension and makes PCE is more competitive in real applications.
- PCE is computational efficient, which only involves performing Singular Value Decomposition (SVD) over training data one time.

TABLE I
SOME USED NOTATIONS.

Notation	Definition
n	the number of data points
n_i	data size of the i-th subject
m	the dimension of input
m'	the dimension of feature space
s	the number of subject
r	the rank of a given matrix
$\mathbf{y}$	a given testing sample
$\mathbf{z}$	the low-dimensional feature
$\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_n]$	training data set
$\mathbf{D} = \mathbf{U\Sigma V}^T = \mathbf{U}_r\mathbf{\Sigma}_r\mathbf{V}_r^T$	full and skinny SVD of $\mathbf{D}$
$\mathbf{D}_0$	the desired clean data set
$\mathbf{E}$	the errors existing into $\mathbf{D}$
$\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_n]$	the representation of $\mathbf{D}_0$
$\sigma_i(\mathbf{C})$	the i-th singular value of $\mathbf{C}$
$\mathbf{\Theta} \in \mathbb{R}^{m\times m'}$	the projection matrix

The rest of this paper is organized as follows. Section II briefly introduces some related works. Section III presents our proposed algorithm. Section IV reports the experimental results and Section V concludes this work.

II. RELATED WORKS

A. Notations and Definitions

In the following, **lower-case bold letters** represent column vectors and **UPPER-CASE BOLD ONES** denote matrices. $\mathbf{A}^T$ and $\mathbf{A}^\dagger$ denote the transpose and pseudo-inverse of the matrix $\mathbf{A}$, respectively. $\mathbf{I}$ denotes the identity matrix.

For a given data matrix $\mathbf{D} \in \mathbb{R}^{m\times n}$, the Frobenius norm of $\mathbf{D}$ is defined as

$$\|\mathbf{D}\|_F = \sqrt{trace(\mathbf{DD}^T)} = \sqrt{\sum_{i=1}^{r}\sigma_i(\mathbf{D})}, \tag{1}$$

where $\sigma_i(\mathbf{D})$ denotes i-th singular value of $\mathbf{D}$, and r denotes the rank of $\mathbf{D}$.

The full Singular Value Decomposition (SVD) and the skinny SVD of $\mathbf{D}$ are defined as $\mathbf{D} = \mathbf{U\Sigma V}^T$ and $\mathbf{D} = \mathbf{U}_r\mathbf{\Sigma}_r\mathbf{V}_r^T$, where $\mathbf{\Sigma}$ and $\mathbf{\Sigma}_r$ are in descending order. $\mathbf{U}_r$, $\mathbf{V}_r$ and $\mathbf{\Sigma}_r$ consist of the top (*i.e.*, largest) r singular vectors and singular values of $\mathbf{D}$. TABLE I summarizes some notations used throughout the paper.

B. Locally Linear Embedding

In [15], Yan et al. have shown that most unsupervised, semi-supervised, and supervised subspace learning methods

[1] In this paper, we adopt a PCA-like definition on corruptions and errors, *i.e.*, the corruptions correspond to the minor parts of inputs.

can be unified into a framework known as graph embedding. Under this framework, subspace learning methods obtain low-dimensional features by preserving some desirable geometric relationships from a high-dimensional space into a low-dimensional one. Thus, the performance of subspace learning largely depends on the identified relationship which is usually described by a similarity graph (*i.e.*, affinity matrix). In the graph, each vertex corresponds to a data point and the edge weight denotes the similarity between two connected points. There are two popular ways to measure the similarity among data points, *i.e.*, pairwise distance such as Euclidean distance [35] and linear reconstruction coefficients introduced by Locally Linear Embedding (LLE) [34].

For a given data matrix $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_n]$, LLE solves the following problem:

$$\min_{\mathbf{c}_i} \sum_{i=1}^{n} \|\mathbf{d}_i - \mathbf{B}_i \mathbf{c}_i\|_2, \quad \text{s.t.} \sum_j c_{ij} = 1, \tag{2}$$

where $\mathbf{c}_i \in \mathbb{R}^p$ is the linear representation of $\mathbf{d}_i$ over $\mathbf{B}_i$, c_{ij} denotes the j-th entry of $\mathbf{c}_i$, and $\mathbf{B}_i \in \mathbb{R}^{m \times p}$ consists of p nearest neighbors of $\mathbf{d}_i$ that are chosen from the collection of $[\mathbf{d}_1, \ldots, \mathbf{d}_{i-1}, \mathbf{d}_{i+1}, \ldots, \mathbf{d}_n]$ in terms of Euclidean distance.

By assuming the reconstruction relationship $\mathbf{c}_i$ is invariant to ambient space, LLE obtains the low-dimensional features $\mathbf{Y} \in \mathbb{R}^{m' \times n}$ of $\mathbf{D}$ by

$$\min_{\mathbf{Y}} \|\mathbf{Y} - \mathbf{Y}\mathbf{W}\|_F^2, \quad \text{s.t. } \mathbf{Y}^T\mathbf{Y} = \mathbf{I}, \tag{3}$$

where $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_n]$ and the nonzero entries of $\mathbf{w}_i \in \mathbb{R}^n$ corresponds to $\mathbf{c}_i$.

However, LLE cannot handle the out-of-sample data that are not included into $\mathbf{D}$. To solve this problem, NPE [35] calculates the projection matrix $\mathbf{\Theta}$ instead of $\mathbf{Y}$ by replacing $\mathbf{Y}$ with $\mathbf{\Theta}^T\mathbf{D}$ into (3).

C. L1-Graph

By following the framework of LLE and NPE, Qiao et al. [13] and Cheng et al. [14] proposed SPP and L1-graph, respectively. The methods sparsely encode each data points by solving the following sparse coding problem:

$$\min_{\mathbf{c}_i} \|\mathbf{d}_i - \mathbf{D}_i \mathbf{c}_i\|_2 + \lambda \|\mathbf{c}_i\|_1, \tag{4}$$

where $\mathbf{D}_i = [\mathbf{d}_1, \ldots, \mathbf{d}_{i-1}, \mathbf{0}, \mathbf{d}_{i+1}, \ldots, \mathbf{d}_n]$.

After obtaining $\mathbf{C} \in \mathbb{R}^{n \times n}$, SPP and L1-graph embed $\mathbf{C}$ into the feature space by following NPE. The advantage of sparsity based subspace methods is that they can automatically determine the neighbourhood for each data point without the parameter of neighbourhood size. Inspired by the success of SPP and L1-graph, a number of representation based methods [36], [37], [38] have been proposed. However, these methods including L1-graph and SPP have still required specifying the dimension of feature space.

D. Robust Principal Component Analysis

RPCA [24] is proposed to improve the robustness of PCA, which solves the following optimization problem:

$$\min_{\mathbf{D}_0, \mathbf{E}} \text{rank}(\mathbf{D}_0) + \lambda \|\mathbf{E}\|_0 \quad \text{s.t. } \mathbf{D} = \mathbf{D}_0 + \mathbf{E}, \tag{5}$$

where $\lambda > 0$ is the parameter to balance the possible corruptions and the desired clean data, and $\|\cdot\|_0$ is ℓ_0-norm to count the number of nonzero entries of a given matrix or vector.

Since the rank operator and ℓ_0-norm are non-convex and discontinuous, ones usually relax them with nuclear norm and ℓ_1-norm [39]. Then, (5) is approximated by

$$\min_{\mathbf{D}_0, \mathbf{E}} \|\mathbf{D}_0\|_* + \lambda \|\mathbf{E}\|_1 \quad \text{s.t. } \mathbf{D} = \mathbf{D}_0 + \mathbf{E}, \tag{6}$$

where $\|\mathbf{D}\|_* = trace(\sqrt{\mathbf{D}^T\mathbf{D}}) = \sum_{i=1}^{r} \sigma_i(\mathbf{D})$ denotes the nuclear norm of $\mathbf{D}$ and $\sigma_i(\mathbf{D})$ is the i-th singular value of $\mathbf{D}$.

(6) can be solved by a lot of algorithms such as [40]. RPCA and its extension have achieved remarkable performance in a lot of applications, *e.g.*, image alignment [25], background subtraction[26], dimension reduction [27].

