Sparse Representation Classifier Steered Discriminative Projection With Applications to Face Recognition

Jian Yang, *Member, IEEE*, Delin Chu, Lei Zhang, *Member, IEEE*, Yong Xu, *Member, IEEE*, and Jingyu Yang

***Abstract*— A sparse representation-based classifier (SRC) is developed and shows great potential for real-world face recognition. This paper presents a dimensionality reduction method that fits SRC well. SRC adopts a class reconstruction residual-based decision rule, we use it as a criterion to steer the design of a feature extraction method. The method is thus called the SRC steered discriminative projection (SRC-DP). SRC-DP maximizes the ratio of between-class reconstruction residual to within-class reconstruction residual in the projected space and thus enables SRC to achieve better performance. SRC-DP provides low-dimensional representation of human faces to make the SRC-based face recognition system more efficient. Experiments are done on the AR, the extended Yale B, and PIE face image databases, and results demonstrate the proposed method is more effective than other feature extraction methods based on the SRC.**

***Index Terms*— Dimensionality reduction, discriminant analysis, face recognition, feature extraction, sparse representation.**

I. Introduction

FACE recognition aroused broad interests in pattern recognition and computer vision areas in the past 20 years. Simultaneously, numerous face representation and classification methods are developed [1]. Recently, Wright *et al.* [2] presented a sparse representation-based classification (SRC) method. Xu *et al.* [27] suggested a two-phase test sample sparse representation method. He *et al.* [39] proposed a two-stage sparse representation for robust recognition on large-scale databases. Borrowing the idea of robust statistics, He *et al.* [40], [41] presented iteratively robust sparse representation methods for pattern recognition tasks. Yang *et al.* [31] gave a robust sparse coding method for face recognition. All of these sparsity-based methods are applied to real-world face recognition problems and demonstrate to be very effective and robust to varying expression and illumination as well as occlusion and disguise. The basic idea of SRC is to represent a given test sample as a sparse linear combination of all training samples; the sparse nonzero representation coefficients are supposed to concentrate on the training samples with the same class label as the test sample. Yang *et al.* [32] gave an insight into SRC and provided some theoretical supports for its effectiveness. Zhang *et al.* [45] argued that the collaborative representation strategy plays a more important role than the L_1-norm-based sparsity constraint and presented a collaborative representation classifier (CRC), which is computationally more efficient. CRC, however, does not provide a mechanism for noise removing, so it is not a robust method for face recognition.

Sparse representation involves an underdetermined system of linear equation $\mathbf{y} = \mathbf{A}\mathbf{w}$, where $\mathbf{A} \in R^{N\times M}$ and $N < M$. For SRC, the columns of matrix A are all training sample vectors. To obtain a sparse solution, the dimension of feature vectors should be smaller than the number of training samples. To deal with small sample problems like face recognition, where the dimension of images is larger than the training sample size, a dimensionality reduction (feature extraction) step becomes necessary before implementing SRC. Wright [2] showed that the choice of features is not critical, as long as the sparse representation is correctly computed and the number of features is sufficiently large. But, when the number of features is relatively small, there exist remarkable performance differences between different feature extraction methods. A small amount of representation features is preferable for the real-world face recognition problems, because it can reduce the storage requirements and improve the classification efficiency. So, the goal of this paper is to explore a method that can use a small amount of representation features to achieve better performance using SRC.

By far, numerous dimensionality reduction methods are developed for face representation. In addition to the most well-known methods like principal component analysis (PCA) and Fisher linear discriminant analysis (FLDA) [3], [4], a family of kernel-based and manifold learning-related methods aroused wide research interests. Yang *et al.* [33] proposed a complete kernel Fisher discriminant framework for feature extraction. Cevikalp *et al.* [34] presented a discriminative

Manuscript received August 30, 2012; revised December 26, 2012; accepted February 15, 2013. Date of publication March 13, 2013; date of current version May 14, 2013. This work was supported by the National Science Fund for Distinguished Young Scholars under Grant 61125305 and Grant 61233011 and the Key Project of Chinese Ministry of Education.

J. Yang and J. Yang are now with School of Computer Science and Technology, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: csjyang@njust.edu.cn; yangjy@njust.edu.cn).

D. Chu is with the Department of Mathematics, National University of Singapore, 119076 Singapore (e-mail: matchudl@nus.edu.sg).

L. Zhang is with the Department of Computing, Hong Kong Polytechnic University, Kowloon, Hong Kong (e-mail: cslzhang@comp.polyu.edu.hk).

Y. Xu is with the Bio-Computing Research Center, Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen 518055, China (e-mail: laterfall2@yahoo.com.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2013.2249088

common vector method with kernels. Zafeiriou *et al.* [35] suggested a regularized kernel discriminant analysis method for face recognition and verification. Kim and Kittler [5] presented the locally linear discriminant analysis for multi-modally distributed classes. He *et al.* [6] proposed a method called locality preserving projections (LPP) and applied to face representation. Chen *et al.* [7] suggested the local discriminant embedding and its two variants. Yan *et al.* [8] provided a graph embedding-based dimensionality reduction framework with marginal Fisher analysis. Cai *et al.* [10], [11] presented the orthogonal Laplacianfaces and a spatially smooth subspace for face recognition. Fan *et al.* [36] proposed a sample neighbors-based local linear discriminant analysis framework.

Recently, the idea of sparse representation is used to design some feature extraction methods. Qiao *et al.* [9] presented the sparsity preserving projections (SPP) method, which uses all training samples to sparsely represent a given sample and seeks a linear projection such that the sparse representation coefficients are preserved. Zhang *et al.* [44] recently presented a graph optimization for dimensionality reduction with sparsity constraints (GODRSC). GODRSC aims to simultaneously seek the sparse representation coefficients and the projection matrix. GODRSC essentially learns a sparse relationship graph in the transformed space, thus it can be viewed as an extension of SPP. Clemmensen *et al.* [28] provided a sparse linear discriminant analysis (SLDA), which imposes a sparseness constraint on projection vectors. The sparse projection vector yields a set of interpretable features for classification. Lai *et al.* [29] suggested a sparse version of the 2-D local discriminant projections (S2DLDP), which provides an intuitive, semantic, and interpretable feature subspace for face representation. They also showed that the optimal sparse subspace approximates to the eigen subspace that is obtained by solving a generalized eigenfunction [30]. Wang *et al.* [37] presented a sparse tensor discriminant analysis method for color space learning and face verification. He *et al.* [42], [43] presented the nonparametric maximum entropy criterion-based PCA and discriminant analysis for robust feature extraction.

All feature extraction methods, however, have no direct connection to SRC. In this paper, our goal is to develop a feature extraction method fitting SRC well. Observing that SRC adopts a class reconstruction residual-based decision rule, we use it as a criterion to steer the design of a feature extraction method, which is thus coined the SRC steered discriminative projection (SRC-DP). The basic idea of SRC-DP is to seek a linear transformation such that in the transformed low-dimensional space, the within-class reconstruction residual is as small as possible and simultaneously the between-class reconstruction residual is as large as possible. Therefore, SRC can achieve better classification performance in the SRC-DP transformed space.

Compared with existing dimensionality reduction methods, SRC-DP has the following advantages. Initially, the proposed method has a natural connection to pattern classification. The SRC-DP-based feature extractor and the SRC can be seamlessly integrated into a face recognition system. Secondly, similar to GODRSC [44], the proposed method characterizes scatters of samples in the low-dimensional transformed space where the classifier practically works, thus it can achieve desirable classification performance. The classical FLDA also characterizes scatters of samples in the transformed space. But, most existing locality (or sparsity)-characterization-based discriminant analysis methods [6]–[11] are designed-based on scatters of samples in the input space. This is because the locality (or sparsity) relationship of samples, unlike the population mean or class mean of FLDA, might be changed after a linear transformation. This scatter characterization in the input space cannot guarantee good performance of classifiers in the transformed space.

This paper is an extended version of our international conference on pattern recognition (ICPR) paper [38]. In contrast, in this paper, we provide an initialization method, discuss the convergence of the iterative SRC-DP algorithm, and show the connections to other sparse feature extraction methods. We also present a flexible version of SRC-DP, and further suggest an extended SRC-DP method that provides a mechanism to deal with occlusion or corruption. In addition, more experiments are done to evaluate the effectiveness of the proposed method $\mathbb{R}^{d\times M_i}$.

II. Sparse Representation-Based Classifier

If there are c known pattern classes. Let $\mathbf{A}_i$ be the matrix formed by the training samples of class i, i.e., $\mathbf{A}_i = [\mathbf{y}_{i1}, \mathbf{y}_{i2}, \ldots, \mathbf{y}_{iM_i}] \in \mathbb{R}^{d\times M_i}$ where M_i is the number of training samples of class i. Let us define a matrix $\mathbf{A} = [\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_c] \in \mathbb{R}^{d\times M}$, where $M = \sum_{i=1}^{c} M_i$. The matrix $\mathbf{A}$ is obviously composed of entire training samples.

Given a test sample y, we represent y in an overcomplete dictionary whose basis vectors are training sample themselves, i.e., $\mathbf{y} = \mathbf{A}\mathbf{w}$. If the system of linear equation is underdetermined ($N < M$), this representation is naturally sparse. The sparsest solution can be sought by solving the following optimization problem

$$(L_0) \quad \hat{\mathbf{w}}_0 = \arg\min ||\mathbf{w}||_0, \text{ subject to } \mathbf{A}\mathbf{w} = \mathbf{y} \tag{1}$$

where $||\cdot||_0$ is the L_0-norm, which counts the number of nonzero entries in a vector.

