Integrated Low-Rank-Based Discriminative Feature Learning for Recognition

Pan Zhou, Zhouchen Lin, *Senior Member, IEEE*, and Chao Zhang, *Member, IEEE*

***Abstract*— Feature learning plays a central role in pattern recognition. In recent years, many representation-based feature learning methods have been proposed and have achieved great success in many applications. However, these methods perform feature learning and subsequent classification in two separate steps, which may not be optimal for recognition tasks. In this paper, we present a supervised low-rank-based approach for learning discriminative features. By integrating latent low-rank representation (LatLRR) with a ridge regression-based classifier, our approach combines feature learning with classification, so that the regulated classification error is minimized. In this way, the extracted features are more discriminative for the recognition tasks. Our approach benefits from a recent discovery on the closed-form solutions to noiseless LatLRR. When there is noise, a robust Principal Component Analysis (PCA)-based denoising step can be added as preprocessing. When the scale of a problem is large, we utilize a fast randomized algorithm to speed up the computation of robust PCA. Extensive experimental results demonstrate the effectiveness and robustness of our method.**

***Index Terms*— Feature learning, low-rank representation (LRR), recognition, robust principal component analysis (PCA).**

I. Introduction

FEATURE learning is a critical step for almost all recognition tasks, such as image classification and face recognition. There has been a lot of work [1]–[10] focusing on learning discriminative features. For example, Belhumeur *et al.* [1] projected the image space to a low-dimensional subspace based on Fisher's linear discriminant and produce well-separated features. Lazebnik *et al.* [2], Huang *et al.* [3], and Liu *et al.* [4] viewed distances between samples as features and Wang *et al.* [6] projected each descriptor into a local-coordinate system as a feature.

Recently, representation-based feature learning methods [5], [7], [8], [11]–[14] have drawn a lot of attention. The first representation-based method may be sparse representation classification (SRC) [5]. SRC finds the smallest number of training samples to represent a test sample and adopts the representation coefficients as a feature vector of the test sample. It is reported that SRC achieves surprisingly high accuracy in face recognition even under occlusion [5]. Unfortunately, SRC breaks down when the training data are wildly corrupted, e.g., under unreasonable illumination or pose. To overcome this drawback, a series of low-rank representation (LRR)-based feature learning methods [7], [8], [11]–[14] have been proposed. These methods assume that the samples in the same class should be located in the same low-dimensional subspace. Since the dimension of the subspace corresponds to the rank of the representation matrix, these methods enforce a low-rank constraint on the representation matrix, and thus enhancing the correlation among the representation coefficient vectors. As a result, these methods have achieved great success in a lot of recognition problems, such as face and object recognitions.

However, most existing representation-based methods consist of two separate steps: 1) extracting discriminative features by learning from training data and 2) inputting the features into a specific classifier for classification. Such separation may limit the overall recognition performance. To overcome this problem, in this paper, we propose a simultaneous feature learning and data classification method, by integrating latent LRR (LatLRR) with a ridge regression-based classifier. LatLRR is a recently proposed method for unsupervised feature clustering and learning [15]. We choose LatLRR because when there is no noise it has nonunique closed-form solutions [16], which is remarkable among all representation-based methods.

The contributions of this paper are as follows.

1) We propose a simple yet effective model for simultaneous feature learning and data classification. By integrating the closed-form solutions to LatLRR with a ridge regression-based classifier, our model achieves an overall optimality in recognition in some sense.
2) While most existing representation-based methods minimize the sparsity or the rank of some solutions related to feature learning, which is not directly

Manuscript received September 8, 2014; revised January 21, 2015 and May 3, 2015; accepted May 16, 2015. Date of publication June 11, 2015; date of current version April 15, 2016. The work of P. Zhou and C. Zhang were supported in part by the National Key Basic Research Project of China (973 Program) under Grant 2015CB352303 and Grant 2011CB302400 and in part by the National Natural Science Foundation (NSF) of China under Grant 61071156 and Grant 61131003. The work of Z. Lin was supported in part by the 973 Program of China under Grant 2015CB352502, in part by NSF of China under Grant 61231002 and Grant 61272341, and in part by the Microsoft Research Asia Collaborative Research Program. *(Corresponding author: Zhouchen Lin.)*

The authors are with the Key Laboratory of Machine Perception (Ministry of Education), School of Electrical Engineering and Computer Science, Peking University, Beijing 100871, China, and also with Cooperative Medianet Innovation Center, Shanghai 200240, China (e-mail: pzhou@pku.edu.cn; zlin@pku.edu.cn; chzhang@cis.pku.edu.cn).

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org (File size: 1 MB).

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.2015.2436951

TABLE I
SUMMARY OF NOTATIONS FREQUENTLY USED IN THIS PAPER

Notation	Meaning
capital letter	A matrix. Especially, I is the identity matrix.
M^T	Transpose of matrix M.
M_{ij}	The (i,j)-th entry of matrix M.
M_i	The i-th column of matrix M.
x_i	The i-th entry of vector x.
$\|\|\cdot\|\|_*$	Nuclear norm, the sum of all the singular values.
$\|\|\cdot\|\|_1$	$\|\|M\|\|_1 = \sum_{i,j} \|M_{ij}\|$.
$\|\|\cdot\|\|_F$	Frobenious norm, $\|\|M\|\|_F = \sqrt{\sum_{i,j} M_{ij}^2}$.
$\|\|\cdot\|\|_2$	Vector Euclidean norm, $\|\|x\|\| = \sqrt{\sum_i x_i^2}$.
diag$(\cdot)$	The diagonal entries of a matrix.

connected to the subsequent recognition problem, our model directly minimizes the regularized classification error. As a consequence, our method achieves higher accuracy in recognition.

3) Due to the closed-form solutions to LatLRR, our feature learning approach is fast. When there is noise, we propose to denoise the data with a robust Principal Component Analysis (PCA) [17] first. We also incorporate a fast randomized algorithm for solving the robust PCA when the rank is relatively low compared with the matrix size. As a consequence, our method also excels in speed when the scale of a problem is large.

Extensive experimental results testify the advantages of our method. Note that the idea of incorporating empirical error into specific learning tasks has appeared before. For example, Argyriou *et al.* [18] incorporated an empirical error into multitask feature learning. Mairal *et al.* [19] proposed a task-driven dictionary learning (TDDL) method, which incorporated the empirical error into the dictionary learning. There are other similar works, such as [20] and [21]. By comparison, we focus on integrating empirical error with representation-based discriminative feature learning, which is different from prior work.

The remainder of this paper is organized as follows. Section II reviews related work on the existing representation-based feature learning methods for classification. In Section III, we present our integrated a low-rank-based feature learning method, which integrates the closed-form solutions to LatLRR with a ridge regression and utilizes the labels of data to learn discriminative feature on clean data. The way to handling corrupted data by a robust PCA and the fast randomized algorithm for the robust PCA are also presented. Section IV presents the experimental results and analysis. Finally, Section V concludes this paper and discusses the future work.

II. RELATED WORK

In this section, we review the existing representation-based feature learning methods. For brevity, we summarize some main notations in Table I. We further denote the training data matrix as $X = [X_1, X_2, \ldots, X_s] \in R^{d\times m}$, where $X_i \in R^{d\times m_i}$ is the data submatrix of class i and $m = \sum_{i=1}^{s} m_i$. We also denote the dictionary as $D = [D_1, D_2, \ldots, D_k] \in R^{d\times n}$, where $D_i \in R^{d\times n_i}$ is the subdictionary associated with the ith class and $n = \sum_{i=1}^{k} n_i$.

A. Sparse Representation-Based Feature Learning

SRC [5] may be the first representation-based method. The main idea of SRC is to represent the input sample $y \in R^d$ as a linear combination of a few atoms in an overcomplete dictionary D. The corresponding sparse representation $\alpha \in R^n$ can be computed by the following ℓ_1 minimization problem:

$$\min_{\alpha} \|y - D\alpha\|_2^2 + \lambda\|\alpha\|_1. \tag{1}$$

Suppose that $\alpha = [\alpha_1^T, \alpha_2^T, \ldots, \alpha_k^T]^T$, and α_i is the subvector associated with the dictionary D_i of the ith class. A test sample y is classified as class j^* if class j^* results in the least reconstruction error

$$j^* = \arg\min_j \|y - D_j\alpha_j\|_2^2. \tag{2}$$

Although such a sparse coding method has achieved great success in face recognition, it requires the atoms in the dictionary to be well aligned for a reconstruction purpose, which is not always satisfied. Several methods have been proposed to resolve this issue. Wagner *et al.* [22] proposed an extended SRC to handle variations of faces in illumination, alignment, pose, and occlusion. Zhang *et al.* [12], Yang and Zhang [23], and Yang *et al.* [24] also extended SRC to deal with outliers, such as occlusions in face images. In their methods, collaborative representation-based classification (CRC) [12] achieved a much higher face recognition rate. However, when all the data (both training and testing images) are corrupted, these methods do not work well [7], [8]. Furthermore, sparse coding methods represent each test sample independently. This mechanism does not take a full advantage of structural information from the data set [7], [8], [25]. Actually, data from the same class may share common (correlated) features. Therefore, Jenatton *et al.* [25] utilized the structure of data to encourage a group sparse representation.

B. Low-Rank Representation-Based Feature Learning

Before we introduce the LRR-based feature learning methods, we first introduce the robust PCA [17], since some methods, including ours, are based on or related to it.

Robust PCA is a low-rank matrix recovery method. It aims to decompose a data matrix X into $A + E$, where A is a low-rank matrix that we want to recover, which stands for the clean data lying on a low-dimensional subspace, and E is a sparse error matrix. The separation is achieved by solving the following principal component pursuit problem [17]:

$$\min_{A,E} \|A\|_* + \lambda\|E\|_1, \quad \text{s.t. } X = A + E \tag{3}$$

where the nuclear norm and the ℓ_1 norm are convex surrogates of the rank function and the ℓ_0 pseudonorm, i.e., the number of nonzero entries, respectively. λ is a positive parameter tradingoff between low rankness and sparsity. A can be exactly recovered from X as long as the rank of A is sufficiently low and E is sufficiently sparse [17].