III. Principal Coefficients Embedding for Unsupervised Subspace Learning

A. Algorithm Description

In this section, we present an unsupervised algorithm for automatic subspace learning, *i.e.*, Principal Coefficients Embedding (PCE). For a given training data matrix $\mathbf{D}$, PCE removes the corruption $\mathbf{E}$ from $\mathbf{D}$ and then linearly encodes each data point using the clean data set $\mathbf{D}_0$. The proposed objective function is as follows:

$$\min_{\mathbf{C}, \mathbf{D}_0, \mathbf{E}} \frac{1}{2}\|\mathbf{C}\|_F^2 + \frac{\lambda}{2}\|\mathbf{E}\|_F^2 \quad \text{s.t. } \mathbf{D} = \mathbf{D}_0 + \mathbf{E}, \mathbf{D}_0 = \mathbf{D}_0\mathbf{C} \tag{7}$$

where $\|\cdot\|_F$ denotes the Frobenius norm of a given matrix. The Frobenius norm can improve the generalization ability. Moreover, it has shared some desirable properties with nuclear norm based representation [33] as shown in our previous works [41], [42].

To solve (7), we first consider the case of corruption-free, *i.e.*, $\mathbf{E} = \mathbf{0}$. In such setting, the objective function of PCE can be simplified as follows:

$$\min_{\mathbf{C}} \|\mathbf{C}\|_F \quad \text{s.t. } \mathbf{D} = \mathbf{D}\mathbf{C}. \tag{8}$$

Note that, $\mathbf{D}^\dagger\mathbf{D}$ is a feasible solution to $\mathbf{D} = \mathbf{D}\mathbf{C}$, where $\mathbf{D}^\dagger$ denotes the pseudo-inverse of $\mathbf{D}$. However, it remains unknown what is the unique minimizer to (8). To solve this problem, we have the following lemma:

Lemma 1. *Let $\mathbf{D} = \mathbf{U}_r\mathbf{\Delta}_r\mathbf{V}_r^T$ be the skinny SVD of the data matrix $\mathbf{D} \neq \mathbf{0}$. The unique solution to*

$$\min \|\mathbf{C}\|_F \quad \text{s.t. } \mathbf{D} = \mathbf{D}\mathbf{C}, \tag{9}$$

is given by $\mathbf{C}^* = \mathbf{V}_r\mathbf{V}_r^T$, where r is the rank of $\mathbf{D}$ and $\mathbf{D}$ is a clean data set without any corruptions.

Proof. Let $\mathbf{D} = \mathbf{U}\mathbf{\Delta}\mathbf{V}^T$ be the full SVD of $\mathbf{D}$. The pseudo-inverse of $\mathbf{D}$ is $\mathbf{D}^\dagger = \mathbf{V}_r\mathbf{\Delta}_r^{-1}\mathbf{U}_r^T$. Defining $\mathbf{V}_c$ by $\mathbf{V}^T = \begin{bmatrix} \mathbf{V}_r^T \\ \mathbf{V}_c^T \end{bmatrix}$ and $\mathbf{V}_c^T\mathbf{V}_r = \mathbf{0}$. To prove that $\mathbf{C}^* = \mathbf{V}_r\mathbf{V}_r^T$ is the unique solution to (9), two steps are required.

First, we prove that $\mathbf{C}^*$ is the minimizer to (9), *i.e.*, for any $\mathbf{X}$ satisfying $\mathbf{D} = \mathbf{D}\mathbf{X}$, it must hold that $\|\mathbf{X}\|_F \geq \|\mathbf{C}^*\|_F$.

Since for any column orthogonal matrix $\mathbf{P}$, it must hold that $\|\mathbf{PM}\|_F = \|\mathbf{M}\|_F$. Then, we have

$$\begin{aligned} \|\mathbf{X}\|_F &= \left\| \left[\begin{array}{c} \mathbf{V}_r^T \\ \mathbf{V}_c^T \end{array} \right] [\mathbf{C}^* + (\mathbf{X} - \mathbf{C}^*)] \right\|_F \\ &= \left\| \left[\begin{array}{c} \mathbf{V}_r^T \mathbf{C}^* + \mathbf{C}_r^T (\mathbf{X} - \mathbf{C}^*) \\ \mathbf{V}_c^T \mathbf{C}^* + \mathbf{V}_c^T (\mathbf{X} - \mathbf{C}^*) \end{array} \right] \right\|_F . \end{aligned} \tag{10}$$

As $\mathbf{C}^*$ satisfies $\mathbf{D} = \mathbf{DC}^*$, then $\mathbf{D}(\mathbf{X} - \mathbf{C}^*) = \mathbf{0}$, *i.e.*, $\mathbf{U}_r \mathbf{\Sigma}_r \mathbf{V}_r^T (\mathbf{X} - \mathbf{C}^*) = \mathbf{0}$. Since $\mathbf{U}_r \mathbf{\Sigma}_r \neq \mathbf{0}$, $\mathbf{V}_r^T(\mathbf{X} - \mathbf{C}^*) = \mathbf{0}$. Denote $\mathbf{\Gamma} = \mathbf{\Sigma}_r^{-1} \mathbf{U}_r^T \mathbf{D}$, then $\mathbf{C}^* = \mathbf{V}_r \mathbf{\Gamma}$. Because $\mathbf{V}_c^T \mathbf{V}_r = \mathbf{0}$, we have $\mathbf{V}_c^T \mathbf{C}^* = \mathbf{V}_c^T \mathbf{V}_r \mathbf{\Gamma} = \mathbf{0}$. Then, it follows that

$$\|\mathbf{X}\|_F = \left\| \left[\begin{array}{c} \mathbf{\Gamma} \\ \mathbf{V}_c^T (\mathbf{X} - \mathbf{C}^*) \end{array} \right] \right\|_F . \tag{11}$$

Since for any matrixes $\mathbf{M}$ and $\mathbf{N}$ with the same number of columns, it holds that

$$\left\| \left[\begin{array}{c} \mathbf{M} \\ \mathbf{N} \end{array} \right] \right\|_F^2 = \|\mathbf{M}\|_F^2 + \|\mathbf{N}\|_F^2 . \tag{12}$$

From (11) and (12), we have

$$\|\mathbf{X}\|_F^2 = \|\mathbf{\Gamma}\|_F^2 + \|\mathbf{V}_c^T (\mathbf{X} - \mathbf{C}^*)\|_F^2 , \tag{13}$$

which shows that $\|\mathbf{X}\|_F \geq \|\mathbf{\Gamma}\|_F$.

Furthermore, since

$$\|\mathbf{\Gamma}\|_F = \|\mathbf{V}_r \mathbf{\Gamma}\|_F = \|\mathbf{C}^*\|_F , \tag{14}$$

we have $\|\mathbf{X}\|_F \geq \|\mathbf{C}^*\|_F$.

Second, we prove that $\mathbf{C}^*$ is the unique solution of (9). Let $\mathbf{X}$ be another minimizer, then, $\mathbf{D} = \mathbf{DX}$ and $\|\mathbf{X}\|_F = \|\mathbf{C}^*\|_F$. From (13) and (14),

$$\|\mathbf{X}\|_F^2 = \|\mathbf{C}^*\|_F^2 + \|\mathbf{V}_c^T (\mathbf{X} - \mathbf{C}^*)\|_F^2 . \tag{15}$$

Since $\|\mathbf{X}\|_F = \|\mathbf{C}^*\|_F$, it must hold that $\|\mathbf{V}_c^T (\mathbf{X} - \mathbf{C}^*)\|_F = 0$, and then $\mathbf{V}_c^T (\mathbf{X} - \mathbf{C}^*) = \mathbf{0}$. Together with $\mathbf{V}_r^T (\mathbf{X} - \mathbf{C}^*) = \mathbf{0}$, this gives $\mathbf{V}^T (\mathbf{X} - \mathbf{C}^*) = \mathbf{0}$. Because $\mathbf{V}$ is an orthogonal matrix, it must hold that $\mathbf{X} = \mathbf{C}^*$.