Solving L_0 optimization problem in (1), however, is NP hard and extremely time-consuming. Fortunately, recent research efforts reveal that for certain dictionaries, if the solution $\hat{\mathbf{w}}_0$ is spare enough, finding the solution of the L_0 optimization problem is equivalent to finding the solution to the following L_1 optimization problem [12], [13]

$$(L_1) \quad \hat{\mathbf{w}}_1 = \arg\min ||\,\mathbf{w}||_1, \text{ subject to } \mathbf{A}\mathbf{w} = \mathbf{y}. \tag{2}$$

This problem can be solved in polynomial time by standard linear programming algorithms [14]. A more efficient algorithm, e.g., the homotopy algorithm that has a computational complexity that is linear to the size of the training set, is available recently [15].

After the sparsest solution $\hat{\mathbf{w}}_1$ is obtained, the SRC can be done in the following way [2]. For each class i, let $\delta_i : \mathbb{R}^M \rightarrow \mathbb{R}^M$ be the characteristic function that selects the coefficients associated with the ith class. For $\mathbf{w} \in \mathbb{R}^M$, $\delta_i(\mathbf{w})$ is a vector whose only nonzero entries are the entries in

$\mathbf{w}$ that are associated with class i. Using only the coefficients associated with the ith class, one can reconstruct a given test sample y as $\mathbf{v}^i = \mathbf{A}\delta_i(\hat{\mathbf{w}}_1)$. $\mathbf{v}^i$ is called the prototype of class i with respect to the sample y. The distance (residual) between y and its prototype $\mathbf{v}^i$ of class i are defined by

$$r_i(\mathbf{y}) = ||\mathbf{y} - \mathbf{v}^i||_2 = ||\mathbf{y} - \mathbf{A}\delta_i(\hat{\mathbf{w}}_1)||_2. \tag{3}$$

The SRC decision rule is: if $r_l(\mathbf{y}) = \min_i r_i(\mathbf{y})$, $\mathbf{y}$ is assigned to class l.

III. Sparse Representation Classifier Steered Discriminative Projection

Here, we first present the SRC-DP method, put forward an iterative SRC-DP algorithm and suggest a good initial solution for it. Then, we discuss the convergence of the iterative SRC-DP algorithm, provide a flexible version of SRC-DP, and reveal the connections between SRC-DP and SPP. Finally, we suggest a way to deal with the rank-deficiency problem of SRC-DP as applied to face recognition.

A. SRC-DP: Basic Idea and Algorithm

Let $\mathbf{B} = [\mathbf{B}_1, \mathbf{B}_2, \ldots, \mathbf{B}_c] \in R^{N\times M}$ be the training data matrix in the input space, where $\mathbf{B}_i = [\mathbf{x}_{i1}, \mathbf{x}_{i2}, \ldots, \mathbf{x}_{iM_i}] \in \mathbb{R}^{N\times M_i}$ is the matrix formed by the training samples of class i. Under a linear transformation $\mathbf{y} = \mathbf{P}^T\mathbf{x}$, each data point $\mathbf{x}_{ij}$ in input space $\mathbb{R}^N$ is mapped into $\mathbf{y}_{ij} = \mathbf{P}^T\mathbf{x}_{ij}$ in a d-dimensional space $\mathbb{R}^d$. As a result, the data matrix in the input space is converted into the one in $\mathbb{R}^d$, that is, $\mathbf{A} = \mathbf{P}^T\mathbf{B}$.

Now consider the classification problem using SRC in the mapped d-dimensional space. For each training sample $\mathbf{y}_{ij}$, leave it out from the training set and use the remaining training samples to linearly represent it. By solving the L_1 optimization problem in (2), we obtain a representation coefficient vector $\mathbf{w}_{ij}$. Let $\delta_s(\mathbf{w}_{ij})$ be the representation coefficient vector with respect to class s. Then, the prototype of class s with respect to the sample $\mathbf{y}_{ij}$ is $\mathbf{v}_{ij}^s = \mathbf{A}\delta_s(\mathbf{w}_{ij})$, $s = 1, \ldots, c$. The distance between $\mathbf{y}_{ij}$ and class s is defined as

$$d_s(\mathbf{y}_{ij}) = \left\|\mathbf{y}_{ij} - \mathbf{v}_{ij}^s\right\|^2. \tag{4}$$

Considering the decision rule of SRC, to make the classifier perform well, $\mathbf{y}_{ij}$ is as close as possible to its within-class prototype $\mathbf{v}_{ij}^i$ and simultaneously as far away as possible from its between-class prototypes $\mathbf{v}_{ij}^s$ $(s \neq i)$. That is, the within-class distance $d_i(\mathbf{y}_{ij})$ is supposed to be as small as possible and the between-class distance $d_s(\mathbf{x}_{ij})$ for each $s \neq i$ as large as possible. To make SRC achieve good performance on all training samples, we would like to characterize the average within-class and between-class distances, i.e., the within-class and between-class scatters. Specifically, the within-class scatter is defined as follows:

$$\begin{aligned}\frac{1}{M}\sum_{i,j} d_i(\mathbf{y}_{ij}) &= \frac{1}{M}\sum_{i,j}\left\|\mathbf{y}_{ij} - \mathbf{v}_{ij}^i\right\|^2 \\ &= \frac{1}{M}\sum_{i,j}(\mathbf{y}_{ij} - \mathbf{v}_{ij}^i)^T(\mathbf{y}_{ij} - \mathbf{v}_{ij}^i) \\ &= \mathrm{tr}(\tilde{\mathbf{S}}_w^L)\end{aligned} \tag{5}$$

where $\tilde{\mathbf{S}}_w^L = 1/M\sum_{i,j}(\mathbf{y}_{ij} - \mathbf{v}_{ij}^i)^T(\mathbf{y}_{ij} - \mathbf{v}_{ij}^i)$ is the within-class scatter matrix in the transformed space and $\mathrm{tr}(\cdot)$ is the trace operator.

The between-class local scatter of samples is defined as follows:

$$\frac{1}{M(c-1)}\sum_{i,j}\sum_{s\neq i} d_s(\mathbf{y}_{ij}) = \frac{1}{M(c-1)}\sum_{i,j}\sum_{s\neq i}\left\|\mathbf{y}_{ij} - \mathbf{v}_{ij}^s\right\|^2 = \mathrm{tr}(\tilde{\mathbf{S}}_b^L) \tag{6}$$

where $\tilde{\mathbf{S}}_b^L = 1/M(c-1)\sum_{i,j}\sum_{s\neq i}(\mathbf{y}_{ij} - \mathbf{v}_{ij}^s)(\mathbf{y}_{ij} - \mathbf{v}_{ij}^s)^T$ is the between-class scatter matrix in the transformed space.

According to the SRC decision rule, larger between-class scatter and smaller within-class scatter will lead to better classification results in an average sense. Therefore, we can choose to maximize the following criterion function:

$$J(\mathbf{P}) = \frac{\mathrm{tr}(\tilde{\mathbf{S}}_b^L)}{\mathrm{tr}(\tilde{\mathbf{S}}_w^L)}. \tag{7}$$

Now, consider how to determine a projection matrix P-based on the criterion. Finally, we try to convert the criterion to be a function with respect to P.

Inserting $\mathbf{y}_{ij} = \mathbf{P}^T\mathbf{x}_{ij}$, $\mathbf{A} = \mathbf{P}^T\mathbf{B}$, and $\mathbf{v}_{ij}^s = \mathbf{A}\delta_s(\mathbf{w}_{ij})$ into the formula of $\tilde{\mathbf{S}}_w^L$ and $\tilde{\mathbf{S}}_b^L$, respectively, we have

$$\begin{aligned}\tilde{\mathbf{S}}_w^L &= \frac{1}{M}\sum_{i,j}[\mathbf{P}^T\mathbf{x}_{ij} - \mathbf{P}^T\mathbf{B}\delta_i(\mathbf{w}_{ij})][\mathbf{P}^T\mathbf{x}_{ij} - \mathbf{P}^T\mathbf{B}\delta_i(\mathbf{w}_{ij})]^T \\ &= \mathbf{P}^T\mathbf{S}_w^L\mathbf{P}\end{aligned} \tag{8}$$

$$\begin{aligned}\tilde{\mathbf{S}}_b^L &= \frac{1}{M(c-1)}\sum_{i,j}\sum_{s\neq i}[\mathbf{P}^T\mathbf{x}_{ij} - \mathbf{P}^T\mathbf{B}\delta_s(\mathbf{w}_{ij})] \\ &\quad\times[\mathbf{P}^T\mathbf{x}_{ij} - \mathbf{P}^T\mathbf{B}\delta_s(\mathbf{w}_{ij})]^T \\ &= \mathbf{P}^T\mathbf{S}_b^L\mathbf{P}\end{aligned} \tag{9}$$

where the two matrices $\mathbf{S}_w^L$ and $\mathbf{S}_b^L$ are defined as

$$\mathbf{S}_w^L = \frac{1}{M}\sum_{i,j}[\mathbf{x}_{ij} - \mathbf{B}\delta_i(\mathbf{w}_{ij})][(\mathbf{x}_{ij} - \mathbf{B}\delta_i(\mathbf{w}_{ij})]^T \tag{10}$$

and

$$\mathbf{S}_b^L = \frac{1}{M(c-1)}\sum_{i,j}\sum_{s\neq i}[\mathbf{x}_{ij} - \mathbf{B}\delta_s(\mathbf{w}_{ij})][\mathbf{x}_{ij} - \mathbf{B}\delta_s(\mathbf{w}_{ij})]^T. \tag{11}$$

$\mathbf{S}_w^L$ and $\mathbf{S}_b^L$ are called the within-class and between-class sparse scatter matrices, respectively.