A low-rank matrix recovery with a structural incoherence-based classification (LRSIC) method [7] is a feature learning method based on the robust PCA [17]. It uses the robust PCA to decompose the training data matrix $X = [X_1, X_2, \ldots, X_s]$

into a low-rank matrix $A = [A_1, A_2, \ldots, A_s]$ and a sparse error matrix $E = [E_1, E_2, \ldots, E_s]$, where X_i is the training data matrix for class i, and A_i and E_i are decompositions of X_i. To remove the noise in data and reduce the feature dimension, Chen *et al.* [7] applied PCA to A to obtain a projection matrix W, and then projected both training data and testing data with W. Finally, they used SRC [5] for classification. This method is called the low-rank matrix recovery-based classification (LRC) method. To promote discriminating ability of the LRC method, structure incoherence is considered. The model of the LRSIC (LRC with structure incoherence) can be written as follows:

$$\min_{A,E} \sum_{i=1}^{s} (\|A_i\|_* + \|E_i\|_1) + \eta \sum_{j \neq i} \left\|A_j^T A_i\right\|_F^2$$
$$\text{s.t. } X_i = A_i + E_i, \quad i = 1, 2, \ldots, s \tag{4}$$

where η is a positive parameter. Then, similar to LRC, PCA and SRC are used for classification. However, Zhang *et al.* [8] pointed out that these two methods could not preserve structural information well. Moreover, these two methods require a removal of noise from training samples class by class. This process is computationally expensive when the number of classes is large [8].

Structured LRR for classification (SLRRC) [8] is another low-rank-based feature learning method. It first learns a structured low-rank dictionary by introducing an ideal coding-based regularization term. Then with the learned dictionary, it learns a sparse and structured representation for an image classification. More specifically, suppose that $D = [D_1, D_2, \ldots, D_k] \in R^{d \times n}$ is the dictionary we need to learn, where D_i is associated with class i. Ideally, the optimal representation matrix $Z \in R^{n \times m}$ should be block diagonal, i.e., the samples in different classes are not chosen for representing each other. By further assuming that the ideal within-class representation coefficients should be all ones, Zhang *et al.* [8] used an ideal representation matrix $Q = [q_1, q_2, \ldots, q_m] \in R^{n \times m}$ as a prior, where q_i corresponding to sample x_i is in a form of $[0, \ldots, 0, 1, \ldots, 1, 0, \ldots, 0]^T$. Namely, if x_i belongs to class j, the coefficients in q_i for D_j are all ones, while the others are all zeros. Then, the model for learning dictionary D can be formulated as

$$\min_{Z,D,E} \|Z\|_* + \beta\|Z\|_1 + \alpha\|Z - Q\|_F^2 + \lambda\|E\|_1$$
$$\text{s.t. } X = DZ + E \tag{5}$$

where λ, α, and β are all positive parameters. By solving the above problem (5), a dictionary D can be obtained. After the dictionary D is learned, the representations Z of all samples (training and testing samples) are computed by disregarding the term with Q in (5), i.e., solving the following model:

$$\min_{Z,E} \|Z\|_* + \beta\|Z\|_1 + \lambda\|E\|_1$$
$$\text{s.t. } X = DZ + E. \tag{6}$$

Zhang *et al.* [8] also proposed learning a dictionary by setting $\alpha = 0$, i.e., removing the ideal representation matrix Q from (5). We call this method the LRRC method, as it removes the structural information encoded in Q.

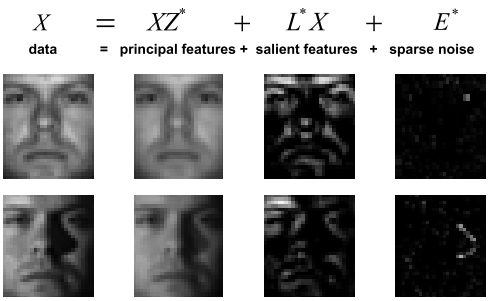

Fig. 1. Examples of decomposition of data matrix by LatLRR, adapted from [15].

Zhang *et al.* [8] reported good results of image classification by the SLRRC and LRRC methods. However, using Q as the ideal representation matrix is questionable, because it is unreasonable that the within-class coefficients are all ones. Moreover, the problem (5) is nonconvex. Its solution depends on initialization (it uses the K singular value decomposition (SVD) [26] method to initialize D). Finally, it is also difficult to tune the three parameters.

LatLRR [15] is a recently proposed feature learning method, which is also based on the LRR. To handle the case of insufficient samples, which often happens for high-dimensional data, LatLRR supposes that some unobserved samples should be involved in representing the observed samples. Its model can be formulated as

$$\min_{Z} \|Z\|_*, \quad \text{s.t. } X = [X, X_H]Z \tag{7}$$

where X is the observed sample and X_H is the unobserved hidden sample. With Bayesian inference,[1] X can be represented as $X = XZ + LX$, where $Z \in R^{m \times m}$ is the LRR of $X \in R^{d \times m}$ and $L \in R^{d \times d}$ is a low-rank projection matrix. So, the LatLRR can be formulated as

$$\min_{Z,L} \|Z\|_* + \|L\|_*, \quad \text{s.t. } X = XZ + LX. \tag{8}$$

From the experimental results (Fig. 1), they found that the features represented by XZ were visually similar to PCA features. For a certain image, its principal features can be roughly regarded as its projection onto a subspace that represents the image [15]. Hence, they called the features XZ principal features. They also noted that the features LX correspond to the key object parts (e.g., the eyes) and are usually discriminative for recognition. So, they call the features LX salient features. If the data matrix X is noisy, sparse noise E is considered

$$\min_{Z,L} \|Z\|_* + \|L\|_* + \lambda\|E\|_1$$
$$\text{s.t. } X = XZ + LX + E \tag{9}$$

where λ is a positive parameter.

However, recently, Zhang *et al.* [16] doubted the effectiveness of LatLRR. They proved that the solution to the noiseless LatLRR model (8) would be nonunique. Such nonuniqueness makes the recognition performance of LatLRR unpredictable.

All the aforementioned representation-based methods perform feature learning and classification in two separate steps.

[1]Please refer to [15] for the details of deduction.

Such mechanism may not be optimal for recognition tasks as feature learning, which mainly exploits sparsity and low rankness, is not closely related to the classification error.

To overcome this drawback, we present a novel feature learning method. We integrate the closed-form solutions to LatLRR with a ridge regression-based classifier, where the regulated classification error is minimized for choosing the optimal linear transform L. Therefore, the recognition accuracy can be significantly improved. In Section III, we provide a detailed description of our method.

III. Integrated Low-Rank-Based Discriminative Feature Learning

In this section, we first present how to integrate the closed-form solutions to LatLRR with the ridge regression and utilize the labels of data to learn discriminative features of clean data. More often than not, data are corrupted (e.g., noise) in real applications. Then, we extend our framework to handle corrupted data. The features extracted by our method can be used for recognition directly.

A. Closed-Form Solutions to Noiseless LatLRR

To begin with, we quote the following theorem in [16] on the complete closed-form solutions to the noiseless LatLRR model (8).

Theorem 1 [16]: The complete solutions to problem (8) are as follows:

$$Z^* = V_X(I-S)V_X^T \quad \text{and} \quad L^* = U_X S U_X^T \tag{10}$$

where $U_X \Sigma_X V_X^T$ is the skinny SVD of X and S is any block-diagonal matrix that satisfies two constraints: 1) its blocks are compatible with Σ_X, i.e., if $(\Sigma_X)_{ii} \neq (\Sigma_X)_{jj}$, then $S_{ij} = 0$ and 2) both S and $I - S$ are positive semidefinite. Note that S can usually be chosen as diagonal with diagonal entries being any number between 0 and 1.

Although the nonuniqueness of solutions undermines the validity of LatLRR, it also brings us a benefit. Namely, we can choose the most appropriate solution for subsequent classification among the solution set.

B. Model for Integrated Low-Rank-Based Discriminative Feature Learning

Our basic idea is to utilize the supervised information, e.g., the labels of training samples, to learn discriminative features LX resulting from the LatLRR model. During the training phase of classification, features of samples are fed into a classifier $f(x, W)$ to learn its model parameters W. We aim at optimizing for L by minimizing the classification error. In this way, our discriminative feature learning method is tightly coupled with classification. Our objective function for learning projection matrix L and parameters W of classifier can be defined as

$$\begin{aligned} &\min_{L,W} \sum_{i=1}^{m} \varphi(h_i,\, f(Lx_i, W)) + \alpha \|W\|_F^2 \\ &\text{s.t. } L = U_X S U_X^T \end{aligned} \tag{11}$$

where $x_i \in R^d$ is the ith sample in $X \in R^{d\times m}$, d is the dimension of feature vectors, and m is the number of samples. $U_X \in R^{d\times r}$ and $S \in R^{r\times r}$ satisfy the constraints in Theorem 1, where r is the rank of X. W is the parameters of classifier $f(x, W)$. φ is the classification loss function. h_i is the label vector of the ith sample. $\alpha > 0$ is a regularization parameter.

In this paper, we use a linear predictive classifier $f(x, W) = Wx$ and a quadratic loss function, i.e., adopt the multivariate ridge regression [27], where $W \in R^{c\times d}$ and c is the number of categories. For other classifiers, the optimization can still be performed but is more involved. We leave it as a future work. By our choice, the optimization problem (11) can be written as

$$\begin{aligned} &\min_{W,L} \|H - WLX\|_F^2 + \alpha \|W\|_F^2 \\ &\text{s.t. } L = U_X S U_X^T \end{aligned} \tag{12}$$

where $H = [h_1, h_2, \ldots, h_n] \in R^{c\times m}$ is the label matrix and $h_i = [0, 0, \ldots, 1, \ldots, 0, 0]^T \in R^c$ is the label of x_i. The term $\|H - WLX\|_F^2$ represents the classification error [19], [21]. By solving this optimization problem, an optimal projection matrix L and parameters W can be learned. Accordingly, discriminative features LX can be obtained.