The proof is complete. □

Based on Lemma 1, we consider the robust version of PCE (*i.e.*, $\mathbf{E} \neq \mathbf{0}$) and have the theorem as follows:

Theorem 1. *Let $\mathbf{D} = \mathbf{U\Sigma V}^T$ be the full SVD of $\mathbf{D} \in \mathbb{R}^{m \times n}$, where the diagonal entries of $\mathbf{\Sigma}$ are in descending order, $\mathbf{U}$ and $\mathbf{V}$ are corresponding left and right singular vectors, respectively. Suppose there exists a clean data set and errors, denoted by $\mathbf{D}_0$ and $\mathbf{E}$, respectively. The optimal $\mathbf{C}$ to*

$$\min_{\mathbf{E}, \mathbf{D}_0, \mathbf{C}} \frac{1}{2} \|\mathbf{C}\|_F^2 + \frac{\lambda}{2} \|\mathbf{E}\|_F^2 \quad \text{s.t. } \mathbf{D} = \mathbf{D}_0 + \mathbf{E},\ \mathbf{D}_0 = \mathbf{D}_0 \mathbf{C}, \tag{16}$$

is given by $\mathbf{C}^* = \mathbf{V}_k \mathbf{V}_k^T$, where λ is a balanced factor, $\mathbf{V}_k$ consists of the first k right singular vectors of $\mathbf{D}$, $k = \operatorname{argmin}_r r + \lambda \sum_{i>r} \sigma_i^2$, and σ_i denotes the i-th diagonal entry of $\mathbf{\Sigma}$.

Proof. (16) can be rewritten as

$$\min_{\mathbf{D}_0, \mathbf{C}} \frac{1}{2} \|\mathbf{C}\|_F^2 + \frac{\lambda}{2} \|\mathbf{D} - \mathbf{D}_0\|_F^2 \quad \text{s.t. } \mathbf{D}_0 = \mathbf{D}_0 \mathbf{C}, \tag{17}$$

Let $\mathbf{D}_0^* = \mathbf{U}_r \mathbf{\Sigma}_r \mathbf{V}_r^T$ be the skinny SVD of $\mathbf{D}_0$, where r is the rank of $\mathbf{D}_0$. Let $\mathbf{U}_c$ and $\mathbf{V}_c$ be the basis that orthogonal to $\mathbf{U}_r$ and $\mathbf{V}_r$, respectively. Clearly, $\mathbf{I} = \mathbf{V}_r \mathbf{V}_r^T + \mathbf{V}_c \mathbf{V}_c^T$. By Lemma 1, the representation over the clean data $\mathbf{D}_0$ is given by $\mathbf{C}^* = \mathbf{V}_r \mathbf{V}_r^T$. Next, we will bridge $\mathbf{V}_r$ and $\mathbf{V}$.

Using Lagrange method, we have

$$\mathcal{L}(\mathbf{D}_0, \mathbf{C}) = \frac{1}{2} \|\mathbf{C}\|_F^2 + \frac{\lambda}{2} \|\mathbf{D} - \mathbf{D}_0\|_F^2 + < \beta, \mathbf{D}_0 - \mathbf{D}_0 \mathbf{C} >, \tag{18}$$

where β denotes the Lagrange multiplier and the operator $< \cdot >$ denotes dot product.

Letting $\frac{\partial \mathcal{L}(\mathbf{D}_0, \mathbf{C})}{\partial \mathbf{D}_0} = 0$, it given that

$$\beta \mathbf{V}_c \mathbf{V}_c^T = \lambda \mathbf{E}. \tag{19}$$

Letting $\frac{\partial \mathcal{L}(\mathbf{D}_0, \mathbf{C})}{\partial \mathbf{C}} = 0$, it given that

$$\mathbf{V}_r \mathbf{V}_r^T = \mathbf{V}_r \mathbf{\Sigma}_r \mathbf{U}_r^T \beta. \tag{20}$$

From (20), β must be in the form of $\beta = \mathbf{U}_r \mathbf{\Sigma}_r^{-1} \mathbf{V}_r^T + \mathbf{U}_c \mathbf{M}$ for some $\mathbf{M}$. Substituting β into (19), it given that

$$\mathbf{U}_c \mathbf{M} \mathbf{V}_c \mathbf{V}_c^T = \lambda \mathbf{E}. \tag{21}$$

Thus, we have $\|\mathbf{E}\|_F^2 = \frac{1}{\lambda^2} \|\mathbf{M}\mathbf{V}_c\|_F^2$. Since $\mathbf{V}_c^T \mathbf{V}_c = \mathbf{I}$, $\|\mathbf{E}\|_F^2$ is minimized when $\mathbf{M}\mathbf{V}_c$ is a diagonal matrix and can be chosen as $\mathbf{M}\mathbf{V}_c = \mathbf{\Sigma}_c$, *i.e.*, $\mathbf{E} = \frac{1}{\lambda} \mathbf{U}_c \mathbf{\Sigma}_c \mathbf{V}_c$. Thus, the SVD of $\mathbf{D}$ could be chosen as

$$\mathbf{D} = \mathbf{U\Sigma V}^T = [\mathbf{U}_r\ \mathbf{U}_c] \left[\begin{array}{cc} \mathbf{\Sigma}_r & \mathbf{0} \\ \mathbf{0} & \frac{1}{\lambda} \mathbf{\Sigma}_c \end{array} \right] \left[\begin{array}{c} \mathbf{V}_r^T \\ \mathbf{V}_c^T \end{array} \right] . \tag{22}$$

Thus, the minimal cost of (17) is given by

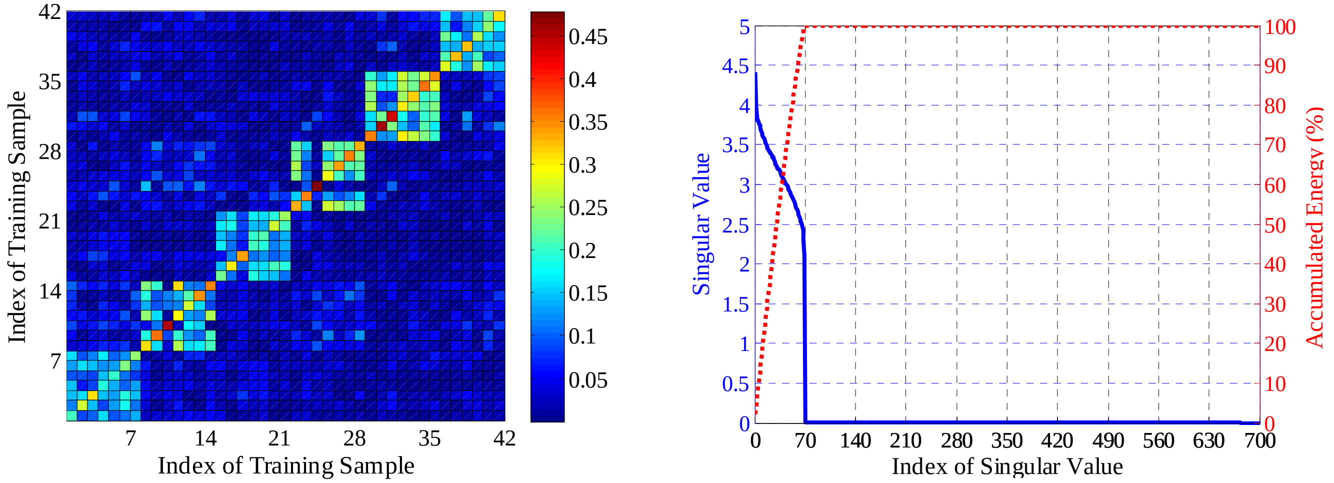
$$\begin{aligned} \mathcal{L}_{\min}(\mathbf{D}_0^*, \mathbf{C}^*) &= \frac{1}{2} \|\mathbf{V}_r \mathbf{V}_r^T\|_F^2 + \frac{\lambda}{2} \|\frac{1}{\lambda} \mathbf{\Sigma}_c\|_F^2 \\ &= \frac{1}{2} r + \frac{\lambda}{2} \sum_{i=r+1}^{\min\{m,n\}} \sigma_i^2 , \end{aligned} \tag{23}$$

where σ_i is the i-th largest singular value of $\mathbf{D}$. Let k be the optimal r to (23), then we have $k = \operatorname{argmin}_r r + \lambda \sum_{i>r} \sigma_i^2$.

The proof is complete. □

Theorem 1 shows that the skinny SVD of $\mathbf{D}$ is automatically separated into two parts, the top and the bottom one correspond to a desired clean data $\mathbf{D}_0$ and the possible corruptions $\mathbf{E}$, respectively. Such PCA-like result provides a good explanation toward the robustness of our method. Furthermore, PCE estimates the dimension of the feature space with the rank of the coefficient matrix $\mathbf{C}$ (*i.e.*, k) which is further determined by the parameter λ. Because λ actually measures the ratio between the clean data $\mathbf{D}_0$ and $\mathbf{E}$, this again verifies our motivation, *i.e.*, the key of automatic subspace learning is removing possible errors from inputs.