The criterion in (7) thereby becomes

$$J(\mathbf{P}) = \frac{\mathrm{tr}(\mathbf{P}^T\mathbf{S}_b^L\mathbf{P})}{\mathrm{tr}(\mathbf{P}^T\mathbf{S}_w^L\mathbf{P})}. \tag{12}$$

If the two matrices $\mathbf{S}_w^L$ and $\mathbf{S}_b^L$ can be constructed directly in the input space and $\mathbf{S}_w^L$ is nonsingular, the optimal projection matrix P can be determined by maximizing the criterion in (12). Generally, we add the constraint $\mathbf{P}^T\mathbf{S}_w^L\mathbf{P} = \mathbf{I}$ such that the extracted features are uncorrelated [21]. Then, the column vectors of the optimal projection matrix can be chosen as the generalized eigenvectors of $\mathbf{S}_b^L\varphi = \lambda\mathbf{S}_w^L\varphi$ corresponding to d largest eigenvalues. However, the representation coefficient vector $\mathbf{w}_{ij}$ in the formula of $\mathbf{S}_w^L$ and $\mathbf{S}_b^L$ is unknown to us without the projection matrix P being given in advance, $\mathbf{w}_{ij}$ is calculated in the transformed space $\mathbb{R}^d$.

In summary, given an initial projection matrix $\mathbf{P}_0$ in advance, we can map the data $\mathbf{x}_{ij}$ in the input space into

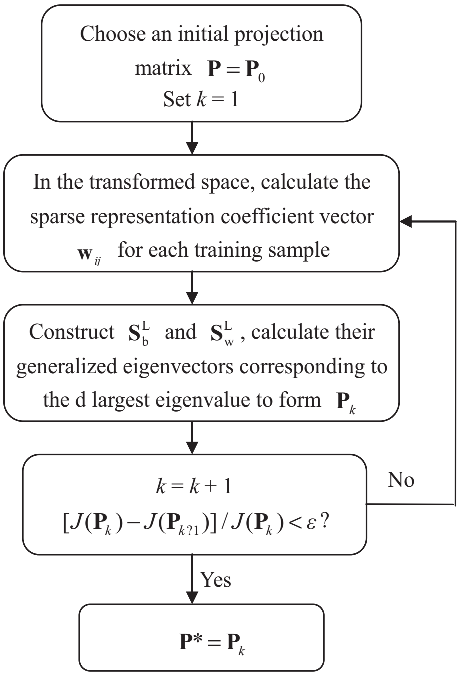

Fig. 1. Overview of iterative SRC-DP algorithm.

$\mathbf{y}_{ij} = \mathbf{P}_0^T \mathbf{x}_{ij}$ in the transformed space. Representing each point $\mathbf{y}_{ij}$ using the remaining training samples and obtain the representation coefficient vector $\mathbf{w}_{ij}$ by solving the L_1 optimization problem. We then can construct the matrices $\mathbf{S}_w^L$ and $\mathbf{S}_b^L$ in the input space and obtain a new projection matrix $\mathbf{P}_1$ by solving the corresponding generalized eigenvalue problem. Based on this idea, we can derive an iterative SRC-DP algorithm. In the kth iteration of the algorithm, we use $\mathbf{P}_{k-1}$, the resulting projection matrix after the (k-1)th iteration, as the initial projection matrix to input the system to yield a new projection matrix $\mathbf{P}_k$. The iteration procedure continues until the algorithm converges. Convergence may be determined by observing when the value of the criterion function $J(\mathbf{P})$ in (12) stops changing. Specifically, after k times of iterations, if $J(\mathbf{P}_k) - J(\mathbf{P}_{k-1}) < \varepsilon$, we think the algorithm converges and choose $\mathbf{P}* = \mathbf{P}_k$. To eliminate the effect of the magnitude of $J(\mathbf{P})$ onto the choice of the parameter ε, we use the relative difference-based convergence criterion instead here, that is, $[J(\mathbf{P}_k) - J(\mathbf{P}_{k-1})]\big/J(\mathbf{P}_k) < \varepsilon$.

The SRC-DP algorithm (Algorithm 1) is shown in Fig. 1.

B. Initial Solution of SRC-DP

The initial solution (seed) $\mathbf{P}_0$ can be chosen as an $N \times d$ random matrix. The $\mathbf{P}_0$-determined transform $\mathbf{y} = \mathbf{P}_0^T \mathbf{x}$ is generally referred to as a random projection (mapping) [16]. Here we would rather give a better initial solution $\mathbf{P}_0$ for the SRC-DP algorithm, when all of the training samples form an overcomplete dictionary in the input space.

If the sparsity is preserved under linear transform[1], that is, for each point $\mathbf{y}_{ij} = \mathbf{P}^T \mathbf{x}_{ij}$ in the transformed space, the corresponding sparse representation coefficient vector $\mathbf{w}_{ij}$ is exactly the same as that obtained in the input space. In other words, the sparse representation coefficient vector with respect to $\mathbf{x}_{ij}$ in the input space is supposed to be $\mathbf{w}_{ij}$. Under this assumption, we can calculate $\mathbf{w}_{ij}$ directly in the input space. As a result, the two matrices $\mathbf{S}_w^L$ and $\mathbf{S}_b^L$ can be constructed directly in the input space. We can thus get a close solution by maximizing the criterion in (12). Specifically, the algorithm for getting $\mathbf{P}_0$ is given below.

Initial SRC-DP algorithm (Algorithm 2).

1) *Step 1.* For each train sample $\mathbf{x}_{ij}$ in the input space, represent it using the remaining training samples and calculate its corresponding sparse representation coefficient vector $\mathbf{w}_{ij}$ by solving the L_1 optimization problem.
2) *Step 2.* Construct the within-class and between-class sparse scatter matrices $\mathbf{S}_w^L$ and $\mathbf{S}_b^L$ using (10) and (11). Calculate the generalized eigenvectors $\varphi_1, \ldots, \varphi_d$ of $\mathbf{S}_b^L$, and $\mathbf{S}_w^L$ corresponding to the d largest generalized eigenvalues. Let $\mathbf{P}_0 = (\varphi_1, \ldots, \varphi_d)$.

Finally, note that the assumption that the sparsity is preserved under linear transform might not hold in practice. Without this assumption, we have to appeal to the iterative SRC-DP algorithm (Algorithm 1) to find the optimal projection matrix. This assumption is just to make the problem of finding the optimal projection matrix simpler. Obviously, the obtained optimal projection matrix $\mathbf{P}_0$ under this assumption is not necessarily optimal in practice. But, it can be used as a good initial solution of the iterative SRC-DP algorithm.

In addition, the initial SRC-DP algorithm (Algorithm 2) can be looked at as a noniterative version of SRC-DP. The resulting $\mathbf{P}_0$ can be used as an approximation of the optimal projection matrix for feature extraction directly.

C. Convergence of the SRC-DP Algorithm

For convenience in discussing the convergence of our algorithm, we first introduce an equivalent criterion of (12). To this end, let us define the total sparse scatter matrix

$$\mathbf{S}_t^L = \frac{c}{M(c-1)} \sum_{i,j} \sum_{s} [\mathbf{x}_{ij} - \mathbf{B}\delta_s(\mathbf{w}_{ij})][\mathbf{x}_{ij} - \mathbf{B}\delta_s(\mathbf{w}_{ij})]^T \quad (13)$$

where we know that $\mathbf{S}_t^L = \mathbf{S}_b^L + \mathbf{S}_w^L$. Then, when $\mathbf{S}_w^L$ is nonsingular, the criterion in (12) is equivalent to

$$J_t(\mathbf{P}) = \frac{\mathrm{tr}(\mathbf{P}^T \mathbf{S}_b^L \mathbf{P})}{\mathrm{tr}(\mathbf{P}^T \mathbf{S}_t^L \mathbf{P})}. \quad (14)$$

It is easy to show that the column vectors of the projection matrix that maximizes $J_t(\mathbf{P})$ under the constraint $\mathbf{P}^T \mathbf{S}_w^L \mathbf{P} = \mathbf{I}$ are also the generalized eigenvectors of $\mathbf{S}_b^L \varphi = \lambda\, \mathbf{S}_w^L \varphi$. Therefore, in the SRC-DP algorithm (Algorithm 1), the criterion $J(\mathbf{P})$ can be replaced by $J_t(\mathbf{P})$. In the following, we analyze the convergence of Algorithm 1-based on the criterion $J_t(\mathbf{P})$ and draw the following conclusion:

Proposition 1: When $\mathbf{S}_w^L$ is nonsingular, the criterion function of the SRC-DP algorithm converges to a local maximum.

Proof: The criterion function $J_t(\mathbf{P})$ monotonically increases. If $\mathbf{P}_{k-1}$ and $\mathbf{P}_k$ are, respectively, the optimal projection matrices at the $(k-1)$th and the kth iteration step. We have $J(\mathbf{P}_{k-1}) \leq J(\mathbf{P}_k)$. Otherwise, the SRC-DP algorithm stops and $J(\mathbf{P}_{k-1})$ is the local maximum. As the criterion function $J_t(\mathbf{P})$

[1]Actually, a similar assumption has been used in [9] and [6].

has an upper bound when $\mathbf{S}_w^L$ is nonsingular, i.e., $J_t(\mathbf{P}) < 1$, the criterion function of the SRC-DP algorithm must converge to a local maximum.

As we can only show that the criterion function of the SRC-DP algorithm converges to a local maximum, the solution of our algorithm is theoretically locally-optimal. Therefore, choosing a good initial solution for it is important. Algorithm 2 can provide such a solution.

Here, we should make a remark: that the criterion function of the SRC-DP algorithm converges does not mean that the solution of the algorithm converges to a unique matrix. In other words, beginning with different initial solutions, the resulting optimal projection matrices might be different, even if their objective function converges to the same value. One possible reason is: the value of criterion function is invariant under arbitrary orthogonal transformations of a solution. If P* is the optimal solution. Then, QP* is also the optimal solution, where Q is a orthogonal matrix, because $J_t(\mathbf{P}*) = J_t(\mathbf{QP}*)$ [or $J(\mathbf{P}*) = J(\mathbf{QP}*)$]. Fortunately, the orthogonal transformation of a projection matrix does not affect the classification result of the SRC classifier. The justifications are two-fold: 1) an additional orthogonal transformation of samples does not change the solution of the L_1 optimization problem in (2) (i.e., sparse representation coefficients of samples); and 2) an orthogonal transformation does not change the sample-to-class distance defined in (3) in the classification rule. So, different projection matrices with the same criterion function value will result in the same classification performance.