C. Solving the Optimization Problem

To solve problem (12) more easily, we do some simplifications. First, we observe that the singular values of the data matrix X are usually distinct from each other, i.e., $(\Sigma_X)_{ii} \neq (\Sigma_X)_{jj}$ when $i \neq j$. So, the S in the solution (10) degenerates to a diagonal matrix, with all its diagonal entries ranging from 0 to 1. Second, since we only focus on learning the discriminative features, the constraint that $I - S$ is positive semidefinite is not necessary for our purpose. So, we only need to bound $S_{ii} \geq 0$, $\forall i = 1, \ldots, r$. Suppose that $U\Sigma V^T$ is the full SVD of X, then only the first r singular values are nonzeros. Assume that matrix $\Lambda \in R^{d\times d}$ is a square matrix and $\text{diag}(\Lambda) = (S_{11}, S_{22}, \ldots, S_{rr}, 0, 0, \ldots, 0) \in R^d$, then we have

$$L = U_X S U_X^T = U \Lambda U^T. \tag{13}$$

As $UU^T = I$, $U^TU = I$, and $VV^T = I$, we can deduce the following:

$$\begin{aligned} &\|H - WLX\|_F^2 + \alpha \|W\|_F^2 \\ &\quad = \|H - WU\Lambda U^T U\Sigma V^T\|_F^2 + \alpha \|W\|_F^2 \\ &\quad = \|HV - WU\Lambda\Sigma\|_F^2 + \alpha \|W\|_F^2 \\ &\quad = \|HV - WU\Lambda\Sigma\|_F^2 + \alpha \|WU\|_F^2. \end{aligned} \tag{14}$$

Let $\tilde{H} = HV$ and $\tilde{W} = WU$, then the objective function can be further written as

$$\begin{aligned} &\|H - WLX\|_F^2 + \alpha \|W\|_F^2 \\ &\quad = \|\tilde{H} - \tilde{W}\Lambda\Sigma\|_F^2 + \alpha \|\tilde{W}\|_F^2 \\ &\quad = \sum_{i=1}^{r} \|\tilde{H}_i - S_{ii}\sigma_i \tilde{W}_i\|_2^2 + \sum_{i=r+1}^{m} \|\tilde{H}_i\|_2^2 \\ &\qquad + \alpha \sum_{i=1}^{r} \|\tilde{W}_i\|_2^2 + \alpha \sum_{i=r+1}^{m} \|\tilde{W}_i\|_2^2. \end{aligned} \tag{15}$$

Therefore, we can see that for the optimal $\tilde{W}$, $\tilde{W}_i = 0$, and $i = r+1, \ldots, m$.

Now we focus on solving for S_{ii} and $\tilde{W}_i$, where $i = 1, \ldots, r$. The optimization problem reduces to

$$\min_{\substack{S_{11},\ldots,S_{rr}\\ \tilde{W}_1,\ldots,\tilde{W}_r}} \sum_{i=1}^{r} \left(\|\tilde{H}_i - S_{ii}\sigma_i \tilde{W}_i\|_2^2 + \alpha \|\tilde{W}_i\|_2^2\right)$$
$$\text{s.t. } S_{ii} \geq 0, \quad i = 1, 2, \ldots, r. \tag{16}$$

But the optimization problem (16) is not well defined, as the optimal $\tilde{W}_i$ should approach zero, while S_{ii} approaches infinity. To circumvent this situation, we add an additional constraint $\sum_{i=1}^{r} S_{ii}\sigma_i = t$, where t is a positive constant. We use this constraint because $S_{ii}\sigma_i$ is the coefficient of $\tilde{W}_i$, and hence, it can make the entries in $\tilde{W}$ more balanced. We do not use the constraint $\sum_{i=1}^{r} S_{ii} = t$, because the magnitudes of σ_i's can vary significantly, resulting in very unbalanced entries in $\tilde{W}$.

Let $g = [S_{11}\sigma_1, \ldots, S_{rr}\sigma_r]^T$ and $Q = [S_{11}\sigma_1 \tilde{W}_1, \ldots, S_{rr}\sigma_r \tilde{W}_r]$, problem (16) is reformulated as the following problem:

$$\min_{g,Q} \sum_{i=1}^{r} \left(\|\tilde{H}_i - Q_i\|_2^2 + \frac{\alpha}{g_i^2}\|Q_i\|_2^2 \right)$$
$$\text{s.t. } \sum_{i=1}^{r} g_i = t, \quad g_i \geq 0, \quad i = 1, 2, \ldots, r. \tag{17}$$

The optimization problem (17) is not jointly convex with respect to (Q, g). Therefore, we solve it by alternate minimization.

We first solve for Q. By fixing g, the updating of Q is rewritten as

$$\begin{aligned} Q_i &= \arg\min_{Q_i} \|\tilde{H}_i - Q_i\|_2^2 + \frac{\alpha}{g_i^2}\|Q_i\|_2^2 \\ &= \frac{g_i^2}{g_i^2 + \alpha}\tilde{H}_i, \quad i = 1, \ldots, r. \end{aligned} \tag{18}$$

However, when Q is fixed, the updating of g needs a little more effort

$$g = \arg\min_{\sum_{i=1}^{r} g_i = t, g_i \geq 0} \sum_{i=1}^{r} \frac{\alpha}{g_i^2}\|Q_i\|_2^2. \tag{19}$$

We use the method of Lagrange multiplier to solve for g. The Lagrangian function of (19) is

$$\mathcal{L}(g, \tau) = \sum_{i=1}^{r} \frac{\alpha\|Q_i\|_2^2}{g_i^2} + \tau\left(\sum_{i=1}^{r} g_i - t\right). \tag{20}$$

Then, we compute the derivative of $\mathcal{L}$ with respect to g

$$\frac{\partial \mathcal{L}}{\partial g_i} = -\frac{2\alpha\|Q_i\|_2^2}{g_i^3} + \tau. \tag{21}$$

By combining $\sum_{i=1}^{r} g_i = t$ and $\partial\mathcal{L}/\partial g_i = 0$, we can obtain the following solution:

$$g_i = \frac{t\|Q_i\|_2^{\frac{2}{3}}}{\sum_{i=1}^{r}\|Q_i\|_2^{\frac{2}{3}}}. \tag{22}$$

Algorithm 1 Integrated Learning of Discriminative Features from Clean Data

Input: The training data X_{tr}, the testing data X_{ts}, the label matrix H of X_{tr}. The parameters $\alpha > 0$ and $\varepsilon > 0$, and the constant $t > 0$.

Initialize: Conduct full SVD of X_{tr}:

$$X_{tr} = U\Sigma V^T$$

to obtain the rank r of X_{tr} and $\tilde{H} = HV$. Set $g_i^0 = \frac{t}{r}$, $Q^0 = 0$, and $k = 0$.

While $\|g^{k+1} - g^k\|_\infty > \varepsilon$ **or** $\|Q^{k+1} - Q^k\|_\infty > \varepsilon$ **do**

1. Fix g^k to update Q^{k+1}

$$Q_i^{k+1} = \frac{(g_i^k)^2}{(g_i^k)^2 + \alpha}\tilde{H}_i, \quad i = 1, \ldots, r.$$

2. Fix Q^{k+1} to update g^{k+1},

$$g_i^{k+1} = \frac{t\|Q_i^{k+1}\|_2^{\frac{2}{3}}}{\sum_{i=1}^{r}\|Q_i^{k+1}\|_2^{\frac{2}{3}}}, \quad i = 1, \ldots, r.$$

end while

4. Compute the projection matrix $L = U\Lambda U^T$, where $\Lambda_{ii} = g_i/\sigma_i$ $(i = 1, \ldots, r)$ and the values of other entries are all zeros.

5. Compute the extracted features $Z_{tr} = LX_{tr}$, $Z_{ts} = LX_{ts}$.

6. Compute the classifier parameters W. First, compute $\tilde{W}$, $\tilde{W}_i = Q_i/g_i$ $(i = 1, \ldots, r)$ and $\tilde{W}_i = 0$ $(i = r+1, \ldots, m)$. Then compute $W = \tilde{W}U^T$.

Output: Discriminative features Z_{tr}, Z_{ts}, and parameters W of the linear classifier.

The detailed optimization procedure is presented in Algorithm 1.

D. Convergence Analysis

In this section, we give the convergence analysis of our alternative minimization algorithm (i.e., Algorithm 1). We show that our algorithm decreases the objective function value monotonically, and any accumulation point of the sequence $\{(Q^k, g^k)\}$ generated by our algorithm is a Karush-Kuhn-Tucker (KKT) point of problem (17).

Theorem 2: Assume that $F(Q, g) = \sum_{i=1}^{r}(\|\tilde{H}_i - Q_i\|_2^2 + (\alpha/g_i^2)\|Q_i\|_2^2)$ is our objective function. The sequence $\{(Q^k, g^k)\}$ generated by Algorithm 1 satisfies the following properties.

1) $F(Q^k, g^k)$ is monotonically decreasing. Actually, it satisfies the following inequality:

$$F(Q^k, g^k) - F(Q^{k+1}, g^{k+1}) \geq \frac{\gamma - L}{2}\|Q^{k+1} - Q^k\|_F^2 \geq 0 \tag{23}$$

where $L = 2$ is the Lipschitz constant of function $f(x) = \|y - x\|_2^2$, in which x and y are two vectors. γ is a constant satisfying $L = 2 < \gamma \leq 4$.

2) $\lim_{k\to\infty} \|Q^{k+1} - Q^k\|_F^2 = 0$, $\lim_{k\to\infty} \|g^{k+1} - g^k\|_2^2 = 0$.

3) The sequences $\{Q^k\}$ and $\{g^k\}$ are both bounded.

Then, we can prove that any accumulation point of the sequence $\{(Q^k, g^k)\}$ generated by our algorithm is a KKT point of problem (17), as stated in Theorem 3.

Theorem 3: Assume that $\{(Q^k, g^k)\}$ is the sequence generated in Algorithm 1. Then, any accumulation point (Q^*, g^*) of $\{(Q^k, g^k)\}$ is a KKT point of problem (17).

The proofs of Theorems 2 and 3 can be found in the supplementary material.