Fig. 1 gives an example to show the effectiveness of PCE. We carried out experiment using 700 clean AR facial images [43] as training data that distribute over 100 individuals. Fig. 1(a) shows the coefficient matrix $\mathbf{C}^*$ obtained by PCE. One can find that the matrix is approximately block-diagonal, *i.e.*, $c_{ij} \neq 0$ if and only if the corresponding points $\mathbf{d}_i$ and $\mathbf{d}_j$ belong to the same class. The block-diagonal property of $\mathbf{C}^*$ not only guarantees the discrimination of $\mathbf{C}^*$, but also directly



(a) The coefficient matrix $\mathbf{C}^*$ obtained by PCE.

(b) Singular values of $\mathbf{C}^*$

Fig. 1. An illustration using 700 AR facial images. (a) PCE can obtain a block-diagonal affinity matrix, which is benefit to classification. (b) The intrinsic dimension of the used data set is exactly 69, *i.e.*, $m' = k = 69$. This result is obtained without truncating the trivial singular values like PCA. Dotted line denotes the accumulated energy of the first k singular value.

provides a feasible way to estimate the dimension of feature space, *i.e.*, the feature dimension can be estimated by the rank of $\mathbf{C}^*$. To verify the effectiveness of this dimension estimation, we perform SVD over $\mathbf{C}^*$ and show the singular values of $\mathbf{C}^*$ in Fig. 1(b). One can find that only the first 69 singulars values are nonzero. In other words, the intrinsic dimension of the entire data set is 69 and the first 69 singular values preserve 100% energy. It should be pointed out that, PCE does not set a parameter to truncate the trivial singular values like PCA and PCA-like methods [17], which incorporates all energy into a small number of dimension.

After obtaining the coefficient matrix $\mathbf{C}^*$ and the feature dimension k, PCE builds a similarity graph and embeds it into a k-dimensional space by following NPE [11], *i.e.*,

$$\min_{\mathbf{\Theta}} \frac{1}{2}\|\mathbf{\Theta}^T\mathbf{D} - \mathbf{\Theta}^T\mathbf{D}\mathbf{A}\|_F^2, \quad \text{s.t. } \mathbf{D}^T\mathbf{\Theta}\mathbf{\Theta}^T\mathbf{D} = \mathbf{I}, \tag{24}$$

where $\mathbf{\Theta} \in \mathbb{R}^{m\times k}$ denotes the projection matrix. Algorithm 1 summarizes our algorithm.

B. Computational Complexity Analysis

For a training data set $\mathbf{D} \in \mathbb{R}^{m\times n}$, PCE performs the skinny SVD over $\mathbf{D}$ in $O(m^2n + mn^2 + n^3)$. However, a number of fast SVD methods can speed up this procedure. For example, the complexity can be reduced to $O(mnr)$ by Brand's method [44], where r is the rank of $\mathbf{D}$. Moreover, PCE estimates the feature dimension k in $O(rlogr)$ and solves a sparse generalized eigenvector problem in $O(mn + mn^2)$ with Lanczos eigensolver. Putting everything together, the time complexity of PCE is $O(mn + mn^2)$ due to $r \ll \min(m, n)$.

IV. Experiments and Results

In this section, we reported the performance of PCE and six state-of-the-art unsupervised feature extraction methods including Eigenfaces [10], Locality Preserving Projections (LPP) [35], [12], neighbourhood Preserving Embedding (NPE) [11], L1-graph [14], Non-negative Matrix Factorization (NMF) [45], and RPCA [24].

Algorithm 1. Automatic Subspace Learning via Principal Coefficients Embedding

Input: A collection of training data points $\mathbf{D} = \{\mathbf{d}_i\}$ sampled from a union of linear subspaces and the balanced parameter $\lambda > 0$.

1: Perform the full SVD or skinny SVD on $\mathbf{D}$, *i.e.*, $\mathbf{D} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, and get the $\mathbf{C} = \mathbf{V}_k\mathbf{V}_k^T$, where $\mathbf{V}_k$ consists of k column vector of $\mathbf{V}$ corresponding to k largest singular values, where $k = \text{argmin}_r r + \lambda \sum_{i>r} \sigma_i^2(\mathbf{D})$ and $\sigma_i(\mathbf{D})$ is the i-th singular value of $\mathbf{D}$.

2: Construct a similarity graph by $\mathbf{A} = \mathbf{C}$ and normalize each column of $\mathbf{A}$ to have a unit ℓ_2-norm.

3: Embed $\mathbf{A}$ into a k-dimensional space and get the projection matrix $\mathbf{\Theta} \in \mathbb{R}^{m\times k}$ that consists of the eigenvectors corresponding to the k largest eigenvalues of the following generalized eigenvector problem:

$$\mathbf{D}(\mathbf{I} - \mathbf{A})(\mathbf{I} - \mathbf{A})^T\mathbf{D}^T\mathbf{\Theta} = \sigma\mathbf{D}\mathbf{D}^T\mathbf{\Theta}. \tag{25}$$

Output: The projection matrix $\mathbf{\Theta}$. For any data point $\mathbf{y} \in span\{\mathbf{D}\}$, its low-dimensional representation can be obtained by $\mathbf{z} = \mathbf{\Theta}^T\mathbf{y}$.

A. Experimental Setting and Data Sets

We implemented a fast version of L1-graph by using Homotopy algorithm [46] to solve the ℓ_1-minimization problem. According to [47], Homotopy is one of the most competitive ℓ_1-optimization algorithms in terms of accuracy, robustness, and convergence speed. For RPCA, we adopted the accelerated proximal gradient method with partial SVD [48] which has achieved a good balance between computation speed and reconstruction error. As mentioned above, RPCA cannot obtain the projection matrix for subspace learning. For fair com-

TABLE II
THE USED DATABASES. s AND n_i DENOTE THE NUMBER OF SUBJECT AND THE NUMBER OF IMAGES FOR EACH GROUP.

Databases	s	n_i	Original Size	Cropped Size
AR	100	26	165 × 120	55 × 40
ExYaleB	38	58	192 × 168	54 × 48
MPIE-S1	249	14	100 × 82	55 × 40
MPIE-S2	203	10	100 × 82	55 × 40
MPIE-S3	164	10	100 × 82	55 × 40
MPIE-S4	176	10	100 × 82	55 × 40
COIL100	100	10	128 × 128	64 × 64

parison, we incorporated Eigenfaces with RPCA (denoted by RPCA+PCA) to obtain the low-dimensional features of the inputs. Unless otherwise specified, we assigned $m' = 300$ for all the tested methods except PCE which automatically determines the value of m'.

In our experiments, we evaluated the performance of these subspace learning algorithms with three classifiers, *i.e.*, Sparse Representation based Classification (SRC) [49], Support Vector Machine (SVM) with linear kernel [50], and the Nearest Neighbor classifier (NN). We adopted the cross-validation method to determine the optimal parameters for all the methods and reported the mean and standard deviation of classification accuracy and time costs.

We used seven image data sets including AR facial database [43], Expended Yale Database B (ExYaleB) [51], four sessions of Multiple PIE (MPIE) [52], and COIL100 objects database [53].

The used AR data set contains 2600 samples from 50 male and 50 female subjects, of which 1400 samples are clean images, 600 samples are disguised by sunglasses, and the remaining 600 samples are disguised by scarves. ExYaleB contains 2414 frontal-face images of 38 subjects, and we use the first 58 samples of each subject. MPIE contains the facial images captured in four sessions. In the experiments, all the frontal faces with 14 illuminations[2] are investigated. For computational efficiency, we downsized all the data sets from the original size to smaller one. TABLE II provides an overview of the used data sets.

B. The Influence of the Parameter

PCE uses the parameter λ to measure the possible corruptions and estimate the feature dimension m'. To investigate the influence of λ, we increased the value of λ from 1 to 99 with an interval of 2 by performing experiment on a subset of AR database. The used data set includes 1400 clean images over 100 individuals. In the experiment, we randomly divided the data into two parts with equal size for training and testing.

Fig. 2 shows that: 1) while λ increases from 13 to 39, the recognition accuracy of PCE almost remains unchanged, which ranges from 93.86% to 95.29%; 2) with increasing λ, the rank of the obtained coefficient matrix (*i.e.*, k) increases from 10 to 202. However, a larger k cannot make the recognition rate higher.