D. Flexible Version of SRC-DP

In real-world pattern recognition problems, the data are generally noisy or probably there are not enough training samples for representing a testing sample exactly (i.e., the dictionary formed by all training samples is not overcomplete). To deal with these cases, instead of seeking the sparsest exact representation of a test sample, we can seek the sparsest representation that satisfies a given approximation error. That is, the exact sparse representation model in (2) is replaced by

$$\hat{\mathbf{w}}_1 = \arg\min ||\mathbf{w}||_1, \text{ subject to } ||\mathbf{Aw} - \mathbf{y}|| \le \varepsilon \tag{15}$$

where ε is a given approximation error. To further provide more flexibility to the variation of approximation error, we would rather use the following Lasso model:

$$\hat{\mathbf{w}}_1 = \arg\min ||\mathbf{Aw} - \mathbf{y}||_2^2 + \delta||\mathbf{w}||_1 \tag{16}$$

where $\delta > 0$ is the regularization parameter. This model balances the sparsity and approximation error adaptively by modifying the regularization parameter.

If we use (16) instead of (2) to calculate the sparse representation coefficients, the derived sparse representation classifier is called the flexible SRC (FSRC), and the derived SRC-DP is named the FSRC-DP.

E. Connection to Other Sparse Feature Extraction Methods

Motivated by the idea of LPP [6] and sparse representation [2], Qiao *et al.* [9] recently presented the SPP method. LPP uses the nearest neighbors of a point in the input space to characterize the locality and seeks a linear projection such that the neighborhood relationship is preserved after the projection. In contrast, SPP uses all training samples to sparsely represent a point in the input space and look for a linear projection such that the sparse reconstruction relationship is preserved. Specifically, for each training sample $\mathbf{y}_{ij}$, let us represent it sparsely by using the remaining training samples in the input space. The sparse representation weights $\mathbf{w}_{ij}$ can be obtained by solving the L_1 optimization problem in (2) [or (15) or (16)]. Assuming the sparse representation weights $\mathbf{w}_{ij}$ is preserved, one seeks a linear projection such that the total reconstruction residual (error) criterion

$$\sum_{i,j} \left\| \mathbf{P}^T(\mathbf{x}_{ij} - \mathbf{Aw}_{ij}) \right\|^2 = \mathrm{tr}\left\{ \mathbf{P}^T \left[\sum_{ij} (\mathbf{x}_{ij} - \mathbf{Aw}_{ij})(\mathbf{x}_{ij} - \mathbf{Aw}_{ij})^T \right] \mathbf{P} \right\} \tag{17}$$

is minimized under the constraint that $\mathbf{P}^T(\mathbf{AA}^T)\mathbf{P} = \mathbf{I}$.

The difference between SRC-DP and SPP can be specified in the following.

SRC-DP is a supervised method whereas SPP is unsupervised. SPP tries to minimize the total reconstruction residual, which is not very meaningful for classification. So, SPP has no direct connections to SRC or any other classifiers. In contrast, SRC-DP aims to minimize the within-class reconstruction residual in (5) and simultaneously to maximize the between-class reconstruction residual in (6), which is consistent with the classification rule of the SRC classifier. So, SRC-DP fits for SRC very well.

SRC-DP models the separability by computing scatters of samples in the transformed space, whereas SPP does not. The modeling in the transformed space makes more sense because the classifier practically works in such a space. On the other hand, this modeling does not rely on the sparsity-preservation assumption anymore. This is because the SRC-DP calculates the sparse representation coefficients in the transformed space, and uses these coefficients to form two scatter matrices. In contrast, SPP performs sparse representation in the input space and assume the sparse representation coefficients can be preserved in the transformed space. This makes SRC-DP more suitable for real-world problems. However, there is no free dinner; modeling in transformed space brings an additional computational burden: the iterative SRC-DP algorithm (Algorithm 1) is more time-consuming than the noniterative SPP for training.

It is interesting for us to compare the two noniterative algorithms: SPP and the initial SRC-DP (Algorithm 2). Both methods have a same computational complexity, and share a common sparsity-preservation assumption, i.e., using the sparse representation weights computed in the input space in the modeling procedure. The difference is: SPP is an unsupervised method whereas the initial SRC-DP (Algorithm 2) is supervised. The connection between the two methods is similar to the connection between PCA and FLDA.

SRC-DP and SPP both utilize the between-sample sparseness, i.e., representing one sample using the other samples sparsely. The projection vector itself is not necessary sparse.

In contrast, the SLDA [28] and S2DLDP [29] emphasize the within-sample sparseness, yielding sparse projection vectors. The sparse projection vectors are essentially to conduct feature selection, and thus result in a set of interpretable features.

Zhang *et al.* [44] recently extended SPP and presented a GODRSC. GODRSC and SRC-DP share a similar iterative process of optimization. Their difference is: GODRSC is unsupervised method, whereas our SRC-DP is supervised method whose criterion is closely connected to the decision rule of sparse representation classifier.

F. Dealing With the Rank-Deficiency Problem

Recalling that in solving the criterion function in (12) under the constraint $\mathbf{P}^T\mathbf{S}_\omega^L\mathbf{P} = \mathbf{I}$, we require that the within-class sparse scatter matrix $\mathbf{S}_\omega^L$ is nonsingular (full-rank). Otherwise, if $\mathbf{S}_\omega^L$ is singular (rank-deficient), we cannot obtain its solution by solving the generalized eigen-equation $\mathbf{S}_b^L\varphi = \lambda\mathbf{S}_\omega^L\varphi$. So, we need to study the rank of $\mathbf{S}_\omega^L$. To this end, let us formulate $\mathbf{S}_\omega^L$ alternatively by the following derivation. In the derivation, the multiplier $1/M$ in the formula of $\mathbf{S}_\omega^L$ is neglected for convenience

$$\begin{aligned}\mathbf{S}_\omega^L &= \sum_{i,j}[\mathbf{x}_{ij} - \mathbf{B}\delta_i(\mathbf{w}_{ij})][(\mathbf{x}_{ij} - \mathbf{B}\delta_i(\mathbf{w}_{ij})]^T\\ &= \sum_{i,j}\mathbf{x}_{ij}\mathbf{x}_{ij}^T - 2\mathbf{B}\sum_{i,j}\delta_i(\mathbf{w}_{ij})\mathbf{x}_{ij}^T\\ &\quad +\mathbf{B}\left[\sum_{i,j}\delta_i(\mathbf{w}_{ij})(\delta_i(\mathbf{w}_{ij}))^T\right]\mathbf{B}^T\\ &= \mathbf{B}\mathbf{B}^T - 2\mathbf{B}\Delta\mathbf{B}^T + \mathbf{B}\Delta\Delta^T\mathbf{B}^T\\ &= \mathbf{B}(\mathbf{I}-\Delta)(\mathbf{I}-\Delta)^T\mathbf{B}^T \end{aligned} \tag{18}$$

where $\mathbf{B} \in R^{N\times M}$ is the matrix formed by all training samples, I is the M by M identity matrix, and the matrix Δ is defined as

$$\Delta = \Big[\delta_1(\mathbf{w}_{11}), \ldots, \delta_1(\mathbf{w}_{1M_1}), \delta_2(\mathbf{w}_{21}), \ldots, \delta_2(\mathbf{w}_{2M_1}), \ldots, \delta_c(\mathbf{w}_{c1}), \ldots, \delta_2(\mathbf{w}_{cM_c})\Big]. \tag{19}$$

From (18), we know that $\mathrm{rank}(\mathbf{S}_\omega^L) \le \min\{M, N\}$, where N is the dimension of input space and M is the number of training samples. Actually, following a similar derivation procedure, we can reformulate $\mathbf{S}_b^L$ and show that $\mathrm{rank}(\mathbf{S}_b^L) \le \min\{M, N\}$.

In face recognition problems, the training sample size is generally smaller than the dimension of image pixel space, we will encounter two problems if we use the image pixel space as the input space and directly implement SRC-DP in such a space: 1) the within-class sparse scatter matrix $\mathbf{S}_\omega^L$ is rank-deficient when $M<N$, thus it is difficult to solve the criterion in (12) directly in the iterative SRC-DP algorithm and 2) the dictionary formed by all training samples is undercomplete, so it becomes impossible to find the solution of the L_1 optimization problem in (2). Thereby, Algorithm 2 cannot be used to obtain an initial solution.

To address these two problems, we can use PCA for dimensionality reduction prior to implementing SRC-DP. PCA is probably the most popular method for data preprocessing and always used to alleviate the rank-deficiency problems. The FLDA [3], [4], LPP [6], and SPP [9] all use PCA as a preprocessing step when applied to face recognition. For consistency, we use PCA to reduce the dimension of face images and then perform SRC-DP in the PCA-transformed space.

G. Dealing With Occlusion or Corruption

The SRC method [2] has a built-in mechanism to deal with occlusion or corruption, via solving an extended L_1 optimization problem. Specifically, in occlusion or corruption, the linear combination of the image y can be modified as

$$\mathbf{y} = \mathbf{A}\mathbf{w} + \mathbf{e} = [\mathbf{A}, \mathbf{I}]\begin{bmatrix}\mathbf{w}\\ \mathbf{e}\end{bmatrix} = \bar{\mathbf{A}}\bar{\omega}. \tag{20}$$

If the error vector $\mathbf{e}$ is also sparse, we then can recover the representation coefficients and the error vector together by solving the following, extended L_1 optimization problem

$$\hat{\mathbf{w}}_1 = \arg\min ||\bar{\mathbf{w}}||_1, \ \text{subject to } \bar{\mathbf{A}}\bar{\omega} = \mathbf{y}. \tag{21}$$

Now, let us consider how to extend our SRC-DP to deal with occlusion or corruption in images. There might exist some occluded or corrupted images in the training set. So, in the training phase, for each training image, we leave it out from the training set and use the remaining training samples to linearly represent it using (20). By solving the problem in (21), we get the optimal representation coefficient vector $\mathbf{w}$. Based on this $\mathbf{w}$, we can construct $\mathbf{S}_\omega^L$ and $\mathbf{S}_b^L$ and perform SRC-DP. This SRC-DP is called the extended SRC-DP, as it uses the weights derived from the extended L_1 optimization problem in (21). In addition, based on this optimal weight vector $\mathbf{w}$, we obtain clean training images via $\mathbf{y}_{\text{clean}} = \mathbf{A}\mathbf{w}$.