E. Handling Corrupted Data

Now, we consider the situation when data X are corrupted. In this case, the LatLRR model is defined as problem (9). When the data are noisy, LatLRR (9) uses the contaminated data as the dictionary (the term XZ) and also extracts features from noisy X (the term LX). However, Wei and Lin [28] and Favaro *et al.* [29] pointed out that adopting the contaminated data as the dictionary is valid only when the percentage of corruption is relatively low and the noise level is also low. To overcome this limitation, Wei and Lin [28], Favaro *et al.* [29], and Zhang *et al.* [30] proposed to denoise X first, and then apply the noiseless LRR or the LatLRR to the denoised data. This leads to the following model [30]:

$$\begin{aligned} &\min_{Z,L,E} \|Z\|_* + \|L\|_* + \lambda\|E\|_1 \\ &\text{s.t. } X - E = (X - E)Z + L(X - E). \end{aligned} \tag{24}$$

Zhang *et al.* [30] proved that solving the above problem is equivalent to denoising X with the robust PCA [17] first to obtain (A, E), and then solving noiseless LatLRR (8) with X replaced by A. They proved the following theorem.

Theorem 4 [30]: Let the pair (A^*, E^*) be any optimal solution to the robust PCA problem. Then, the new noisy LatLRR model (24) has minimizers (Z^*, L^*, E^*), where

$$Z^* = V_A(I - S)V_A^T \quad \text{and} \quad L^* = U_A S U_A^T \tag{25}$$

in which U_A, Σ_A, V_A, and S satisfy the conditions in Theorem 1, with X replaced by A.

Accordingly, solving (24) simply reduces to solving the robust PCA problem, and thus the computation cost is greatly reduced. In our method, we first use the robust PCA to remove the noise of training data X_{tr} and obtain clean data A_{tr}. Then, we utilize A_{tr} to obtain the clean data $A_{ts} = U_{A_{tr}} U_{A_{tr}}^T X_{ts}$ of testing data matrix X_{ts}, where the columns of $U_{A_{tr}}$ are the left singular vectors of skinny SVD of A_{tr}, since clean training data A_{tr} and clean testing data A_{ts} should span the same subspace. Our feature learning method in the case of noise is described in Algorithm 2. Note that the rank r of training data is estimated by robust PCA and we can control it by tuning the parameter λ in the robust PCA.

F. Classification

When (12) is solved, we obtain both the extracted features $Z = [Z_{tr}, Z_{ts}]$ and a linear classifier's parameters W. We can directly use the obtained classifier parameters W for classification. In this way, we do not normalize the extracted features. Suppose $z \in Z_{ts}$ is the feature of a testing sample, its label is assigned as

$$j^* = \underset{j}{\operatorname{argmax}} \ (Wz)_j. \tag{26}$$

Algorithm 2 Integrated Learning of Discriminative Features From Corrupted Data

Input: The training data X_{tr}, the testing data X_{ts}, the label matrix H of X_{tr}. The parameters λ, α and $\varepsilon > 0$, the constant $t > 0$.

1. With the parameters λ and ε, conduct Robust PCA on the training data matrix X_{tr} and obtain the clean data A_{tr} and its skinny SVD $A_{tr} = U_{A_{tr}} \Sigma_{A_{tr}} V_{A_{tr}}^T$. Then we can obtain the clean data $A_{ts} = U_{A_{tr}} U_{A_{tr}}^T X_{ts}$ of testing data matrix X_{ts}.

2. With input A_{tr}, A_{ts}, H, α, ε and t, use Algorithm 1 to obtain discriminative feature Z_{tr} and Z_{ts} and parameters W of the linear classifier, where the full SVD of A_{tr} need not be recomputed because the skinny SVD of A_{tr} is output by Robust PCA [31] and we only need to augment $U_{A_{tr}}$ and $V_{A_{tr}}$ with their respective orthogonal complements.

Output: Discriminative features Z_{tr} and Z_{ts} and parameters W of the linear classifier.

This is our original integrated low-rank discriminative feature learning (ILRDFL) method. But empirically, we find that normalizing extracted features can lead to even better recognition results. A possible reason is that normalized features may have a more uniform statistical distribution. In this case, parameters W need to be updated accordingly, which does not take much time since ridge regression has a closed-form solution. So, we view these two extra low-computation steps as postprocessing. We call this method the normalized ILRDFL (NILRDFL) method. In Section IV, we will compare these two methods with state-of-the-art ones.

G. Parameter Settings

At a first glance, our algorithm has three parameters to tune, λ, α, and t. In reality, we only need to tune one parameter λ in the robust PCA problem (3). Indeed, the parameter α is a regularization parameter in the optimization problem (17) and our methods are insensitive to it, which can be seen in Fig. 5. In all of our experiments, we set $\alpha = 10^{-4}$. As for the parameter t, we simply set $t = d$. In the ILRDFL method, we directly use the learned classifier parameters W for recognition. In the NILRDFL method, after computing S, we normalize the extracted features and then retrain the ridge regression classifier. Therefore, only the parameter λ needs to be tuned. Fortunately, Candès *et al.* [17] have provided a suggested value $1/(\max(d, m))^{1/2}$, which provides good reference on the order of magnitude when we tune λ.

H. Fast Algorithm for Robust PCA [32]

The major computation of our algorithm lies in the step 1 of Algorithm 2. It requires performing the robust PCA, whose complexity is $O(\text{rdm})$ at each iteration, where r is the rank of data matrix and $d \times m$ is the size of data matrix [31]. When the size of data set is large, it is a very expensive computation task. Liu *et al.* [32] presented an algorithm called ℓ_1-filtering, which is a fast randomized algorithm for solving the robust PCA. Its complexity is $O(r^2(d + m))$ at each iteration [32], which is much lower than $O(\text{rmd})$ when d and m are large.

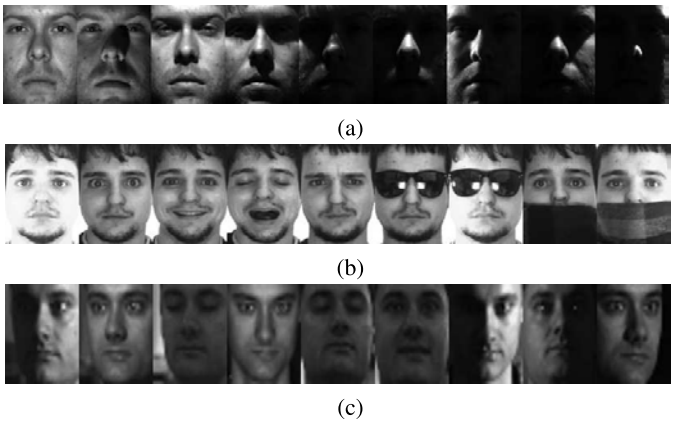

Fig. 2. Samples of face databases. (a) Samples of Extended YaleB. (b) Samples of the AR database. (c) Samples of the PIE database.

When $r/\min(d, m)$ is sufficiently small, with high probability, the ℓ_1-filtering produces the same solution as by solving the full-scale robust PCA directly [32]. However, if this condition is not satisfied, using ℓ_1-filtering for solving the robust PCA may cause a degraded recognition rate. The detail of ℓ_1-filtering can be found in the supplementary material.

IV. Experiments

In this section, we first evaluate our ILRDFL and NILRDFL on three widely used face databases: 1) Extended YaleB [33]; 2) AR [34]; and 3) PIE [35]. Note that the difficulties of these three face databases are not the same. As shown in Fig. 2, Extended YaleB is relatively simple. For each individual, it has ∼64 near frontal images under different illuminations. The challenge of AR is that it contains different facial expressions, illumination conditions, and occlusions (sun glasses and scarf). The PIE database is taken under different poses, expressions, and illuminations. Compared with the first two databases, it is more difficult to identify. We also test our methods on three more different types of databases: 1) Caltech 101 database [36], for object recognition; 2) Fifteen Scene Categories [2], for scene classification; and 3) UCF50 [37], for action recognition. Since the feature dimensions of Caltech 101, Fifteen Scene Categories, and UCF50 are too high, PCA is applied to reduce their dimensions to 1500, 3000, and 3000, respectively.

In all the above recognition tasks, we compare our two methods with SRC [5], CRC [12], the locality constrained linear coding (LLC) methods [6], LRC [7], LRSIC [7], LRRC [8], SLRRC [8], and TDDL [19]. To further demonstrate that our methods benefit more from a feature extraction, we also compare our methods with LatLRR [15] and robust PCA [17]. Since some methods, we compare with and our methods all use the multivariate ridge regression classifier, which usually achieves better performance than K-Nearest Neighbors. For fairness, LatLRR and robust PCA also use the multivariate ridge regression classifier. In each specific task, we further compare with other state-of-the-art methods for that task. The platform is MATLAB 2013a under Windows 8 on a PC equipped with a 3.4-GHz CPU and 16-GB memory.

TABLE II
Recognition Rates (%) on the Extended YaleB Database

#Training Samples per Person	10	15	20	25
Fisherfaces [1]	85.4	91.1	93.9	95.7
SRC [5]	87.9	93.6	**96.4**	98.0
CRC [12]	84.4	91.7	95.7	97.3
LLC [6]	77.8	87.0	90.9	93.9
LRC [7]	87.7	92.3	94.6	96.4
LRSIC [7]	88.2	94.0	95.1	96.7
LRRC [8]	84.8	91.6	93.6	96.3
SLRRC [8]	85.2	92.2	94.8	96.6
TDDL [19]	84.7	89.5	93.8	95.6
LatLRR [15]	83.3	88.8	92.3	94.6
Robust PCA [17]	86.2	91.3	94.1	95.9
ILRDFL (our method)	**88.3**±0.7	**94.2**±0.7	**96.4**±0.5	**98.7**±0.6
NILRDFL (our method)	**88.8**±0.7	**95.0**±0.6	**97.1**±0.4	**98.9**±0.4

A. Face Recognition

In the face recognition task, besides, for example, SRC, that we have mentioned above, we further compare with Fisherfaces [1].

1) Extended YaleB: Extended YaleB [33] consists of 2414 cropped frontal face images of 38 people. Every image has $192 \times 168 = 32\,256$ pixels. There are between 59 and 64 images for each person. In the experiments, we down-sample these images by 4 such that the down-sampled feature dimension d is 2016. We randomly select 10, 15, 20, and 25 training images from each person and the remaining images are used for testing. Every experiment runs 10 times.