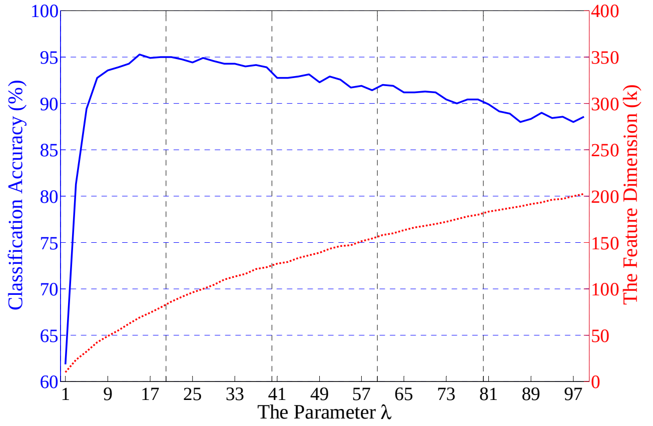


Fig. 2. The influence of the parameter λ of PCE, where the NN classifier and 1400 AR clean images are used.

To further show the effectiveness of our dimension determination method, we investigated the performance of PCE by manually specifying $m' = 300$, denoted by PCE2. We carried out the experiments on ExYaleB by choosing 40 samples from each subject as training data and using the rests for testing. TABLE III reports the result from which we can find that:

- the automatic version of our method, *i.e.*, PCE, performs competitive to PCE2 which manually set $m' = 300$. This shows that our dimension estimation method can accurately estimate the feature dimension.
- both PCE and PCE2 outperform the other methods by a considerable performance margin. For example, PCE is 3.68% at least higher than the second best method when the NN classifier is used.
- PCE is also the one of the fastest algorithms, which is remarkably faster than L1-graph, NMF, and RPCA+PCA.

C. Performance with Increasing Training Data and Feature Dimension

In this section, we examined the performance of PCE with increasing training samples and increasing feature dimension. In the first test, we randomly sampled n_i clean AR images from each subject for training and used the rest for testing. Beside RPCA+PCA, we also reported the performance of RPCA without dimension reduction.

In the second test, we randomly chose a half of images from ExYaleB for training and used the rest for testing. We reported the recognition rate of the NN classifier with the first m' features extracted by all the tested subspace learning methods, where m' increases from 1 to 600 with an interval of 10.

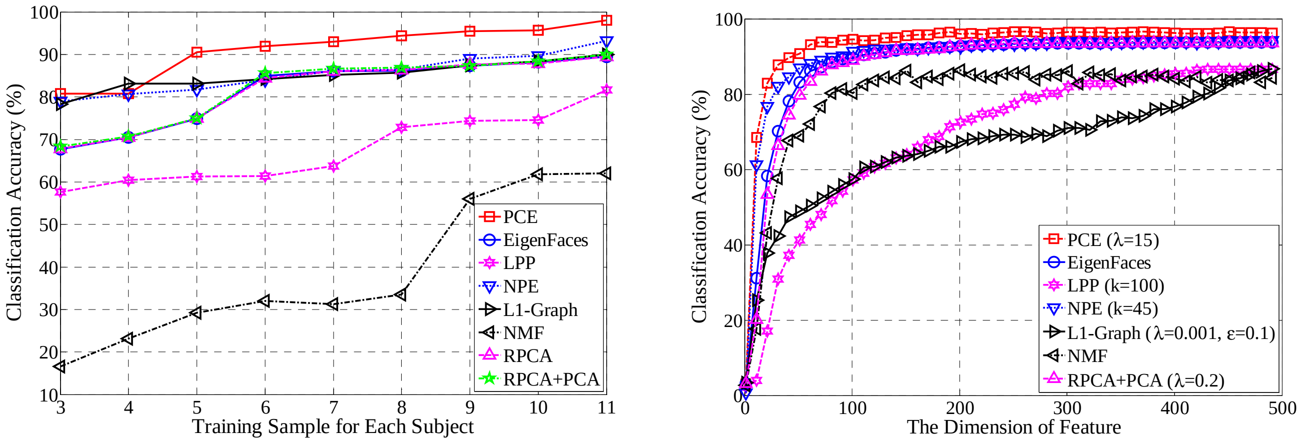
From Fig. 3, we can conclude:

- PCE performs well even though only a few of training samples are available. Its accuracy is about 90% when $n_i = 5$, whereas the second best method achieves the same accuracy when $n_i = 9$.
- RPCA and RPCA+PCA perform very close, however, RPCA+PCA is more efficient than RPCA.

[2] illuminations: 0,1,3,4,6,7,8,11,13,14,16,17,18,19.

TABLE III
PERFORMANCE COMPARISON AMONG DIFFERENT ALGORITHMS USING **EXYALEB**, WHERE TRAINING DATA AND TESTING DATA CONSIST OF 1520 AND 380 SAMPLES, RESPECTIVELY. NOTE THAT, PCE, EIGENFACES, AND NMF HAVE ONLY ONE PARAMETER. PCE NEEDS SPECIFYING THE PARAMETER λ, AND EIGENFACES AND NMF HAVE TO SET THE FEATURE DIMENSION. ALL METHODS EXCEPT PCE EXTRACT 300 FEATURES FOR CLASSIFICATION. 'PARA.' INDICATES THE TUNED PARAMETERS USING VALIDATION DATA SET. THE SECOND PARAMETER OF PCE DENOTES m' (*i.e.*, k) WHICH IS AUTOMATICALLY CALCULATED VIA THEOREM 1.

Classifiers	SRC			SVM			NN		
Algorithms	Accuracy	Time (s)	Para.	Accuracy	Time (s)	Para.	Accuracy	Time (s)	Para.
PCE	**96.90±0.74**	23.50±2.36	10, 169	**98.93±0.18**	7.44±0.37	50, 331	**97.03±0.57**	6.96±0.71	5, 118
PCE2	**96.92±0.59**	28.02±2.84	16.00	98.20±0.43	8.07±0.67	26.00	96.86±0.57	7.89±0.88	19.00
Eigenfaces	95.32±0.80	27.79±0.22	-	95.53±0.85	5.65±0.14	-	82.53±1.70	4.97±0.14	-
LPP	83.87±6.59	17.20±0.71	9.00	87.92±9.12	7.40±0.12	2.00	79.97±1.36	7.18±0.19	3.00
NPE	90.47±15.72	37.80±0.45	50.00	82.50±8.74	27.57±0.24	47.00	93.35±0.53	28.37±0.30	49.00
L1-graph	91.29±0.60	633.95±47.94	1e-2,1e-1	82.08±1.66	870.04±61.01	1e-3,1e-3	89.75±0.70	988.27±74.98	1e-2,1e-3
NMF	87.54±1.15	137.46±6.26	-	91.59±1.09	19.39±0.36	-	72.11±1.44	11.13±0.03	-
RPCA+PCA	95.88±0.56	497.48±32.72	0.30	95.79±1.02	466.17±35.85	0.10	82.57±1.18	466.11±42.70	0.20



(a) (b)

Fig. 3. (a) The performance of the evaluated subspace learning methods with the NN classifier on AR images. (b) The recognition rates of the NN classifier with different subspace learning methods on ExYaleB. Note that, PCE does not automatically determine the feature dimension in the experiment of performance versus increasing feature dimension.

- Fig. 3(b) shows that PCE consistently outperforms the other methods. This benefits an advantage of PCE, *i.e.*, PCE obtains a more compact representation which can use a few of variables to represent the entire data.

D. Subspace Learning on Clean Images

In this section, we performed the experiments using MPIE and COIL100. For each data set, we split it into two parts with equal size. As did in the above experiments, we set $m' = 300$ for all the tested methods except PCE. TABLEs IV–VIII report the results, from which one can find that:

- with three classifiers, PCE outperforms the other investigated approaches on these five data sets by a considerable performance margin. For example, the recognition rates of PCE with these three classifiers are 6.59%, 5.83%, and 7.90% at least higher than the rates of the second best subspace learning method on MPIE-S1.
- PCE is more stable than other tested methods. Although SRC generally outperforms SVM and NN with the same feature, such superiority is not distinct for PCE. For example, SRC gives an accuracy improvement of 1.02% over NN to PCE on MPIE-S4. However, the corresponding improvement to RPCA+PCA is about 49.50%.
- PCE achieves the best results in all the tests, while using the least time to perform dimension reduction and classification. PCE is more efficient than L1-graph, NMF, and RPCA+PCA, and only Eigenfaces, LPP, and NPE can be competitive to it in computational efficiency.