In the testing phase, for each testing image (which might be occluded or corrupted), we use the obtained clean training images to represent it using (20) and then get the clean testing image by solving the problem in (21). We then perform feature extraction upon the clean testing image using the projection matrix of the extended SRC-DP.

The extended SRC-DP must be performed upon image vectors as it involves a noise-removing process that must be operated in image space. That is, we cannot use PCA as a preliminary step before implementing the extended SRC-DP. To address the rank-deficiency problem mentioned above, we can choose to reduce the size of images directly by downsampling.

IV. Face Recognition Experiments and Analysis

A. Experiment Using the AR Database

The AR face [17] contains over 4000 color face images of 126 people, including frontal views of faces with different facial expressions, lighting conditions, and occlusions. The pictures of 120 individuals (65 men and 55 women) are taken in two sessions (separated by two weeks) and each section

TABLE I
MAXIMAL RECOGNITION RATES (%) OF PCA, FLDA, LPP, SPP, SLDA, S2DLDP, AND SRC-DP UNDER SRC CLASSIFIER, CORRESPONDING DIMENSIONS AND CPU TIME (s) FOR TRAINING ON AR DATABASE

Method	PCA	FLDA	LPP	SPP	SLDA	S2DLDP	SRC-DP (P_0)	SRC-DP
Recognition rate	69.0	78.9	70.0	75.2	81.4	73.1	82.4	83.3
Dimension	110	90	150	150	110	400	140	150
Training time (s)	11.37	15.94	20.06	689.95	5.907×10^5	49.18	697.25	6.110×10^3

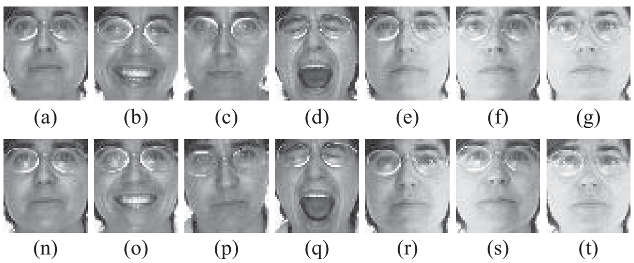

Fig. 2. (a)–(g) and (n)–(t) Samples of cropped images of one person in AR database.

contains 13 color images. Fourteen face images without occlusions (each session containing seven) of these 120 individuals are selected and used in our experiment. The face portion of each image is manually cropped and then normalized to 50×45 pixels. The sample images of one person are shown in Fig. 2. These images vary as follows: (a) neutral expression, (b) smiling, (c) angry, (d) screaming, (e) left light on, (f) right light on, (g) all sides light on, and (n)–(t) are taken under the same conditions as Fig. 2(a)–(g).

Images from the first session [i.e., Fig. 2(a)–(g)] are used for training, and images from the second session [i.e., Fig. 2(n)–(t)] are used for testing. Thus, the total number of training samples is 840. PCA (Eigenfaces [3]), Fisherfaces (FLDA [4]), Laplacianfaces (LPP [6]), (SPP [9]), (SLDA [28]), (S2DLDP [29]), and the proposed SRC-DP method are, respectively, used for feature extraction. In SRC-DP for all experiments, we use Algorithm 2 to obtain the initial solution and then use Algorithm 1 to get the optimal projection matrix $\mathbf{P}$. To avoid overfitting, we first perform PCA and reduce the dimension to be 200 before implementing FLDA, LPP, SPP, and SRC-DP. The K-nearest neighborhood parameter K in LPP is chosen as $l - 1$, where l is training sample size per class. Finally, the SRC classifier is employed for classification. Here in SRC and SRC-DP, the MATLAB function l1eq_pd from the l1-magic [18] is used to calculate the sparse representation coefficients. For all feature extraction methods mentioned, the SRC classifier is used for classification. The recognition rate curve of each 1-D method (except for S2DLDP) versus the variation of dimensions is shown in Fig. 3.[2] The maximal recognition rate of each method, the corresponding dimension and training time are shown in Table I. Additionally, we test the initial SRC-DP algorithm (Algorithm 2), notated by the SRC-DP ($\mathbf{P}_0$) here, and show its maximal recognition rate and the training time in Table I for comparison.

[2]Note that it is not convenient for us to show the results of S2DLDP in the figure because S2DLDP needs much more features to achieve its top recognition rate.

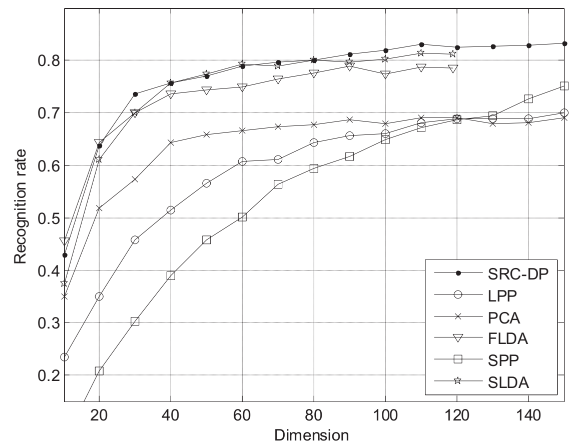

Fig. 3. Recognition rate curve of each method versus variation of dimensions.

From Fig. 3 SRC-DP consistently outperforms PCA, FLDA, LPP, SPP, and SLDA under the SRC classifier, when the dimension is over 90. Table I shows that the maximal recognition rate of SRC-DP is better than those of other methods. In terms of the CPU time for training, SRC-DP is nine times slower than SPP because it needs ten runs of iterations for updating the projection matrix P. However, if we prefer a lower computational time, we can apply the initialization method of SRC-DP, Algorithm 2, to determine the initial projection matrix $\mathbf{P}_0$ and use it directly for feature extraction. Table I shows SRC-DP ($\mathbf{P}_0$) is only slightly weaker than the iterative SRC-DP in performance, but the training time is significantly reduced. SRC-DP ($\mathbf{P}_0$) is as fast as SPP for training, yet remarkably improves the recognition performance of SPP. SLDA also achieves a nice recognition result in this experiment, but it is too computationally intensive for training. It costs almost 100 times of the CPU time in contrast to SRC-DP. S2DLDP is very fast but its performance is not very satisfying.

We would evaluate the performance of random initialization (random projection [2]). We use PCA to reduce the dimension to 200 and then generate a 200×150 random matrix. Using this matrix as projection matrix, the resulting features of each image are input into SRC and obtain the classification result. Fig. 4(a) shows the recognition rates corresponding to ten randomly generated matrices. We then use each of these ten random matrices as the initial solution $\mathbf{P}_0$ to input the SRC-DP algorithm, and obtain the result of SRC-DP with random initialization. The performance of SRC-DP corresponding to each initial solution (random matrix) is shown in Fig. 4(a) for comparison. The average recognition rate is shown

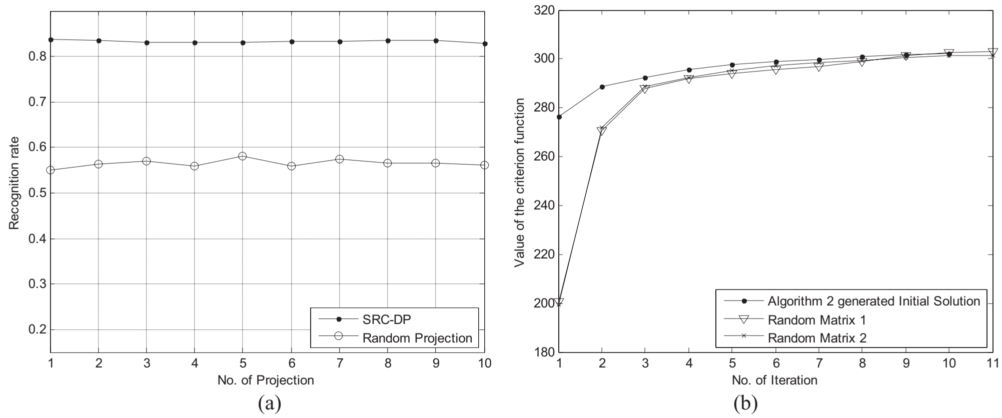

Fig. 4. (a) Performance comparison: random projection versus SRC-DP. (b) Convergence of SRC-DP algorithm.

TABLE II
COMPARISONS OF RANDOM INITIALIZATION AND PROPOSED INITIALIZATION METHOD AND THEIR EFFECT ON FINAL SOLUTION

Method	Random Initialization Average	Proposed Initialization	SRC-DP With Random Initialization Average	SRC-DP
Recognition rate	56.5 ± 0.82	82.4	83.3 ± 0.29	83.3

TABLE III
MAXIMAL RECOGNITION RATES (%) OF PCA, FLDA, LPP, SPP, SLDA, S2DLDP, AND FSRC-DP UNDER FSRC CLASSIFIER AND CORRESPONDING DIMENSIONS ON AR DATABASE

Method	PCA	FLDA	LPP	SPP	SLDA	S2DLDP	FSRC-DP (P_0)	FSRC-DP
Recognition rate	72.7	79.4	70.6	78.8	81.7	77.6	82.9	83.9
Dimension	130	119	130	150	119	400	120	150

in Table II. The proposed initialization method of SRC-DP, SRC-DP ($\mathbf{P}_0$), is significantly better than the random initialization. However, an interesting finding is that different initialization methods have little effect on the final result of SRC-DP. SRC-DP with random initialization achieves almost the same results as the SRC-DP with the proposed initialization method. In addition, Fig. 4(a) shows that the performance of SRC-DP is also insensitive to variations of the initial solution.