As a common setting, Fisherfaces reduce the feature dimension to 37 [38]. When we evaluate SRC [5], CRC [12], LRC [7], and LRSIC [7], all training samples are used as the dictionary. The number of neighbors of LLC [6] is set to 5, which is the same as that in [6]. As [8] did, the dictionary size for LRRC [8], SLRRC [8], and TDDL [19] is all 140, i.e., the trained dictionary has five atoms for each person. We set $\lambda = 0.03$ in our methods. The experimental results are summarized in Table II. Note that we also report the standard deviations of our two methods next to our average recognition rates. We can see that with different numbers of training samples, our two methods always achieve the best recognition results. Furthermore, our NILRDFL method obtains better results than our ILRDFL method.

The discriminative feature faces of Extended YaleB extracted by ILRDFL are shown in Fig. 3. The ILRDFL method is different from dimensionality reduction methods. Actually, the ILRDFL method does not reduce the feature dimension. It only aims to find the most discriminative feature, while dimensionality reduction methods aim to reduce feature dimension and retain some discriminant information at the same time.

We then test the robustness of our methods. In this experiment, all images are resized to 48×42 pixels. We randomly select 25 training samples per person and the remaining ones are used for testing. As [5] and [8] did, a percentage of randomly chosen pixels in the training samples are replaced with independent identically distributed noise, which is uniformly distributed on $[0, y_{\max}]$, where $y_{\max}$ is the largest possible pixel value. The percentage of corrupted pixels varies from 10% to 90%. Fig. 4 shows the recognition

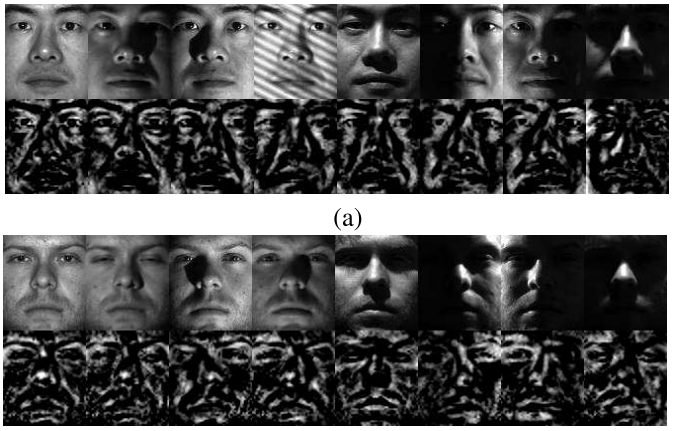

Fig. 3. Examples of discriminative features extracted from the Extended YaleB database. (a) and (b) First rows are original face images and the second rows are the corresponding discriminative features.

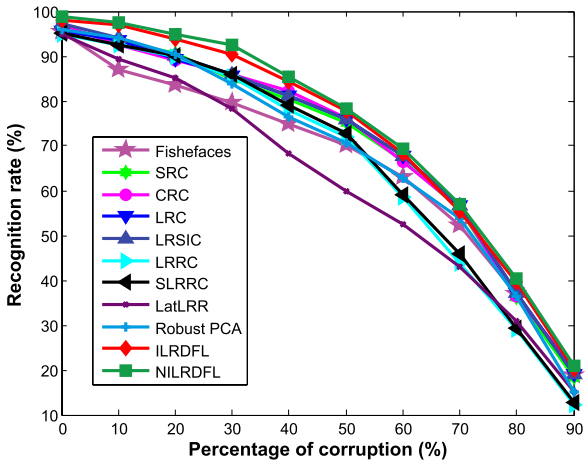

Fig. 4. Recognition rates on Extended YaleB under different percentages of random corruption.

rates of our methods and other nine competitors. Our methods outperform the others at all levels of corruption.

2) AR: The AR database [34] contains over 4000 color images corresponding to 126 people's faces (70 men and 56 women). Each person has 26 face images taken during two sessions. In each session, each person has 13 images, in which three images with sunglasses, another three with scarfs, and the remaining seven with different facial expressions and illumination conditions. All these images are of 165×120 pixels. Following the common experimental setting [7], [8], we select a subset of the database consisting of 2600 images from 50 male subjects and 50 female subjects. In this experiment, we also down-sample all images. When testing the LLC algorithm, the down-sample rate is two, while for other methods the down-sample rate is three. The reason we set different down-sample rates is that LLC encodes the Scale-Invariant Feature Transform (SIFT) features [39] and we should maintain a certain amount of SIFT features. The number of neighbors of LLC is set to five. Fisherfaces still reduce the feature dimension to 37. SRC, CRC, LRC, and LRSIC take all training samples as the dictionary. The trained dictionary for LRRC, SLRRC, and TDDL has 500 atoms. The

TABLE III
RECOGNITION RATES (%) ON THE AR DATABASE

Scenario	Sunglasses	Scarf	Mixed
Fisherfaces [1]	86.9	85.6	83.9
SRC [5]	88.6	85.6	83.2
CRC [12]	90.0	87.1	86.9
LLC [6]	87.1	85.8	84.1
LRC [7]	84.7	78.6	81.3
LRSIC [7]	87.2	79.5	83.5
LRRC [8]	86.1	83.4	82.7
SLRRC [8]	89.0	85.3	84.8
TDDL [19]	83.6	83.3	82.2
LatLRR [15]	87.1	86.3	84.3
Robust PCA [17]	88.1	86.6	86.5
ILRDFL (our method)	**90.9**±0.3	**91.8**±0.7	**91.2**±0.8
NILRDFL (our method)	**92.1**±0.2	**92.2**±0.6	**90.6**±0.7

parameter λ in our methods is set to 0.032. As [7] and [8] did, we consider the following three scenarios.

a) Sunglasses: In this scenario, the training samples contain seven neutral images and one image with the occlusion of sunglasses from session 1. Testing samples consist of seven neutral images from session 2 and five images with sunglasses, in which two are the remaining images with sunglasses and three from session 2.

b) Scarf: In this scenario, we only consider unobscured images and corrupted images due to the occlusion of scarf. We select seven unobscured images plus one image with scarf from session 1 for training. The remaining images with scarf (from sessions 1 and 2) and the unobscured images from session 2 are used for testing.

c) Sunglasses and scarf: In this scenario, we choose seven neutral images plus one with sunglasses and one with scarf from session 1 for training. All the remainder in sessions 1 and 2 are used for testing. Namely, we use nine images for training and the remaining seventeen images for testing.

We repeat these experiments three times and report the average recognition rates in Table III. The performances of our two methods are both better than all our compared methods. For the sunglasses, the scarf, and the mixed scenarios, the ILRDFL method achieves ∼0.9%, 4.7%, and 4.3% improvements, respectively, while NILRDFL makes ∼2.1%, 5.1%, and 3.7% improvements, respectively. Compared with other methods, our methods are very robust when there exists much noise in data, due to the effectiveness of robust PCA in removing corruptions.

3) PIE: The PIE database [35] contains 41 368 images of 68 people, each with 13 different poses, 43 different illumination conditions, and 4 different expressions. We select a subset of PIE for experiment, which contains five near frontal poses (C05, C07, C09, C27, C29) and all the images are taken under different illuminations and expressions. In our experiment, there are about 170 images for each person. In the LLC method [6], each image is normalized to a size of 64×64 pixels. In other methods, the size of each image is only 32×32 pixels. The down-sample rates are different because of the same reason as before. All the training samples are used as the dictionary for SRC [5], CRC [12], LRC [7], and LRSIC [7]. The size of learned dictionary for LRRC [8],

TABLE IV
RECOGNITION RATES (%) ON THE PIE DATABASE

#Training Samples per Person	10	15	20	25
Fisherfaces [1]	74.6	78.1	80.3	85.2
SRC [5]	77.3	87.2	90.5	93.3
CRC [12]	**83.3**	**88.1**	90.4	93.1
LLC [6]	77.1	85.5	89.9	93.0
LRC [7]	79.1	84.7	88.3	93.4
LRSIC [7]	82.4	87.7	90.6	93.5
LRRC [8]	79.8	85.2	89.1	91.3
SLRRC [8]	80.9	86.0	89.9	91.8
TDDL [19]	78.4	84.4	87.9	91.0
LatLRR [15]	79.4	85.8	89.6	91.6
Robust PCA [17]	80.3	84.1	87.8	90.7
ILRDFL (our method)	**83.3**±0.8	87.8±0.7	**90.7**±0.4	**93.6**±0.2
NILRDFL (our method)	**85.6**±0.8	**89.5**±0.6	**91.9**±0.3	**94.0**±0.2

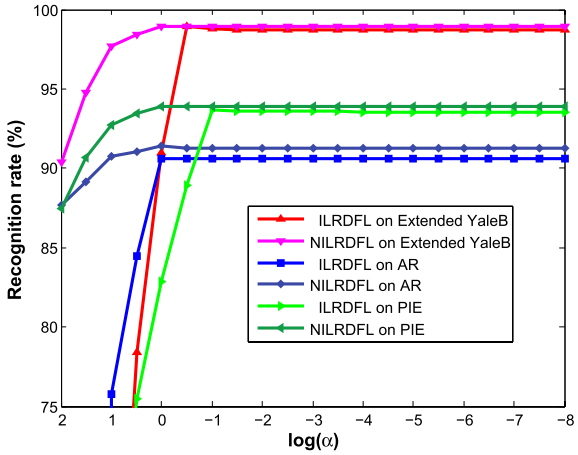

Fig. 5. Effects of parameter α on our two methods.

SLRRC [8], and TDDL [19] is 340. We set $\lambda = 0.06$ in our methods.

We select different numbers of training samples per person to test these methods. The recognition rates are summarized in Table IV. Our methods achieve good results and outperform the compared methods.

As stated earlier, our methods are insensitive to the regularization parameter α in the multivariate ridge regression (12). We verify this by testing the effect of the value of α on our algorithms on three data sets, Extended YaleB, AR, and PIE. As shown in Fig. 5, when the value of α ranges from 10^{-1} to 10^{-8}, all the recognition rates on Extended YaleB, AR, and PIE are stable. Our methods are robust to the choice of α.

B. Object Recognition

In our experiment, we use the Caltech 101 database [36] to evaluate our methods for object recognition. Caltech 101 is a widely used database for object recognition. It contains a total of 9146 images, split between 101 distinct objects (including faces, watches, ants, pianos, and so on) and a background category. So, there are 102 categories in total. Each object category contains about 31 to 800 images. The size of each image is roughly 300×200 pixels.