E. Subspace Learning on Corrupted Facial Images

In this section, we investigated the robustness of PCE against two corruptions using ExYaleB and the NN classifier. The corruptions includes white Gaussian noise (additive noise) and random pixel corruption (non-additive noise) [49].

In our experiments, we chosen a half of images (29 images per subject) to corrupt using these two noises. Specifically, we added white Gaussian noise into the sampled data $\mathbf{d}$ via $\tilde{\mathbf{d}} = \mathbf{d} + \rho\mathbf{n}$, where $\tilde{\mathbf{d}} \in [0\ 255]$, ρ is the corruption ratio, and $\mathbf{n}$ is the noise following the standard normal distribution. For random pixel corruption, we replaced the value of a percentage of pixels randomly selected from the image with the values following a uniform distribution over $[0,\ p_{max}]$, where p_{max} is the largest pixel value of $\mathbf{d}$. After adding the noises into

TABLE IV
PERFORMANCE COMPARISON AMONG DIFFERENT ALGORITHMS USING **THE FIRST SESSION OF MPIE (MPIE-S1)**. ALL METHODS EXCEPT PCE EXTRACT 300 FEATURES FOR CLASSIFICATION.

Classifiers	SRC			SVM			NN		
Algorithms	Accuracy	Time (s)	Para.	Accuracy	Time (s)	Para.	Accuracy	Time (s)	Para.
PCE	**99.27±0.32**	51.96±0.29	55.00	**96.56±1.23**	14.38±0.52	45.00	**97.72±0.55**	13.21±0.59	40.00
Eigenfaces	92.64±0.56	90.64±0.73	-	90.73±1.81	12.87±0.20	-	55.03±0.93	6.21±0.22	-
LPP	81.84±0.94	30.58±2.60	10.00	70.16±0.07	7.38±0.54	55.00	71.31±2.39	4.85±0.39	4.00
NPE	80.56±0.41	58.95±0.77	29.00	80.25±0.15	36.38±0.55	43.00	77.71±1.65	36.19±0.38	49.00
L1-graph	80.36±0.17	3856.69±280.16	1e-1,1e-1	86.79±1.62	5726.08±444.82	1e-6,1e-5	89.82±1.44	8185.55±503.80	1e-6,1e-4
NMF	65.18±0.87	520.94±6.27	-	66.42±1.66	121.89±0.58	-	41.78±1.18	11.03±0.00	-
RPCA+PCA	92.68±0.57	1755.27±490.99	0.10	90.51±1.26	1497.13±329.00	0.30	54.95±1.38	1557.33±358.93	0.10

TABLE V
PERFORMANCE COMPARISON AMONG DIFFERENT ALGORITHMS USING **THE SECOND SESSION OF MPIE (MPIE-S2)**. ALL METHODS EXCEPT PCE EXTRACT 300 FEATURES FOR CLASSIFICATION.

Classifiers	SRC			SVM			NN		
Algorithms	Accuracy	Time (s)	Para.	Accuracy	Time (s)	Para.	Accuracy	Time (s)	Para.
PCE	**93.87±0.82**	29.00±0.36	10.00	**92.63±0.95**	4.97±0.11	20.00	**93.18±0.87**	4.14±0.14	5.00
Eigenfaces	64.36±2.42	81.71±14.99	-	51.72±2.81	0.50±0.11	-	30.86±1.44	0.36±0.06	-
LPP	59.62±2.33	36.69±7.64	2.00	34.28±2.53	2.73±0.60	2.00	62.64±2.20	2.73±0.84	3.00
NPE	84.65±0.77	33.03±1.51	41.00	64.66±3.03	12.45±0.30	27.00	85.56±0.92	12.24±0.24	49.00
L1-graph	47.67±3.09	874.91±53.69	1e-3,1e-3	65.41±1.69	657.69±53.51	1e-3,1e-3	74.15±1.67	703.54±37.97	1e-2,1e-3
NMF	81.88±1.31	323.93±8.70	-	83.19±1.47	46.72±1.22	-	57.21±1.38	26.01±0.01	-
RPCA+PCA	91.18±1.11	401.62±7.46	0.20	91.18±1.11	401.62±7.46	0.20	67.80±1.93	366.50±8.78	0.10

TABLE VI
PERFORMANCE COMPARISON AMONG DIFFERENT ALGORITHMS USING **THE THIRD SESSION OF MPIE (MPIE-S3)**. ALL METHODS EXCEPT PCE EXTRACT 300 FEATURES FOR CLASSIFICATION.

Classifiers	SRC			SVM			NN		
Algorithms	Accuracy	Time (s)	Para.	Accuracy	Time (s)	Para.	Accuracy	Time (s)	Para.
PCE	**97.79±0.81**	13.14±0.19	85.00	**95.37±1.82**	2.74±0.08	90.00	**94.04±0.84**	2.29±0.04	70.00
Eigenfaces	88.04±0.70	29.51±0.36	-	80.99±2.28	2.01±0.05	-	37.96±1.18	0.94±0.05	-
LPP	78.73±2.04	28.61±4.62	40.00	60.44±2.49	1.61±0.25	3.00	65.96±2.49	1.03±0.13	75.00
NPE	77.83±3.14	25.79±1.02	46.00	72.29±0.99	7.56±0.07	7.00	79.18±2.38	7.06±0.09	48.00
L1-graph	70.40±0.22	1315.37±192.65	1e-1,1e-5	79.28±2.54	1309.27±193.38	1e-3,1e-3	89.40±2.80	1539.26±226.57	1e-3,1e-3
NMF	60.94±0.80	90.64±0.91	-	51.34±1.68	40.04±0.37	-	39.89±1.04	4.28±0.01	-
RPCA+PCA	88.49±2.17	630.08±88.89	0.10	81.02±2.52	491.36±26.75	0.30	37.85±0.83	481.87±25.01	0.30

the images, we randomly divide the data into training and testing sets. In other words, both training data and testing data probably contains corruptions. TABLE IX shows that:

- PCE is more robust than the other approaches. When 10% pixels are randomly corrupted, the accuracy of PCE is at least 9.46% higher than that of the other methods.
- with the increase of level of noise, the dominance of PCE is further strengthen. For example, the improvement in accuracy of PCE increases from 9.46% to 23.23% when more pixels are randomly corrupted.

F. Subspace Learning on Disguised Facial Images

Besides the above tests on the robustness to corruptions, we also investigated the robustness to real disguises. TABLEs X–XI reports results on two subsets of AR database. The first subset contains 600 clean images and 600 images disguised with sunglasses (occlusion rate is about 20%), and the second one includes 600 clean images and 600 images disguised by scarves (occlusion rate is about 40%). Like the above experiment, both training data and testing data will contains the disguised images. From the results, one can conclude that:

- PCE significantly outperforms the other tested methods. When the images are disguised by sunglasses, the recognition rates of PCE with SRC, SVM, and NN are 5.88%, 23.03%, and 11.75% higher than these of the second best method. With respect to the images with scarves, the corresponding improvements over the second best method are 12.17%, 21.30%, and 17.64%.
- PCE is still the most computationally efficient method. Considering SRC is used, the time of speedup of PCE ranges from 2.27 (over NPE) to 497.16 (over L1-graph) on the images with sunglasses and from 2.17 (over NPE) to 484.94 (over L1-graph) on the images with scarves.

V. CONCLUSION

Based on a key assumption, *i.e.*, dimension determination can be realized by removing the corruptions from inputs, this paper has proposed a novel unsupervised subspace learning method, called Principal Coefficients Embedding (PCE). Unlike existing subspace learning methods, PCE automatically determines the dimension of feature space without specifying the parameter. Experimental results on several popular image

TABLE VII
PERFORMANCE COMPARISON AMONG DIFFERENT ALGORITHMS USING **THE FOURTH SESSION OF MPIE (MPIE-S4)**. ALL METHODS EXCEPT PCE EXTRACT 300 FEATURES FOR CLASSIFICATION.