The convergence of the SRC-DP algorithm associated with the initial $\mathbf{P}_0$ determined by Algorithm 2 and two randomly generated matrices is shown in Fig. 4(b). It seems that the criterion function of our algorithm converges well, independent of the choice of initial solution. However, using the initial $\mathbf{P}_0$ obtained by Algorithm 2 might speedup the convergence as $\mathbf{P}_0$ yields a better criterion value. The convergence speed of SRC-DP algorithm is fast; it always converges around ten times of iterations.

Let us further examine the performance of the FSRC-DP. To obtain the solution of the model in (16), we use the MATLAB function l1_ls provided by Kim *et al.* [19]. The classification result of FSRC-DP is shown in Table III. For comparison, the results of PCA, FLDA, LPP, and SPP under the same FSRC classifier are also shown in the table. Comparing with the results in Table I, we see that using the flexible sparse representation model in (16) instead of the exact model in (2) can improve the performance of the SRC classifier and the SRC-DP method.

TABLE IV
COMPARISONS OF IMAGE-BASED AND SRC-DP (OR SRC-DP) FEATURE-BASED CLASSIFICATION

Method	Image-Based SRC	Image-Based FSRC	SRC-DP	FSRC-DP
Recognition rate	67.3	78.6	83.3	83.9
Dimension	2250	2250	150	150
Testing time (s)	8873.2	18374.1	466.5	546.5

To highlight the role of SRC-DP-based feature extraction, we would like to compare the method with the image-based SRC or FSRC, i.e., applying SRC or FSRC directly on original image vectors (without any feature extraction step), and show the results in Table IV. From Table IV, the SRC-DP (or FSRC-DP) improves the performance of the image-based SRC (or FSRC). In addition, the feature extraction method significantly accelerates the classification process, as the SRC-DP only uses 150 features, which is much smaller than the dimension of image space.

We would like to assess the performance of the six methods mentioned using the strategy of ten-fold cross validation. We randomly choose K samples from each class for training, whereas the remaining samples for testing. Let K vary from four to seven. For each K, we run the system ten times and obtain ten different training and testing sample sets for performance evaluation. Here, for improving

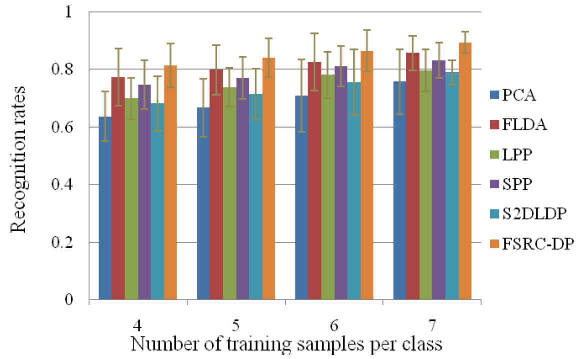

Fig. 5. Illustration of average recognition rates (%) and std of PCA, FLDA, LPP, SPP, S2DLDP, and FSRC-DP under FSRC classifier using ten-run test on AR database.

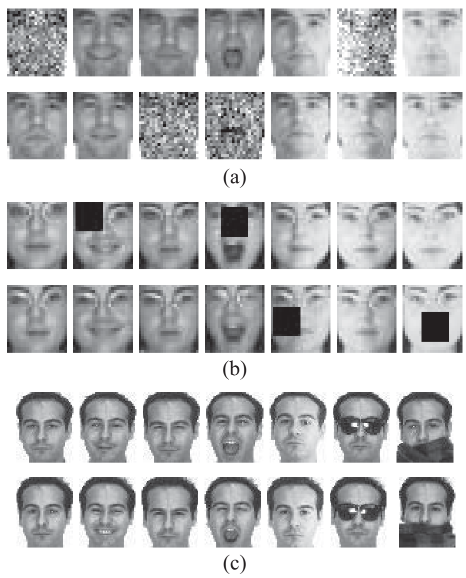

Fig. 6. Examples of training and testing images of one person. (a) Artificial corruption. (b) Artificial occlusion. (c) Real occlusion.

the computational efficiency, we use the efficient Euclidean projections algorithm (the MATLAB function LeastR in the SLEP package) [22], [23] to solve the Lasso model in (16). The average recognition rates and the standard deviations (std) of PCA, FLDA, LPP, SPP, S2DLDP, and FSRC-DP under the FSRC classifier across ten tests are shown in Fig. 5. The results in Fig. 5 are generally consistent with those in Tables I and III. FSRC-DP outperforms others irrespective of the variation of training sample size.

Finally, we compare the computational efficiency of the proposed FSRC-DP (or SRC-DP) using three different sparse representation algorithms: l1eq_pd from the l1-magic [18], l1_ls [19], and LeastR from the SLEP package [23]. We randomly choose seven samples from each class for training and the remaining for testing. The average CPU time for training and testing of ten runs are shown in Table V. The average iteration times of SRC-DP using l1eq_pd is around ten, whereas that of FSRC-DP using l1_ls or LeastR is around five. From this table, LeastR is the most efficient one among the three methods. It is ten times faster than l1_ls, and 35 times faster than l1eq_pd for training. It is almost 50 times faster than the other two for testing. Therefore, using LeastR can significantly improve the computational efficiency of FSRC-DP (or SRC-DP).

TABLE V
AVERAGE CPU TIME FOR TRAINING AND TESTING OF FSRC-DP (OR SRC-DP) WHEN NUMBER OF TRAINING SAMPLES PER CLASS IS SEVEN USING DIFFERENT SPARSE REPRESENTATION ALGORITHMS

SR Algorithms	l1eq_pd [18]	l1_ls [19]	LeastR [23]
Training time	6.072×10^3	1.780×10^3	175.15
Testing time	479.86	492.43	9.87

TABLE VI
COMPARISONS OF IMAGE-BASED AND SRC-DP (OR SRC-DP) FEATURE-BASED CLASSIFICATION IN CORRUPTION AND OCCLUSION

Method	Image-Based Extended SRC	Robust PCA-Based Extended SRC	Extended SRC-DP
Artificial corruption	69.5	64.3	74.3
Artificial occlusion	57.4	49.8	59.4
Real occlusion	77.9	70.1	79.4

B. Experiment on Occluded/Corrupted Face Images

To evaluate the extended SRC-DP, we experiment on the AR database with artificial corruption, artificial occlusion, and real occlusion. Specifically, to generate artificial corruption, we use the training and testing images of each person as adopted in the experiment in Section IV-A, we randomly choose two training images and two testing images, and add Gaussian white noise (zero-mean and the variance is 0.03) into them. Similarly, to generate artificial occlusion, we randomly choose two training images and two testing images from each person and add a black-square into them. The black-square is randomly located in the face image. For real occlusion, we use one image with glasses and one image with scarf in the training set and testing set, respectively. Fig. 6 shows examples of training and testing images of one person with corruption and occlusion.

We use the extended SRC-DP that is described in Section III-G for feature extraction and the FSRC for classification. In comparison, we perform image-based extended SRC, i.e., applying the extended SRC [by solving the extended L1 optimization problem in (21)] to face image vectors directly, and robust PCA [26]-based extended SRC, i.e., apply the robust PCA [26] to remove corruption (corruption) in training sample images and then use the extended SRC for testing. The recognition results are shown in Table VI. The extended SRC-DP-based feature extraction is more effective for improving the performance of the extended SRC, irrespective of whether it is artificial corruption (occlusion) or real occlusion. However, it should be mentioned that the extended SRC-DP is more time-consuming than the extended SRC for test, because it involves an operation in image space, i.e., solving the extended L_1 optimization problem as it is done in the extended SRC.

C. Experiment Using the Extended Yale B Database

The extended Yale B face database [20] contains 38 human subjects under nine poses and 64 illumination conditions.

Fig. 7. Samples of person under different illuminations in extended Yale B face database.

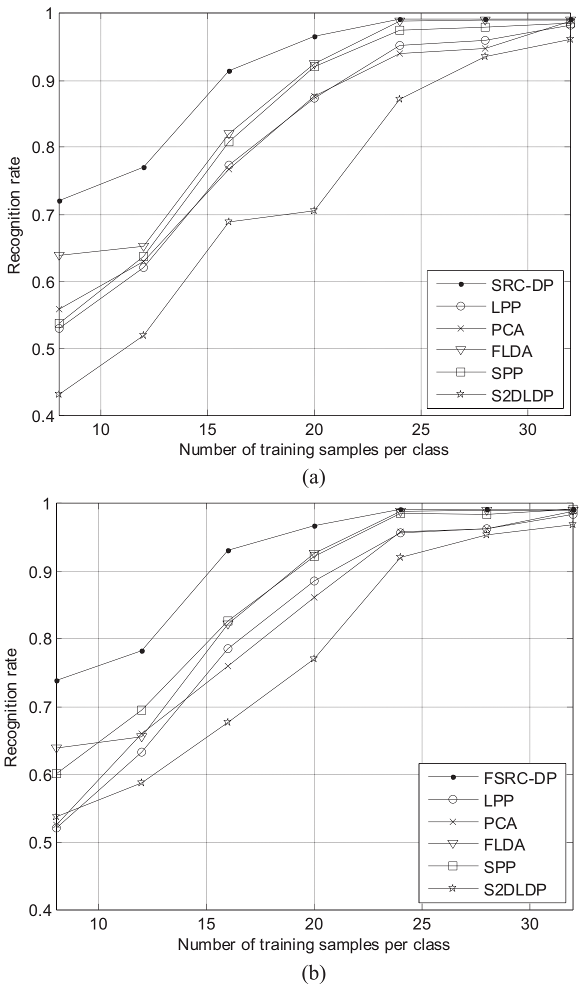

Fig. 8. Illustration of recognition rates of PCA, FLDA, LPP, SPP, S2DLDP, and SRC-DP (or FSRC-DP) with variations of training sample size on extended Yale B database. (a) Results under SRC. (b) Results under FSRC.