As [8] and [21] did, we also evaluate our methods using spatial pyramid features. The features can be computed as follows. First, we extract SIFT descriptors of 16×16 over a grid with a spacing of eight pixels. Second, we build three-level spatial pyramid features based on the extracted SIFT features with three kind of grids with size 1×1, 2×2, and 4×4, respectively. Then, we code the three-level spatial pyramid features with a codebook of size 1024. Since the feature dimension is too high, PCA is used to reduce the feature dimension to 1500. In the experiments, we randomly select 30 samples per category as training data and use the remaining samples for testing. As [21] did, LLC [6] is the original LLC, which uses sparse coding to encode SIFT descriptors [39], while LLC* uses sparse coding to encode the spatial pyramid features. For fairness, SRC [5], CRC [12], LRC [7], LRSIC [7], LRRC [8], SLRRC [8], LatLRR [15], robust PCA [17], and our two methods all use the spatial pyramid features. We evaluate SRC, CRC, LRC, LRSIC, LRRC, SLRRC, and TDDL [19] with a dictionary size 3060, i.e., for 30 dictionary items per category. We set both the neighborhood size of LLC and LLC* as 30. The parameter λ in our methods is 0.3.

TABLE V
RECOGNITION RATES (%) ON CALTECH 101 DATABASE

Method	Accuracy	Method	Accuracy
SRC [5]	69.3	Robust PCA [17]	68.4
CRC [12]	71.5	Lazebnik [2]	64.6
LLC [6]	64.8	Gemert [40]	64.1
LLC* [6]	70.8	Yang [41]	73.2
LRC [7]	69.9	Geusebroek [42]	64.1
LRSIC [7]	70.7	Y. Ng [43]	72.6
LRRC [8]	70.1	Malik [44]	56.6
SLRRC [8]	71.0	LC-KSVD1 [21]	73.4
TDDL [19]	68.1	LC-KSVD2 [21]	**73.6**
LatLRR [15]	63.5		
ILRDFL (ours)	72.6±0.9	NILRDFL (ours)	**75.2**±0.8

As Table V shows, our NILRDFL method performs the best among all the compared methods and has ∼1.6% improvement over the runner-up. ILRDFL also achieves a good recognition rate. It is worth noting that when we evaluate ILRDFL and NILRDFL, there are a total of twelve and seventeen classes that achieve 100% recognition rate, respectively.

C. Scene Classification

We test scene classification with the Fifteen Scene Categories database [2]. It is a database of 15 natural scene categories that expands on the 13-category database released in [45]. It contains 4485 images falling into 15 categories, such as bedrooms, kitchens, streets, and country scenes. Each category has 200 to 400 images.

The feature data of Fifteen Scene Categories are provided in [21], which can be downloaded from http://www.umiacs.umd.edu/~zhuolin/projectlcksvd.html. The features are computed as follows. First, computing a spatial pyramid feature with a four-level spatial pyramid and a SIFT-descriptor codebook with size of 200, and then PCA is applied to reduce the feature dimension to 3000. As [2] did, we randomly select 100 images per category as training data and use the remaining samples for testing. The detailed comparison results are shown in Table VI. SRC, CRC, LRC, LRSIC, SLR, LRRC, SLRRC, LatLRR, robust PCA, and our two methods all use the spatial pyramid feature

TABLE VI
RECOGNITION RATES (%) ON 15 SCENE CATEGORIES DATABASE

Method	Accuracy	Method	Accuracy
SRC [5]	91.8	LatLRR [15]	91.5
CRC [12]	92.3	Robust PCA [17]	95.3
LLC [6]	79.4	Lazebnik [2]	81.4
LLC* [6]	89.2	Gemert [40]	76.7
LRC [7]	91.9	Yang [41]	80.3
LRSIC [7]	92.4	Lian [46]	86.4
LRRC [8]	90.1	Boureau [47]	84.3
SLRRC [8]	91.3	LC-KSVD1 [21]	90.4
TDDL [19]	92.1	LC-KSVD2 [21]	92.9
ILRDFL (ours)	**97.6**±0.3	NILRDFL (ours)	**97.8**±0.3

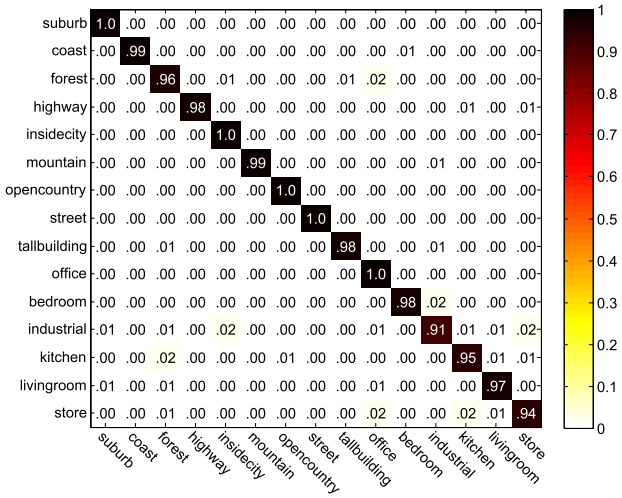

Fig. 6. Confusion matrix of ILRDFL on the Fifteen Scene Categories database. The average classification rates of each class are along the diagonal. The entry in the ith row and j column is the percentage of images from class i that are misidentified as class j.

provided in [21]. The dictionary sizes of SRC, CRC, LRC, LRSIC, LRRC, SLRRC, and TDDL are all 450. LLC and LLC* both have 30 neighborhoods. We set the parameter $\lambda = 0.5$ in our methods.

As Table VI shows, our methods perform the best among all the competitors. ILRDFL and NILRDFL make ∼2.3% and 2.5% improvement over the third best method, respectively. The confusion matrix of the ILRDFL method can be seen in Fig. 6, where the average recognition rates for each class are along the diagonal. There is no class that is classified badly and the worst recognition rate is as high as 91%.

D. Action Recognition

Finally, we test our methods and related algorithms with action recognition, using the UCF50 database [37]. The UCF50 database is one of the largest action recognition databases, consisting of realistic videos taken from Youtube. It contains 50 action categories with a total of 6617 action videos and the categories are *Baseball Pitch*, B*asketball Shooting*, *Biking*, *Diving*, *Tennis Swing*, and so on. Some images from this database are shown in Fig. 7.

For this database, we use the action feature representations presented in [48], whose code and feature data can be downloaded from http://www.cse.buffalo.edu/~jcorso/r/actionbank. As the dimension of action feature is very high, we use

Fig. 7. Ten categories in the UCF50 database.

TABLE VII
RECOGNITION RATES (%) ON UCF50 DATABASE

Method	Accuracy	Method	Accuracy
SRC [5]	61.4	TDDL [19]	57.1
CRC [12]	63.9	LatLRR [15]	66.4
LLC* [6]	57.0	Robust PCA [17]	69.7
LRC [7]	57.2	Gist [49]	38.8
LRSIC [7]	58.6	Laptev et al. [50], [51]	47.9
LRRC [8]	54.3	Action Bank [48]	57.9
SLRRC [8]	56.9		
ILRDFL (ours)	**73.6**±0.8	NILRDFL (ours)	**75.9**±0.6

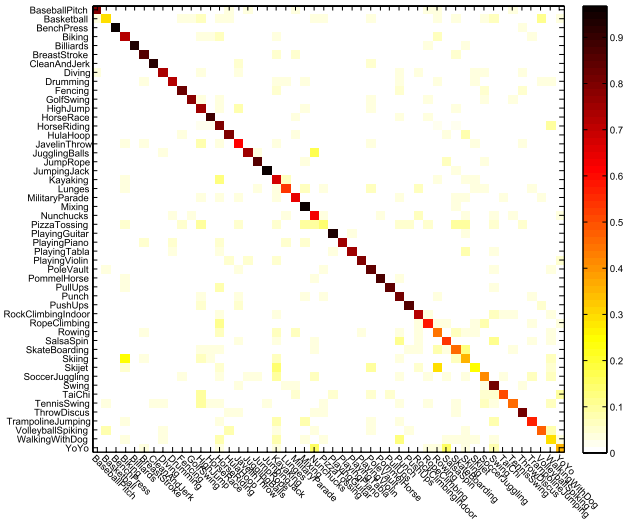

Fig. 8. Confusion matrix of ILRDFL on the UCF50 database. The classification rates are not shown. The color legend is drawn on the right, best viewed in color.

PCA to reduce the feature dimension to 3000. Then, we take dimension-reduced feature to evaluate SRC, CRC, LLC, LRC, LRSIC, LRRC, SLRRC, TDDL, LatLRR, robust PCA, and our methods. Following the common experiment settings [48]–[51], we test these methods with the fivefold groupwise cross-validation methodology. The dictionary sizes for SRC, CRC, LRC, LRSIC, TDDL are all 1500, i.e., 30 dictionary atoms for each category. When we evaluate LLC*, we use the original LLC method to encode the action feature and the neighborhood number is 30. We set $\lambda = 0.5$ in our methods.

Table VII presents the detailed comparison results. Note that our methods outperform the others. ILRDFL and NILRDFL make ∼3.9% and 6.2% improvement over the third best, respectively. The confusion matrix of our ILRDFL method, which is shown in Fig. 8, shows a dominant diagonal with no stand-out confusion among the classes. Only two categories

TABLE VIII
AVERAGE TESTING TIME (SECONDS) ON THE SIX DATABASES

Method	Extended YaleB	AR	PIE	Caltech 101	15 Scene Categories	UCF50
SRC [5]	0.4254	0.4381	0.2825	9.5604	0.1433	2.2606
LRC [7]	0.2847	0.4374	0.2632	4.9783	0.0373	1.6752
LRSIC [7]	0.2834	0.4245	0.2605	6.7942	0.0362	1.6840
LRRC [8]	0.0394	0.1828	0.2647	1.7433	0.0278	0.0666
SLRRC [8]	0.0398	0.1834	0.2568	1.9324	0.0280	0.0668
TDDL [19]	0.0253	0.1267	0.1954	1.5766	0.0208	0.0433
ILRDFL (our method)	1.9812e-04	6.6498e-04	1.2282e-04	2.3528e-04	4.7273e-04	0.0018
NILRDFL (our method)	2.2904e-04	8.1731e-04	1.3707e-04	2.9237e-04	4.9138e-04	0.0021

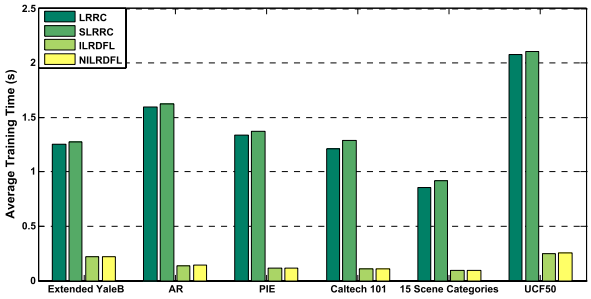

Fig. 9. Average training time (seconds) on the six databases. The training time of our methods contains the robust PCA denoising time.