Classifiers	SRC			SVM			NN		
Algorithms	Accuracy	Time (s)	Para.	Accuracy	Time (s)	Para.	Accuracy	Time (s)	Para.
PCE	**98.36±0.41**	14.07±0.31	40.00	**90.55±1.02**	3.04±0.12	40.00	**97.34±0.78**	2.73±0.09	85.00
Eigenfaces	92.05±1.37	32.43±0.32	-	82.18±3.88	2.34±0.05	-	43.74±1.17	1.12±0.05	-
LPP	64.67±2.52	27.38±1.57	3.00	61.47±1.12	1.94±0.20	2.00	73.69±2.68	1.11±0.17	2.00
NPE	84.74±1.50	30.45±1.28	46.00	63.80±1.56	9.87±0.49	49.00	87.30±1.10	8.54±0.36	45.00
L1-graph	70.45±0.31	1928.24±212.21	1e-3,1e-3	84.67±2.46	1825.09±197.62	1e-3,1e-3	93.56±1.13	1767.57±156.61	1e-3,1e-3
NMF	69.41±1.73	98.91±1.37	-	53.48±2.07	47.26±0.44	-	25.47±1.40	4.85±0.00	-
RPCA+PCA	93.16±1.17	682.27±39.20	0.30	84.45±3.02	535.31±19.08	0.10	43.66±0.63	514.51±20.82	0.10

TABLE VIII
PERFORMANCE COMPARISON AMONG DIFFERENT ALGORITHMS USING **COIL100**. ALL METHODS EXCEPT PCE EXTRACT 300 FEATURES FOR CLASSIFICATION.

Classifiers	SRC			SVM			NN		
Algorithms	Accuracy	Time (s)	Para.	Accuracy	Time (s)	Para.	Accuracy	Time (s)	Para.
PCE	**59.60±1.94**	12.25±0.32	5.00	**53.00±1.22**	1.36±0.01	25.00	**57.40±1.83**	1.15±0.03	5.00
Eigenfaces	57.40±1.67	12.97±0.25	-	44.40±2.21	1.04±0.06	-	54.76±1.14	0.67±0.06	-
LPP	45.86±1.51	13.22±0.54	60.00	30.20±3.08	0.80±0.11	2.00	41.10±2.15	0.63±0.02	90.00
NPE	47.72±2.25	15.30±0.28	43.00	32.78±2.90	5.33±0.08	36.00	44.88±2.12	6.81±0.03	49.00
L1-graph	45.16±1.83	960.80±123.43	1e-2,1e-4	39.42±2.81	801.73±147.83	1e-3,1e-3	38.06±1.96	664.92±93.75	1e-1,1e-5
NMF	51.42±2.17	76.05±1.21	-	41.74±2.05	32.81±0.18	-	56.82±1.46	6.47±0.00	-
RPCA+PCA	58.04±0.90	244.92±50.17	0.30	45.52±2.70	229.54±51.06	0.20	56.48±1.32	227.27±52.66	0.10

TABLE IX
PERFORMANCE OF DIFFERENT SUBSPACE LEARNING ALGORITHMS WITH THE NN CLASSIFIER USING **THE CORRUPTED EXYALEB**. ALL METHODS EXCEPT PCE EXTRACT 300 FEATURES FOR CLASSIFICATION. RPC IS THE SHORT FOR RANDOM PIXEL CORRUPTION. THE NUMBER IN THE PARENTHESES DENOTES THE LEVEL OF CORRUPTION.

Corruptions	Gaussian (10%)		Gaussian (30%)		RPC (10%)		RPC (30%)	
Algorithms	Accuracy	Para.	Accuracy	Para.	Accuracy	Para.	Accuracy	Para.
PCE	**95.05±0.63**	10.00	**93.18±0.87**	5.00	**90.12±0.98**	10.00	**83.48±1.04**	10.00
Eigenfaces	41.69±2.01	-	30.86±1.44	-	30.35±2.05	-	25.37±1.56	-
LPP	76.94±0.75	2.00	62.64±2.20	3.00	55.86±1.27	2.00	42.76±1.53	2.00
NPE	91.54±0.76	49.00	85.56±0.92	49.00	80.66±0.86	49.00	60.25±1.64	43.00
L1-graph	87.36±0.81	1e-3,1e-4	74.15±1.67	1e-2,1e-3	71.63±0.90	1e-3,1e-4	55.02±2.07	1e-4,1e-4
NMF	67.42±1.41	-	57.21±1.38	-	60.57±1.88	-	46.13±1.41	-
RPCA+PCA	76.26±1.12	0.20	67.80±1.93	0.10	64.56±0.67	0.10	52.12±1.34	0.10

databases have validated that our PCE shows good performance with respect to additive noise, non-additive noise, and partial disguised images.

The work would be extended or improved from the following aspects. First, the paper only considers one category of image recognition, *i.e.*, image identification. In the future, PCE can be extended to handle the other category of image recognition, *i.e.*, image verification. Second, supposing the entire or limited training data set are labelled, one can develop supervised or semi-supervised PCE by incorporating the label information into our model. Third, PCE can be extended to handle outliers by enforcing $\ell_{2,1}$-norm over the errors term.

REFERENCES

[1] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of eugenics*, vol. 7, no. 2, pp. 179–188, 1936.

[2] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov, "Neighbourhood components analysis," in *Advances in Neural Information Processing Systems*. MIT Press, 2004, pp. 513–520.

[3] H.-T. Chen, H.-W. Chang, and T.-L. Liu, "Local discriminant embedding and its variants," in *EEE Conference on Computer Vision and Pattern Recognition*, vol. 2, June 2005, pp. 846–853 vol. 2.

[4] Z. Cui, W. Li, D. Xu, S. Shan, and X. Chen, "Fusing robust face region descriptors via multiple metric learning for face recognition in the wild," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2013, pp. 3554–3561.

[5] J. Lu, X. Zhou, Y.-P. Tan, Y. Shang, and J. Zhou, "Neighborhood repulsed metric learning for kinship verification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 2, pp. 331–345, 2014.

[6] S. Nikitidis, A. Tefas, and I. Pitas, "Maximum margin projection subspace learning for visual data analysis," *IEEE Transactions on Image Processing*, vol. 23, no. 10, pp. 4413–4425, Oct 2014.

[7] D. Cai, X. He, and J. Han, "Semi-supervised discriminant analysis," in *International Conference on Computer Vision*, Oct 2007, pp. 1–7.

[8] S. Yan and H. Wang, "Semi-supervised learning by sparse representation." in *SDM*. SIAM, 2009, pp. 792–801.

[9] X. Shi, Z. Guo, Z. Lai, Y. Yang, Z. Bao, and D. Zhang, "A framework of joint graph embedding and sparse regression for dimensionality reduction," *IEEE Transactions on Image Processing*, vol. 24, no. 4, pp. 1341–1355, 2015.

[10] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

[11] X. He, D. Cai, S. Yan, and H.-J. Zhang, "Neighborhood preserving embedding," in *IEEE International Conference on Computer Vision*, vol. 2, Oct 2005, pp. 1208–1213 Vol. 2.

[12] X. He, S. Yan, Y. Hu, P. Niyogi, and H.-J. Zhang, "Face recognition using laplacianfaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 3, pp. 328–340, 2005.

TABLE X
PERFORMANCE COMPARISON AMONG DIFFERENT ALGORITHMS USING **THE AR IMAGES DISGUISED BY SUNGLASSES**. ALL METHODS EXCEPT PCE EXTRACT 300 FEATURES FOR CLASSIFICATION.

Classifiers	SRC			SVM			NN		
Algorithms	Accuracy	Time (s)	Para.	Accuracy	Time (s)	Para.	Accuracy	Time (s)	Para.
PCE	**83.88±1.38**	8.73±0.90	90.00	**87.80±1.57**	0.90±0.10	90.00	**68.58±1.96**	0.71±0.11	60.00
Eigenfaces	72.87±1.99	45.48±5.18	-	64.77±2.96	1.62±0.40	-	36.42±1.69	0.78±0.19	-
LPP	51.73±2.77	44.20±8.25	95.00	44.88±1.93	1.60±0.57	2.00	37.37±2.19	1.12±0.30	85.00
NPE	78.00±2.27	19.84±0.51	47.00	49.17±3.33	4.30±0.04	47.00	56.83±1.83	4.16±0.04	49.00
L1-graph	52.00±1.42	4340.22±573.64	1e-4,1e-4	48.53±2.06	3899.81±487.89	1e-4,1e-4	49.28±2.68	4189.73±431.98	1e-4,1e-4
NMF	47.87±2.64	108.46±2.98	-	43.05±2.39	24.34±0.81	-	31.35±2.04	8.01±0.01	-
RPCA+PCA	72.07±2.30	1227.08±519.27	0.10	63.70±3.74	1044.46±462.33	0.20	36.93±0.90	965.76±385.19	0.10

TABLE XI
PERFORMANCE COMPARISON AMONG DIFFERENT ALGORITHMS USING **THE AR IMAGES DISGUISED BY SCARVES**. ALL METHODS EXCEPT PCE EXTRACT 300 FEATURES FOR CLASSIFICATION.