The 64 images of a subject in a particular pose are acquired at camera frame rate of 30 frames/s, so there is only small change in head pose and facial expression for those 64 images. All frontal-face images marked with P00 are used, and each image is resized to 42×48 pixels in our experiment. Some sample images of one person are shown in Fig. 7.

In the first experiment, we use the first 8, 12, 16, 20, 24, 28, and 32 images per subject, respectively, for training, and the remaining images for testing. PCA, FLDA, LPP, SPP, S2DLDP, and the proposed SRC-DP method are used for feature extraction. Before implementing FLDA, LPP, SPP, and SRC-DP, we use PCA to reduce the dimension to be 100, 120, 140, 160, 180, 200, and 220 based on different number of training samples per class. Finally, the SRC classifier is employed for classification. The recognition rate of each method is shown in Fig. 8(a). In contrast, the results of PCA, FLDA, LPP, SPP, S2DLDP, and FSRC-DP under the FSRC classifier are shown in Fig. 8(b).

TABLE VII
CPU TIME (s) FOR TRAINING WHEN NUMBER OF TRAINING SAMPLES PER CLASS IS 16 ON YALE B DATABASE

PCA	FLDA	LPP	SPP	S2DLDP	FSRC-DP
16.53	20.56	18.94	20.79	36.91	66.35

Fig. 8 shows that when the number of training samples per class is relative small (e.g., 8, 12, 16, and 20), SRC-DP (or FSRC-DP) significantly outperforms PCA, FLDA, LPP, SPP, and S2DLDP. When the training sample size becomes large, SRC-DP is still slightly better than the other methods. This makes sense for real-world face recognition, because there are generally very limited face images available for training in practice. By comparing Fig. 8(a) and (b), the flexible sparse representation model can further improve the performance of the SRC-DP method when the training sample size is relative small. Table VII shows the CPU time for training of each method when the number of training samples per class is 16. FSRC-DP uses LeastR from the SLEP package [23] for solving the Lasso model in (16). FSRC-DP needs three iterations. In each iteration step, FSRC-DP is as fast as FLDA.

In the second experiment, we perform the ten-fold cross validation by partitioning class samples in different ways. We randomly choose K samples from each class for training, whereas the remaining samples for testing. Here, we allow K to vary from 8 to 32 with interval of four. For each K, we perform ten runs of tests for each of the six methods: PCA, FLDA, LPP, SPP, S2DLDP, and FSRC-DP. The average recognition rates and the stds of each method under the FSRC classifier across ten tests are shown in Fig. 9. Fig. 9 shows that our method FSRC-DP consistently performs better than the other methods, irrespective of the variation of training sample size. In general, the performance difference between FSRC-DP and others become less and less significant with the increase of the training sample size K.

D. Experiment Using the PIE Database

The CMU PIE face database contains 68 subjects with over 40 000 face images [24]. Images of each person are taken across 13 different poses, under 43 different illumination conditions, and with four different expressions. Here we use a subset containing images of pose C05 (a nearly frontal pose) of 68 persons, each with 49 images. All images are manually aligned, cropped, and resized to be 64×64 pixels [25] in our experiments.

Using the same methodology as adopted in the foregoing experiment, we perform the ten-fold cross validation. We randomly choose K = 10, 15, 20, and 25 images from each class for training, and the remaining images for test. For each K, we perform ten runs of tests for each of the six methods mentioned under the FSRC classifier. The average recognition rates and the stds of each method across ten tests are shown in Fig. 10. Fig. 10 shows that our

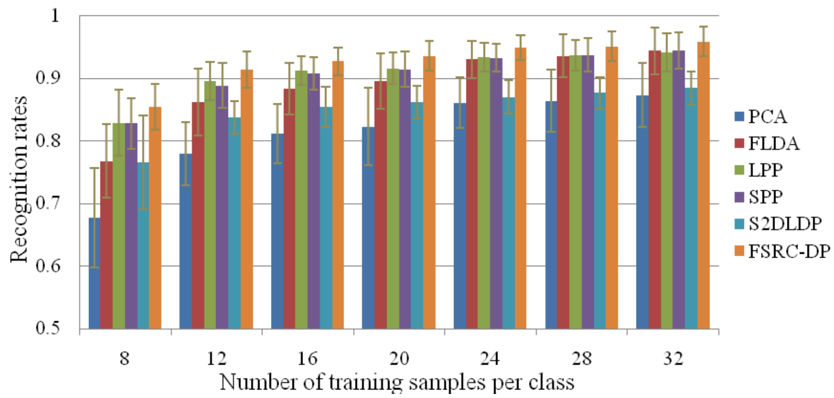

Fig. 9. Illustration of average recognition rates (%) and std of PCA, FLDA, LPP, SPP, S2DLDP, and FSRC-DP under FSRC classifier using ten-run test on extended Yale B database.

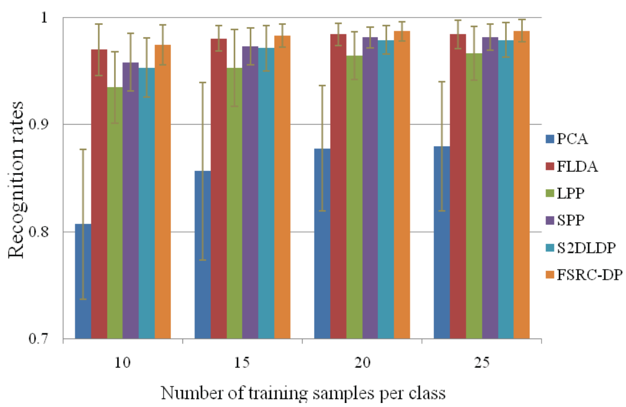

Fig. 10. Illustration of average recognition rates (%) and std of PCA, FLDA, LPP, SPP, S2DLDP, and FSRC-DP under FSRC classifier using ten-run test on PIE database.

TABLE VIII
AVERAGE CPU TIME (s) FOR TRAINING WHEN NUMBER OF TRAINING SAMPLES PER CLASS IS 20 ON PIE DATABASE

PCA	FLDA	LPP	SPP	S2DLDP	FSRC-DP
23.06	74.21	37.58	57.14	73.52	274.46

method FSRC-DP achieves the best results among all methods, irrespective of the variation of training sample size. However, it should be mentioned that on this database, the FLDA also performs very well. The performance difference between FSRC-DP and FLDA seems not significant in the average sense. Table VIII shows the CPU time for training of each method when the number of training samples per class is ten. FSRC-DP uses LeastR from the SLEP package [23] for solving the Lasso model in (16). FSRC-DP needs four iterations in average. In each iteration step, FSRC-DP is as fast as FLDA. FSRC-DP is certainly more time-consuming than FLDA as it needs iterations.

V. CONCLUSION

This paper presented a paradigm of linking dimensionality reduction to classification: seeking a low-dimensional space of data in which the SRC achieved better performance and became more efficient. The decision rule of SRC was used to direct the design of a dimensionality reduction method—SRC-DP. SRC-DP fitted SRC well in spirit, so the SRC-DP-based feature extractor and the SRC can be seamlessly integrated into a face recognition system. Experiments were done on the AR, the extended Yale B, and PIE face image databases, and results demonstrated the performance advantage of the proposed method over others.

If face images were corrupted or occluded, a prior corruption (occlusion)-removing (or image recovering) step was necessary before the application of SRC-DP. From Wright *et al.* [2], we assumed the representation coefficients and corrupted (occluded) pixels were both sparse and compute them simultaneously by solving the extended sparse representation model. We embedded this method into our SRC-DP and provided a mechanism to deal with occlusion or corruption in training and testing images. Our future work is to build a unified framework for corruption removing, feature extraction and classification by combining sparse representation and low-rank approximation.

REFERENCES

[1] W. Zhao, R. Chellappa, A. Rosenfeld, and P. J. Phillips, "Face recognition: A literature survey," *ACM Comput. Surv.*, vol. 35, no. 4, pp. 399–458, Dec. 2003.

[2] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, Feb. 2009.

[3] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 1991, pp. 586–591.

[4] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriengman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 711–720, Jul. 1997.

[5] T.-K. Kim and J. Kittler, "Locally linear discriminant analysis for multimodally distributed classes for face recognition with a single model image," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 3, pp. 318–327, Mar. 2005.

[6] X. He, S. Yan, Y. Hu, P. Niyogi, and H.-J. Zhang, "Face recognition using laplacianfaces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 3, pp. 328–340, Mar. 2005.

[7] H.-T. Chen, H.-W. Chang, and T.-L. Liu, "Local discriminant embedding and its variants," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.* 2005, pp. 846–853.

[8] S. Yan, D. Xu, B. Zhang, H. Zhang, Q. Yang, and S. Lin, "Graph embedding and extension: A general framework for dimensionality reduction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 1, pp. 40–51, Jan. 2007.

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

[10] D. Cai, X. He, J. Han, and H.-J. Zhang, "Orthogonal laplacianfaces for face recognition," *IEEE Trans. Image Process.*, vol. 15, no. 11, pp. 3608–3614, Nov. 2006.

[11] D. Cai, X. He, Y. Hu, J. Han, and T. Huang, "Learning a spatially smooth subspace for face recognition," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–7.

[12] D. Donoho, "For most large underdetermined systems of linear equations the minimal ℓ_1-norm solution is also the sparsest solution," *Commun. Pure Appl. Math.*, vol. 59, no. 6, pp. 797–829, 2006.

[13] E. Candès, J. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Comm. Pure Applied Math.*, vol. 59, no. 8, pp. 1207–1223, 2006.

[14] S. Chen, D. Donoho, and M. Saunders, "Atomic decomposition by basis pursuit," *SIAM Rev.*, vol. 43, no. 1, pp. 129–159, 2001.