(Pizza Tossing and Skijet) obtain relatively bad classification rates. Other categories are all classified well.

E. Comparison of Computation Time

In the above sections, we have compared our methods with other state-of-the-art methods in terms of the recognition rate. Now, we compare the average training and testing time of our methods with those of SRC [5], LRC [7], LRSIC [7], LRRC [8], SLRRC [8], and TDDL [19] on the six testing databases. Note that the testing phases of LatLRR and robust PCA are similar to ours (these two methods only need to do projection and use a linear classifier for classification), so we do not report the testing time of these two methods. The experimental settings in this section are as described in the above sections, respectively. The training time is defined as the time spent on training parameters of a model (it may contain denoising, such as the robust PCA denoising time, learning a dictionary, a projection matrix, and classifier parameters). The testing time is the time from inputting a test sample to outputting its label. The average training time and testing time are both computed as an average over all the training samples and the testing samples, respectively. Note that SRC has no training time and only has testing time, since it only needs to represent inputting testing samples as a linear combination of dictionary items, then use the representation coefficients for recognition. So, when evaluating the average training time, we select LRRC and SLRRC as our competitors. In Fig. 9, we compare our two methods with LRRC and SLRRC on the six benchmark databases. ILRDFL and NILRDFL are about six and ten times faster than LRRC and SLRRC on the Extended YaleB and the remaining five databases, respectively. We also note that NILRDFL is roughly as fast as ILRDFL, since both the time for normalization and retraining a ridge regression classifier are negligible.

The average testing time on each one of the six databases is reported in Table VIII. Both ILRDFL and NILRDFL are more than 20 times faster than the compared methods. This is because in the testing phase, SRC, LRC, LRSIC, LRRC, SLRRC, and TDDL all need to encode the testing samples with a dictionary, which requires a lot of time to solve an optimization problem. In contrast, our methods are very simple. They only need to do projection and use a linear classifier to conduct classification, which is very time efficient. It should also be noticed that all the state-of-the-art methods cost much more testing time on Caltech 101 than on the others. The reason is that when the dictionay size is large, it will be much more computationally expensive, and Caltech 101 has a much larger size of dictionary than other testing databases.

F. Speed Up With the Fast Algorithm

In the previous experiments, we just solve the full-sized robust PCA for our methods. Most of our training and testing processes are faster than other representation-based methods. In this section, we show the effectiveness of speeding up the training process of our methods by solving the robust PCA problem with the ℓ_1-filtering algorithm, called fast ILRDFL (F-ILRDFL), when handling large scale databases. We still use Extended YaleB [33], Caltech 101 [36], and UCF50 [37], but we do not reduce the feature dimension. The experimental settings are as follows.

1) Extended YaleB: The size of data matrix is $37\,600 \times 2414$. We randomly select 30 training images per person and use the remaining for testing. SRC, LRC, and LRSIC use all the training samples as the dictionary. The dictionary for LRRC, SLRRC, and TDDL has 30 dictionary atoms for every person. In the ℓ_1-filtering used in our methods, the size of seed matrix is 1140×1140 and we set $\lambda = 0.1$ when we apply robust PCA to recover the seed matrix.

2) Caltech 101: The size of feature matrix is $21\,506 \times 9144$. We randomly select 30 training samples each category and use the remaining for testing. All the training samples are used as the dictionary for SRC, LRC, and LRSIC. For LRRC, SLRRC, and TDDL, we also train a dictionary with 30 atoms for every class. We set the size of the seed matrix as 2500×2500 and $\lambda = 0.08$ in our methods.

3) UCF50: The size of feature matrix is $14\,965 \times 6617$. We use the fivefold groupwise cross-validation methodology to evaluate these methods. The dictionary sizes for SRC, LRC, LRSIC, LRRC, SLRRC, and TDDL are all 1500, i.e., 30 dictionary atoms for each category. We set the size of seed matrix as 2500×2500 and $\lambda = 0.06$.

TABLE IX
AVERAGE RECOGNITION RATES (%) AND AVERAGE TESTING TIME (SECONDS) ON THE THREE DATABASES, WHERE THE FEATURE DIMENSIONS ARE NOT REDUCED

Methods	Extended YaleB		Caltech 101		UCF50	
	Accuracy	Average Testing Time	Accuracy	Average Testing Time	Accuracy	Average Testing Time
SRC [5]	98.2	0.7971	68.6	12.7507	57.8	2.0312
LRC [7]	97.9	0.2966	69.3	7.3734	62.6	0.6665
LRSIC [7]	98.7	0.2931	71.2	8.9252	65.7	0.8570
LRRC [8]	98.1	2.8756	69.5	2.9491	64.9	2.5449
SLRRC [8]	**98.8**	2.9634	72.6	2.8915	67.8	2.5799
TDDL [19]	96.9	2.3218	67.3	2.3591	60.1	2.0021
F-ILRDFL (our method)	98.7±0.9	0.0024	**74.4**±1.1	0.0030	**70.5**±1.0	0.0035
F-NILRDFL (our method)	**99.3**±0.8	0.0031	**75.3**±0.9	0.0032	**72.1**±1.0	0.0036

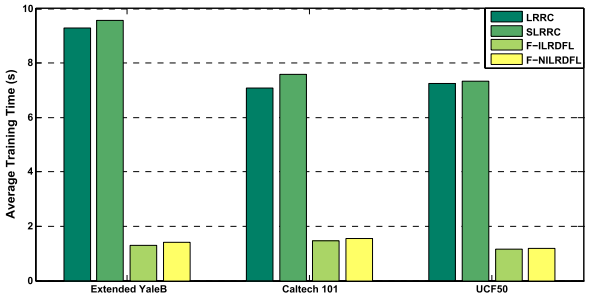

Fig. 10. Average training time (seconds) for fast algorithm on the three test databases. The training time of our methods contains the robust PCA denoising time.

We first compare the average training time of our two methods with LRRC and SLRRC. The results are shown in Fig. 10. Our F-ILRDFL and F-NILRDFL are both several times faster than LRRC and SLRRC. When the size of data is large, learning dictionary and representations of training samples are computationally expensive, since LRRC and SLRRC have to solve a nonconvex problem with a lot of parameters, which converges slowly. We use the ℓ_1-filtering algorithm to speed up the training process of our methods and the effect is evident.

The average testing time and the recognition accuracy are summarized in Table IX. Both F-ILRDFL and F-NILRDFL are several hundred times faster than the compared methods. In the testing phase, all our compared methods have to solve an optimization problem to obtain the representations of testing samples under a learned dictionary. When the scale of data matrix is large, the speed of these methods drop dramatically. However, the testing process of our methods are much simpler, since we have no optimization problem to solve and we only project the testing samples and classify them with a linear classifier. We also note that though the data matrix is large, its rank is low. With high probability, the ℓ_1-filtering in our methods produces the same solution as by solving full-scale robust PCA directly. From Table IX, our F-ILRDFL and F-NILRDFL both achieve higher recognition rates than the compared methods. In conclusion, our fast methods not only run fast, but also achieve better performance.

V. CONCLUSION

We propose a novel supervised a low-rank-based discriminative feature learning method. Unlike other representation-based feature learning methods that separate a feature learning process and subsequent classification into two steps and optimize sparsity or low rankness to extract features, our method learns discriminative features by integrating LatLRR with ridge regression to minimize the classification error directly. This way, the extracted features are directly related to the recognition rate. We also adopt the ℓ_1-filtering algorithm to speed up the computation of robust PCA, which we use for denoising data robustly. Finally, our method has only one parameter that needs tuning, which makes the performance tuning very easy.

Extensive experimental results demonstrate that our method obtains better classification results than other representation-based methods and state-of-the-art recognition methods, even with a simple linear classifier. Our method is also much more robust than other methods in comparison. On large scale data sets, by adopting the ℓ_1-filtering algorithm our method is also much faster than other methods in comparison.

In the future, in the same spirit, we will try integrating other feature learning methods with more sophisticated classification errors.

ACKNOWLEDGMENT

The authors would like to thank Hongyang Zhang for helping draft introduction and for valuable discussions.

REFERENCES

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

[2] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. 19th IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, New York, NY, USA, Jun. 2006, pp. 2169–2178.

[3] Y. Huang, K. Huang, Y. Yu, and T. Tan, "Salient coding for image classification," in *Proc. 24th IEEE Conf. Comput. Vis. Pattern Recognit.*, Colorado Springs, CO, USA, Jun. 2011, pp. 1753–1760.

[4] L. Liu, L. Wang, and X. Liu, "In defense of soft-assignment coding," in *Proc. 13th IEEE Int. Conf. Comput. Vis.*, Barcelona, Spain, Nov. 2011, pp. 2486–2493.

[5] J. Wright, A. Y. Yang, A. Ganesh, S. 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.

[6] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *Proc. 23rd IEEE Conf. Comput. Vis. Pattern Recognit.*, San Francisco, CA, USA, Jun. 2010, pp. 3360–3367.

[7] C.-F. Chen, C.-P. Wei, and Y.-C. F. Wang, "Low-rank matrix recovery with structural incoherence for robust face recognition," in *Proc. 25th IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, Jun. 2012, pp. 2618–2625.