Classifiers	SRC			SVM			NN		
Algorithms	Accuracy	Time (s)	Para.	Accuracy	Time (s)	Para.	Accuracy	Time (s)	Para.
PCE	**83.57±1.16**	8.95±0.96	65.00	**87.70±1.62**	0.90±0.11	90.00	**66.92±1.75**	0.69±0.10	70.00
Eigenfaces	69.32±2.58	36.18±3.97	-	63.93±2.84	1.42±0.34	-	30.58±1.27	0.71±0.19	-
LPP	49.48±1.79	30.80±6.29	2.00	43.07±1.80	1.34±0.39	2.00	33.70±1.70	0.85±0.21	90.00
NPE	62.75±2.16	19.44±0.62	47.00	58.23±2.75	4.40±0.03	49.00	54.33±2.37	4.09±0.03	49.00
L1-graph	49.65±1.42	4340.22±453.64	1e-3,1e-4	48.53±2.06	5381.96±467.89	1e-4,1e-4	49.28±2.68	5189.73±411.98	1e-4,1e-4
NMF	47.17±2.18	109.33±2.22	-	40.55±2.20	23.67±0.92	-	24.58±1.88	5.82±0.01	-
RPCA+PCA	71.40±2.67	241.45±4.39	0.10	66.40±2.62	193.67±7.76	0.10	32.27±1.46	195.21±8.01	0.20

[13] L. S. Qiao, S. C. Chen, and X. Y. Tan, "Sparsity preserving projections with applications to face recognition," *Pattern Recognition*, vol. 43, no. 1, pp. 331–341, 2010.

[14] B. Cheng, J. Yang, S. Yan, Y. Fu, and T. Huang, "Learning with L1-graph for image analysis," *IEEE Transactions on Image Processing*, vol. 19, no. 4, pp. 858–866, 2010.

[15] S. C. Yan, D. Xu, B. Y. Zhang, H. J. Zhang, Q. Yang, and S. Lin, "Graph embedding and extensions: A general framework for dimensionality reduction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 1, pp. 40–51, 2007.

[16] M. Polito and P. Perona, "Grouping and dimensionality reduction by locally linear embedding," in *Advances in Neural Information Processing Systems*. MIT Press, 2001, pp. 1255–1262.

[17] Y. Shuicheng, J. Liu, X. Tang, and T. Huang, "A parameter-free framework for general supervised subspace learning," *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 1, pp. 69–76, March 2007.

[18] B. Kégl, "Intrinsic dimension estimation using packing numbers," in *Advances in neural information processing systems*, 2002, pp. 681–688.

[19] F. Camastra and A. Vinciarelli, "Estimating the intrinsic dimension of data with a fractal-based method," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 10, pp. 1404–1407, Oct 2002.

[20] D. Mo and S. Huang, "Fractal-based intrinsic dimension estimation and its application in dimensionality reduction," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 1, pp. 59–71, Jan 2012.

[21] P. Mordohai and G. Medioni, "Dimensionality estimation, manifold learning and function approximation using tensor voting," *The Journal of Machine Learning Research*, vol. 11, pp. 411–450, 2010.

[22] K. M. Carter, R. Raich, and A. Hero, "On local intrinsic dimension estimation and its applications," *IEEE Transactions on Signal Processing*, vol. 58, no. 2, pp. 650–663, Feb 2010.

[23] L. K. Saul and S. T. Roweis, "Think globally, fit locally: Unsupervised learning of low dimensional manifolds," *The Journal of Machine Learning Research*, vol. 4, pp. 119–155, Dec. 2003.

[24] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *Journal of the ACM (JACM)*, vol. 58, no. 3, p. 11, 2011.

[25] Y. Peng, A. Ganesh, J. Wright, W. Xu, and Y. Ma, "RASL: Robust alignment by sparse and low-rank decomposition for linearly correlated images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2233–2246, 2012.

[26] Q. Zhao, D. Meng, Z. Xu, W. Zuo, and L. Zhang, "Robust principal component analysis with complex noise," in *International Conference on Machine Learning*, 2014, pp. 55–63.

[27] F. Nie, J. Yuan, and H. Huang, "Optimal mean robust principal component analysis," in *International Conference on Machine Learning*. JMLR Workshop and Conference Proceedings, 2014, pp. 1062–1070.

[28] B.-K. Bao, G. Liu, R. Hong, S. Yan, and C. Xu, "General subspace learning with corrupted training data via graph embedding," *IEEE Transactions on Image Processing*, vol. 22, no. 11, pp. 4380–4393, Nov 2013.

[29] G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "Subspace learning from image gradient orientations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 12, pp. 2454–2466, Dec 2012.

[30] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 11, pp. 2765–2781, 2013.

[31] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 171–184, 2013.

[32] X. Peng, Z. Yi, and H. Tang, "Robust subspace clustering via thresholding ridge regression," in *AAAI Conference on Artificial Intelligence (AAAI)*. AAAI, 2015.

[33] P. Favaro, R. Vidal, and A. Ravichandran, "A closed form solution to robust subspace estimation and clustering," in *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2011, pp. 1801–1807.

[34] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[35] X. He and P. Niyogi, "Locality preserving projections," in *Advances in Neural Information Processing Systems*. MIT Press, 2004, pp. 153–160.

[36] Z. Zhang, S. Yan, and M. Zhao, "Pairwise sparsity preserving embedding for unsupervised subspace learning and classification," *IEEE Transactions on Image Processing*, vol. 22, no. 12, pp. 4640–4651, 2013.

[37] J. Lu, G. Wang, W. Deng, and K. Jia, "Reconstruction-based metric learning for unconstrained face verification," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 1, pp. 79–89, Jan 2015.

[38] Y. Cong, J. Liu, J. Yuan, and J. Luo, "Self-supervised online metric learning with low rank constraint for scene categorization," *IEEE Transactions on Image Processing*, vol. 22, no. 8, pp. 3179–3191, 2013.

[39] B. Recht, M. Fazel, and P. Parrilo, "Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization," *SIAM Review*, vol. 52, no. 3, pp. 471–501, 2010.

[40] Z. Lin, R. Liu, and Z. Su, "Linearized alternating direction method with adaptive penalty for low-rank representation," in *Advances in Neural Information Processing Systems*, 2011, pp. 612–620.

[41] H. Zhang, Z. Yi, and X. Peng, “flrr: fast low-rank representation using frobenius-norm,” *Electronics Letters*, vol. 50, no. 13, pp. 936–938, 2014.

[42] X. Peng, C. Lu, Z. Yi, and H. Tang, “Connections between nuclear norm and frobenius norm based representation,” *arXiv:1502.07423*, 2015.

[43] A. Martinez and R. Benavente, “The AR face database,” 1998.

[44] M. Brand, “Fast low-rank modifications of the thin singular value decomposition,” *Linear algebra and its applications*, vol. 415, no. 1, pp. 20–30, 2006.

[45] P. O. Hoyer, “Non-negative matrix factorization with sparseness constraints,” *Journal of Machine Learning Research*, vol. 5, pp. 1457–1469, dec 2004.

[46] M. R. Osborne, B. Presnell, and B. A. Turlach, “A new approach to variable selection in least squares problems,” *IMA Journal of Numerical Analysis*, vol. 20, no. 3, pp. 389–403, 2000.

[47] A. Yang, A. Ganesh, S. Sastry, and Y. Ma, “Fast L1-Minimization algorithms and an application in robust face recognition: A review,” EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2010-13, February 5 2010.

[48] Z. Lin, A. Ganesh, J. Wright, L. Wu, M. Chen, and Y. Ma, “Fast convex optimization algorithms for exact recovery of a corrupted low-rank matrix,” *Computational Advances in Multi-Sensor Adaptive Processing*, vol. 61, 2009.

[49] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, “Robust face recognition via sparse representation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210–227, 2009.

[50] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, “Liblinear: A library for large linear classification,” *The Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.

[51] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman, “From few to many: Illumination cone models for face recognition under variable lighting and pose,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 643–660, 2001.

[52] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker, “Multi-PIE,” *Image and Vision Computing*, vol. 28, no. 5, pp. 807–813, 2010.

[53] S. Nayar, S. A. Nene, and H. Murase, “Columbia object image library (coil 100),” *Department of Comp. Science, Columbia University, Tech. Rep. CUCS-006-96*, 1996.