[15] D. Donoho and Y. Tsaig, "Fast solution of ℓ_1 norm minimization problems when the solution may be sparse," *IEEE Trans. Inf. Theory*, vol. 54, no. 11, pp. 4789–4812, Nov. 2008.

[16] S. Kaski, "Dimensionality reduction by random mapping: Fast similarity computation for clustering," in *Proc. Int. Joint Conf. Neural. Netw.*, vol. 1. 1998, pp. 413–418.

[17] A. M. Martinez and R. Benavente, "The AR face database," CVC, Univ. Autonoma Barcelona, Barcelona, Spain, Tech. Rep. #24, Jun. 1998.

[18] E. Candès and J. Romberg. (2005). *ℓ_1-Magic: Recovery of Sparse Signals via Convex Programming* [Online]. Available: http://www.acm.caltech.edu/l1magic/

[19] S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, "An interior-point method for large-scale ℓ_1-regularized least squares," *IEEE J. Sel. Topics Signal Process.*, vol. 1, no. 4, pp. 606–617, Dec. 2007.

[20] K. C. Lee, J. Ho, and D. Driegman, "Acquiring linear subspaces for face recognition under variable lighting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 5, pp. 684–698, May 2005.

[21] Z. Jin, J. Y. Yang, Z. S. Hu, and Z. Lou, "Face recognition based on uncorrelated discriminant transformation," *Pattern Recognit.*, vol. 33, no. 7, pp. 1405–1416, 2001.

[22] J. Liu and J. Ye, "Efficient Euclidean projections in linear time," in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, 2009, pp. 657–664.

[23] J. Liu and J. Ye. (2008). *SLEP: A Sparse Learning Package* [Online]. Available: http://www.public.asu.edu/~jye02/Software/SLEP/

[24] T. Sim, S. Baker, and M. Bsat, "The CMU pose, illumination, and expression database," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 12, pp. 1615–1618, Dec. 2003.

[25] D. Cai, X. He, and J. Han, "Spectral regression for efficient regularized subspace learning," in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, Oct. 2007, pp. 1–8.

[26] J. Wright, A. Ganesh, S. Rao, Y. Peng, and Y. Ma, "Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization," in *Advances in Neural Information Processing Systems*. New York, USA: MIT Press, Dec. 2009.

[27] Y. Xu, D. Zhang, J. Yang, and J.-Y. Yang, "A two-phase test sample sparse representation method for use with face recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 9, pp. 1255–1262, Sep. 2011.

[28] L. Clemmensen, T. Hastie, D. Witten, and B. Ersboll, "Sparse discriminant analysis," *Technometrics*, vol. 53, no. 4, pp. 406–413, 2011.

[29] Z. Lai, M. Wan, Z. Jin, and J. Yang, "Sparse two-dimensional local discriminant projections for feature extraction," *Neurocomputing*, vol. 74, no. 4, pp. 629–637, 2011.

[30] Z. Lai, W. K. Wong, Z. Jin, J. Yang, and Y. Xu, "Sparse approximation to the eigen subspace for discrimination," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 12, pp. 1948–1960, Dec. 2012.

[31] M. Yang, L. Zhang, J. Yang, and D. Zhang, "Robust sparse coding for face recognition," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 625–632.

[32] J. Yang, L. Zhang, Y. Xu, and J.-Y. Yang, "Beyond sparsity: The role of L_1-optimizer in pattern classification," *Pattern Recognit.*, vol. 45, no. 3, pp. 1104–1118, 2012.

[33] J. Yang, A. F. Frangi, J.-Y. Yang, D. Zhang, and Z. Jin, "KPCA plus LDA: A complete kernel fisher discriminant framework for feature extraction and recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 2, pp. 230–244, Feb. 2005.

[34] H. Cevikalp, M. Neamtu, and M. Wilkes, "Discriminative common vector method with kernels," *IEEE Trans. Neural Netw.*, vol. 17, no. 6, pp. 1550–1565, Nov. 2006.

[35] S. Zafeiriou, G. Tzimiropoulos, M. Petrou, and T. Stathaki, "Regularized kernel discriminant analysis with a robust kernel for face recognition and verification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 3, pp. 526–534, Mar. 2012.

[36] Z. Fan, Y. Xu, and D. Zhang, "Local linear discriminant analysis framework using sample neighbors," *IEEE Trans. Neural Netw.*, vol. 22, no. 7, pp. 1119–1132, Jul. 2011.

[37] S.-J. Wang, J. Yang, M.-F. Sun, X.-J. Peng, M.-M. Sun, and C.-G. Zhou, "Sparse tensor discriminant color space for face verification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 6, pp. 876–888, Jun. 2012.

[38] J. Yang, D. Chu, and J.-Y. Yang, "Sparse representation classifier steered discriminative projection," in *Proc. Int. Conf. Pattern Recognit.*, Aug. 2010, pp. 694–697.

[39] R. He, B.-G. Hu, W.-S. Zheng, and Y. Guo, "Two-stage sparse representation for robust recognition on large-scale database," in *Proc. 24th AAAI Conf. Artif. Intell.*, 2010, pp. 475–480.

[40] R. He, W.-S. Zheng, B.-G. Hu, and X. W. Kong, "A regularized correntropy framework for robust pattern recognition," *Neural Comput.*, vol. 23, no. 8, pp. 2074–2100, 2011.

[41] R. He, W.-S. Zheng, and B.-G. Hu, "Maximum correntropy criterion for robust face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1561–1576, Aug. 2011.

[42] R. He, B.-G. Hu, X. T. Yuan, and W.-S. Zheng, "Principal component analysis based on nonparametric maximum entropy," *Neurocomputing*, vol. 73, nos. 10–12, pp. 1840–1952, 2010.

[43] R. He, B.-G. Hu, and X. T. Yuan, "Robust discriminant analysis based on non-parametric maximum entropy," in *Proc. Asian Conf. Mach. Learn.*, 2009, pp. 120–134.

[44] L.M. Zhang, S. Chen, and L. Qiao, "Graph optimization for dimensionality reduction with sparsity constraints," *Pattern Recognit.*, vol. 45, no. 3, pp. 1205–1210, 2012.

[45] L. Zhang, M. Yang, and X. C. Feng, "Sparse representation or collaborative representation: Which helps face recognition?" in *Proc. IEEE Int. Conf. Comput. Vis.*, Nov. 2011, pp. 471–478.

Jian Yang (M'08) received the B.S. degree in mathematics from the Xuzhou Normal University, Xuzhou, China, in 1995, the M.S. degree in applied mathematics from the Changsha Railway University, Changsha, China, in 1998, and the Ph.D. degree in pattern recognition and intelligence systems from the Nanjing University of Science and Technology (NUST), Nanjing, China, in 2002.

He was a Post-Doctoral Researcher with the University of Zaragoza, Zaragoza, Spain, in 2003. From 2004 to 2006, he was a Post-Doctoral Fellow with the Biometrics Centre of Hong Kong Polytechnic University, Hong Kong. From 2006 to 2007, and with the Department of Computer Science, New Jersey Institute of Technology, Newark, NJ, USA. He is currently a Professor with the School of Computer Science and Technology, NUST. He has authored more than 80 scientific papers in pattern recognition and computer vision. His journal papers have been cited more than 1800 times in the ISI Web of Science, and 3000 times in the Web of Scholar Google. His current research interests include pattern recognition, computer vision and machine learning.

Dr. Yang is currently an Associate Editor of Pattern Recognition Letters and the IEEE Transactions on Neural Networks and Learning Systems.

Delin Chu received the Ph.D. degree from the Department of Applied Mathematics, Tsinghua University, Beijing, China, in 1991.

He is currently with the Department of Mathematics, National University of Singapore, Singapore, as an Associate Professor. His current research interests include numerical linear algebra, scientific computing and numerical analysis, and matrix theory and computations. He has authored more than fifty papers on international journals.

Dr. Chu is currently an Associate Editor of Automatica and Advances in Numerical Analysis.

Lei Zhang (M'04) received the B.S. degree from the Shenyang Institute of Aeronautical Engineering, Shenyang, China, in 1995, the M.S. and Ph.D. degrees in automatic control theory and engineering from Northwestern Polytechnical University, Xi'an, China, in 1998 and 2001, respectively.

He was a Research Associate with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong, from 2001 to 2002. From 2003 to 2006, he was a Post-Doctoral Fellow with the Department of Electrical and Computer Engineering, McMaster University, Hamilton, ON, Canada. Since 2006, he has been an Associate Professor with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong. His current research interests include image and video processing, biometrics, pattern recognition, multisensor data fusion and optimal estimation theory.

Dr. Zhang is currently an Associate Editor of the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART C: APPLICATIONS AND REVIEWS.

Yong Xu (M'06) was born in Sichuan, China, in 1972. He received the B.S. and M.S. degrees in 1994 and 1997, respectively, and the Ph.D. degree in pattern recognition and intelligence system from the Nanjing University of Science and Technology, Nanjing, China, in 2005. Currently, he is with the Bio-Computing Research Center, Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen, China. His current research interests include pattern recognition, biometrics, machine learning, image processing, and video analysis.

Jingyu Yang received the B.S. degree in computer science from the Nanjing University of Science and Technology (NUST), Nanjing, China.

He was a Visiting Scientist with the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, Urbana, IL, USA, from 1982 to 1984. From 1993 to 1994, he was a Visiting Professor with the Department of Computer Science, Missuria University, Columbia, MI, USA. In 1998, he acted as a Visiting Professor with Concordia University, Montreal, QC, Canada. He is currently a Professor and Chairman with the Department of Computer Science, NUST. He has authored over 300 scientific papers in computer vision, pattern recognition, and artificial intelligence. His current research interests include pattern recognition, robot vision, image processing, data fusion, and artificial intelligence.

Mr. Yang was the recipient of more than 20 provincial awards and national awards.