[8] Y. Zhang, Z. Jiang, and L. S. Davis, "Learning structured low-rank representations for image classification," in *Proc. 26th IEEE Conf. Comput. Vis. Pattern Recognit.*, Portland, OR, USA, Jun. 2013, pp. 676–683.
[9] G. Liu, Z. Lin, and Y. Yu, "Robust subspace segmentation by low-rank representation," in *Proc. 27th Int. Conf. Mach. Learn.*, Haifa, Israel, Jun. 2010, pp. 663–670.
[10] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 171–184, Jan. 2013.
[11] Y. Wang, Y. Jiang, Y. Wu, and Z.-H. Zhou, "Spectral clustering on multiple manifolds," *IEEE Trans. Neural Netw.*, vol. 22, no. 7, pp. 1149–1161, Jul. 2011.
[12] L. Zhang, M. Yang, X. Feng, Y. Ma, and D. Zhang. (2012). "Collaborative representation based classification for face recognition." [Online]. Available: http://arxiv.org/abs/1204.2358
[13] Y. Deng, Q. Dai, R. Liu, Z. Zhang, and S. Hu, "Low-rank structure learning via nonconvex heuristic recovery," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 3, pp. 383–396, Mar. 2013.
[14] K. Tang, R. Liu, Z. Su, and J. Zhang, "Structure-constrained low-rank representation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 12, pp. 2167–2179, Dec. 2014.
[15] G. Liu and S. Yan, "Latent low-rank representation for subspace segmentation and feature extraction," in *Proc. 13th IEEE Int. Conf. Comput. Vis.*, Barcelona, Spain, Nov. 2011, pp. 1615–1622.
[16] H. Zhang, Z. Lin, and C. Zhang, "A counterexample for the validity of using nuclear norm as a convex surrogate of rank," in *Proc. 23rd Eur. Conf. Mach. Learn.*, Prague, Czech Republic, Sep. 2013, pp. 226–241.
[17] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *J. ACM*, vol. 58, no. 3, 2011, Art. ID 11.
[18] A. Argyriou, T. Evgeniou, and M. Pontil, "Multi-task feature learning," in *Proc. 21st Annu. Conf. Adv. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, Dec. 2007, pp. 41–48.
[19] J. Mairal, F. Bach, and J. Ponce, "Task-driven dictionary learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 4, pp. 791–804, Apr. 2012.
[20] Y. Amit, M. Fink, N. Srebro, and S. Ullman, "Uncovering shared structures in multiclass classification," in *Proc. 24th Int. Conf. Mach. Learn.*, Corvallis, OR, USA, Jun. 2007, pp. 17–24.
[21] Z. Jiang, Z. Lin, and L. S. Davis, "Label consistent K-SVD: Learning a discriminative dictionary for recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2651–2664, Nov. 2013.
[22] A. Wagner, J. Wright, A. Ganesh, Z. Zhou, and Y. Ma, "Towards a practical face recognition system: Robust registration and illumination by sparse representation," in *Proc. 22nd IEEE Conf. Comput. Vis. Pattern Recognit.*, Miami, FL, USA, Jun. 2009, pp. 597–604.
[23] M. Yang and L. Zhang, "Gabor feature based sparse representation for face recognition with Gabor occlusion dictionary," in *Proc. 11th Eur. Conf. Comput. Vis.*, Barcelona, Spain, Sep. 2010, pp. 448–461.
[24] M. Yang, L. Zhang, J. Yang, and D. Zhang, "Robust sparse coding for face recognition," in *Proc. 24th IEEE Conf. Comput. Vis. Pattern Recognit.*, Colorado Springs, CO, USA, Jun. 2011, pp. 625–632.
[25] R. Jenatton, J.-Y. Audibert, and F. Bach, "Structured variable selection with sparsity-inducing norms," *J. Mach. Learn. Res.*, vol. 12, pp. 2777–2824, Feb. 2011.
[26] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4311–4322, Nov. 2006.
[27] G. H. Golub, P. C. Hansen, and D. P. O'Leary, "Tikhonov regularization and total least squares," *SIAM J. Matrix Anal. Appl.*, vol. 21, no. 1, pp. 185–194, 1999.
[28] S. Wei and Z. Lin. (2011). "Analysis and improvement of low rank representation for subspace segmentation." [Online]. Available: http://arxiv.org/abs/1107.1561
[29] P. Favaro, R. Vidal, and A. Ravichandran, "A closed form solution to robust subspace estimation and clustering," in *Proc. 24th IEEE Conf. Comput. Vis. Pattern Recognit.*, Colorado Springs, CO, USA, Jun. 2011, pp. 1801–1807.
[30] H. Zhang, Z. Lin, C. Zhang, and J. Gao, "Relations among some low rank subspace recovery models," *Neural Comput.*, May 2015.
[31] Z. Lin, R. Liu, and Z. Su, "Linearized alternating direction method with adaptive penalty for low-rank representation," in *Proc. 25th Annu. Conf. Adv. Neural Inf. Process. Syst.*, Granada, Spain, Dec. 2011, pp. 612–620.
[32] R. Liu, Z. Lin, Z. Su, and J. Gao, "Linear time principal component pursuit and its extensions using ℓ_1 filtering," *Neurocomputing*, vol. 142, pp. 529–541, Oct. 2014.
[33] A. S. Georghiades, P. N. Belhumeur, and D. Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 6, pp. 643–660, Jun. 2001.
[34] A. Martínez and R. Benavente, "The AR face database," Dept. Centre de Visió per Computador, Univ. Autònoma Barcelona, Barcelona, Spain, CVC Tech. Rep. #24, Jun. 1998.
[35] 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.
[36] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories," in *Proc. 17th IEEE Conf. Comput. Vis. Pattern Recognit.*, Washington, DC, USA, Jun. 2004, pp. 59–70.
[37] K. K. Reddy and M. Shah, "Recognizing 50 human action categories of Web videos," *Mach. Vis. Appl.*, vol. 24, no. 5, pp. 971–981, 2013.
[38] D. Cai, X. He, Y. Hu, J. Han, and T. Huang, "Learning a spatially smooth subspace for face recognition," in *Proc. 20th IEEE Conf. Comput. Vis. Pattern Recognit.*, Minneapolis, MN, USA, Jun. 2007, pp. 1–7.
[39] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.
[40] J. C. van Gemert, J.-M. Geusebroek, C. J. Veenman, and A. W. M. Smeulders, "Kernel codebooks for scene categorization," in *Proc. 10th Eur. Conf. Comput. Vis.*, Marseille, France, Oct. 2008, pp. 696–709.
[41] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *Proc. 22nd IEEE Conf. Comput. Vis. Pattern Recognit.*, Miami, FL, USA, Jun. 2009, pp. 1794–1801.
[42] J. C. van Gemert, C. J. Veenman, A. W. M. Smeulders, and J.-M. Geusebroek, "Visual word ambiguity," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 7, pp. 1271–1283, Jul. 2010.
[43] A. Coates and A. Y. Ng, "The importance of encoding versus training with sparse coding and vector quantization," in *Proc. 28th Int. Conf. Mach. Learn.*, Washington, DC, USA, Jun. 2011, pp. 921–928.
[44] S. Maji, A. C. Berg, and J. Malik, "Efficient classification for additive kernel SVMs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 66–77, Jan. 2013.
[45] L. Fei-Fei and P. Perona, "A Bayesian hierarchical model for learning natural scene categories," in *Proc. 18th IEEE Conf. Comput. Vis. Pattern Recognit.*, San Diego, CA, USA, Jun. 2005, pp. 524–531.
[46] X.-C. Lian, Z. Li, B.-L. Lu, and L. Zhang, "Max-margin dictionary learning for multiclass image categorization," in *Proc. 11th Eur. Conf. Comput. Vis.*, Barcelona, Spain, Sep. 2010, pp. 157–170.
[47] Y.-L. Boureau, F. Bach, Y. LeCun, and J. Ponce, "Learning mid-level features for recognition," in *Proc. 23rd IEEE Conf. Comput. Vis. Pattern Recognit.*, San Francisco, CA, USA, Jun. 2010, pp. 2559–2566.
[48] S. Sadanand and J. J. Corso, "Action bank: A high-level representation of activity in video," in *Proc. 25th IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, Jun. 2012, pp. 1234–1241.
[49] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *Int. J. Comput. Vis.*, vol. 42, no. 3, pp. 145–175, 2001.
[50] I. Laptev, "On space-time interest points," *Int. J. Comput. Vis.*, vol. 64, nos. 2–3, pp. 107–123, 2005.
[51] H. Wang, M. M. Ullah, A. Kläser, I. Laptev, and C. Schmid, "Evaluation of local spatio-temporal features for action recognition," in *Proc. 20th Brit. Mach. Vis. Conf.*, London, U.K., Sep. 2009, pp. 124.1–124.11.

Pan Zhou received the bachelor's degree in computer science and technology from the China University of Geosciences, Wuhan, China, in 2013. He is currently pursuing the master's degree with the Key Laboratory of Machine Perception, School of Electronics Engineering and Computer Science, Peking University, Beijing, China.

His current research interests include computer vision, machine learning, and pattern recognition.

Zhouchen Lin (M'00–SM'08) received the Ph.D. degree in applied mathematics from Peking University, Beijing, China, in 2000.

He was a Guest Professor with Shanghai Jiao Tong University, Shanghai, China, Beijing Jiaotong University, Beijing, and Southeast University, Nanjing, China. He was also a Guest Researcher with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing. He is currently a Professor with the Key Laboratory of Machine Perception, School of Electronics Engineering and Computer Science, Peking University. He is a Chair Professor with Northeast Normal University, Changchun, China. His current research interests include computer vision, image processing, machine learning, pattern recognition, and numerical optimization.

Prof. Lin is an Area Chair of the IEEE Conference on Computer Vision and Pattern Recognition in 2014, the International Conference on Computer Vision in 2015, the Conference on Neural Information Processing Systems in 2015, and the Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence in 2016. He is also an Associate Editor of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE and the *International Journal of Computer Vision*.

Chao Zhang (M'06) received the Ph.D. degree in electrical engineering from Beijing Jiaotong University, Beijing, China, in 1995.

He was a Post-Doctoral Research Fellow with the National Laboratory on Machine Perception, Peking University, Beijing, from 1995 to 1997. He has been an Associate Professor with the Key Laboratory of Machine Perception, School of Electronics Engineering and Computer Science, Peking University, since 1997. His current research interests include image processing, statistical pattern recognition, and visual recognition.