Integrated Low Rank Based Discriminative Feature Learning for Recognition

Pan Zhou, Zhouchen Lin, Senior, 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 these methods achieve great success in many applications. However, these methods perform feature learning and the subsequent classification in two separate steps, which may not be optimal for the recognition tasks. In this paper, we present a supervised low rank based approach for learning discriminative features. Our approach combines feature learning with classification, by integrating Latent Low-Rank Representation (LatLRR) with a ridge regression based classifier, so that the regulated classification error is minimized. Our approach benefits from a recent discovery on the closed-form solutions of noiseless LatLRR. When there is noise, a Robust PCA based denoising step can be added. Extensive experimental results demonstrate the effectiveness and robustness of our method.

Index Terms—Feature Learning, Low-Rank Representation, Recognition.

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], [2], [3], [4], [5], [6], [7], [8], [9] focusing on learning discriminative features. For example, in [1], [2], [3] the authors view distances between samples as features and in [5] the authors project each descriptor into a local-coordinate system as a feature.

Recently, representation based feature learning methods [4], [6], [7], [10], [11], [12], [13] have drawn a lot of attention. The first representation based method may be Sparse Representation Classification (SRC) [4]. SRC finds the fewest number of training samples to represent a test sample and adopts the representation coefficients as a feature of the test sample. It is reported that SRC achieves surprisingly high accuracy in face recognition even under occlusion [4]. 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 [6], [7], [10], [11], [12], [13] have been proposed. These methods assume that samples in the same class should locate 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, thus enhancing the correlation among the representation coefficient vectors. As a result, these methods achieve great success in a lot of recognition problems, such as face and object recognition.

However, all existing representation based methods consist of two separate steps: extracting discriminative features by learning from training data and then inputting the features into a specific classifier for classification. Such a separation may limit the overall recognition performance. To remedy such an issue, in this paper we propose a simultaneous feature learning and data classification method, by integrating Latent Low-Rank Representation (LatLRR) with a ridge regression based classifier. LatLRR is a recently proposed unsupervised clustering and feature learning method [14]. We choose LatLRR because when there is no noise it has non-unique closed-form solutions [15], which is remarkable among all representation based methods.

The contributions of this paper are as follows:

• We propose a simple yet effective model for simultaneous feature learning and data classification. By integrating the closed-form solutions of LatLRR with a ridge regression based classifier, our model achieves an entire optimality in recognition in some sense.

• While the existing representation based methods minimize the sparsity or rankness of some solution related to feature learning, which is not directly connected to the subsequent recognition problem, our model directly minimizes the regularized classification error. As a consequence, our method achieves higher accuracy in recognition.

• Due to closed-form solutions of LatLRR, our feature learning approach is fast. When there is noise, we propose to denoise the data with Robust PCA first. We also incorporate a fast randomized algorithm for solving Robust PCA when the rank is relatively low as compared with the matrix size. As a consequence, our method also excels in speed when the scale of problem is large.

Extensive experimental results testify to the advantages of our method.

The remainder of the paper is organized as follows. ....

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 $X = [X_1, X_2, \ldots, X_k] \in \mathbb{R}^{d \times m}$, where $X_i \in \mathbb{R}^{d \times m_i}$ is the data sub-matrix of class $i$. We also denote the dictionary as $D = [D_1, D_2, \ldots, D_k] \in \mathbb{R}^{d \times k}$, where $D_i \in \mathbb{R}^{d \times k_i}$ is the sub-dictionary associated with the $i$th class.
**A. Sparse Representation Based Feature Learning**

SRC [4] may be the first representation based method. The main idea of SRC is to represent the input sample \( y \in \mathbb{R}^d \) as a linear combination of a few atoms in an overcomplete dictionary \( D \). The corresponding sparse representation \( \alpha \in \mathbb{R}^k \) can be computed by the following \( \ell_1 \) minimization problem:

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

Suppose that \( \alpha = [\alpha_1^T, \alpha_2^T, \cdots, \alpha_k^T]^T \) and \( \alpha_i \) is the sub-vector associated with the dictionary \( D_i \) of the \( i \)th class. A test sample \( y \) is classified as class \( j^* \) if class \( j^* \) results in the least reconstruct 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 reconstruction purpose, which is not always satisfied. Several methods are proposed to resolve this issue. Wagner et al. [17] proposed an extended SRC to handle variations of faces in illumination, alignment, pose, and occlusion. Yang et al. [18, 19, 11] also extended SRC to deal with outliers such as occlusions in face images. In their methods, collaborative representation based classification (CRC) [11] achieves a much higher face recognition rate. However, when all the data (both training and test images) are corrupted, these methods do not work well. Furthermore, sparse coding methods represent each test sample independently. This mechanism does not take advantage of any structural information from the data set. Actually, the data from the same class may share common (correlated) features.

**B. Low Rank Representation Based Feature Learning**

Before we introduce low rank representation based feature learning methods, we first introduce Robust PCA [16], 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 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 [16]:

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

where the nuclear norm and the \( \ell_1 \) norm are respectively the convex surrogates of rank function and the \( \ell_0 \) pseudonorm, i.e., the number of nonzero entries. \( \lambda \) is a positive parameter trading off 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.

Low-rank matrix recovery with structural incoherence based classification (LRSIC) method [6] is a feature learning method based on Robust PCA [16]. It uses Robust PCA to decompose the training data matrix \( X = [X_1, X_2, \cdots, X_k] \) into a low rank matrix \( A = [A_1, A_2, \cdots, A_k] \) and a sparse error matrix \( E = [E_1, E_2, \cdots, E_k] \), where \( X_i \) is the training data matrix for class \( i \) and \( A_i \) and \( E_i \) both correspond to \( X_i \). To remove the noise in data and reduce the feature dimension, Chen et al. [6] apply PCA to \( A \) to obtain a projection matrix \( W \) and then project both training data and testing data with \( W \). Finally, they use SRC [4] 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 LRSIC (LRC with structure incoherence) can be written as follows:

\[
\min_{A_i,E_i} \sum_{i=1}^{k} \left( \| A_i \|_* + \| E_i \|_1 \right) + \eta \sum_{j \neq i} \| A_j^T A_i \|_F^2, \tag{4}
\]

\[\text{s.t.} \quad X_i = A_i + E_i, \quad i = 1, 2, \cdots, k, \]

where \( \eta \) is a positive parameter. Then as LRC does, PCA and SRC is used for classification. However, Zhang et al. [7] points out that these two methods cannot preserve structure information well. Moreover, these two methods need to remove noise from training samples class by class. This process is computationally expensive when the number of classes is large.

Structured low-rank representation for classification (SLRRC) [7] 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 image classification. More specifically, suppose that \( D = [D_1, D_2, \cdots, D_k] \) is the dictionary we need to learn, where \( D_i \) is associated with class \( i \). Ideally, the optimal representation matrix \( Z \) 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. [7] use an ideal representation matrix \( Q = [q_1, q_2, \cdots, q_n] \) as a prior, where \( q_i \) corresponding to sample \( x_i \) is in a form of \([0, \cdots, 0, 1, \cdots, 0] \). Namely, if \( x_i \) belongs to class \( j \), the coefficients in \( q_i \) for \( D_j \) is all 1s, while others are all 0s. Then the model for learning dictionary \( D \) can be formulated as

\[
\min_{Z,D,E} \| Z \|_* + \beta \| Z \|_1 + \alpha \| Z - Q \|_F^2 + \lambda \| E \|_1, \tag{5}
\]

\[\text{s.t.} \quad X = DZ + E, \]

where \( \lambda, \alpha, \) and \( \beta \) are all positive parameters. By solving the above problem [5], a dictionary \( D \) can be obtained. After the dictionary \( D \) is learnt, the representations \( Z \) of all samples

**TABLE I**

<table>
<thead>
<tr>
<th>Notation</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>capital letter</td>
<td>A matrix. Especially, ( I ) is the identity matrix.</td>
</tr>
<tr>
<td>( M_{ij} )</td>
<td>The ((i, j))-th entry of matrix ( M ).</td>
</tr>
<tr>
<td>( M_i )</td>
<td>The (i)-th column of matrix ( M ).</td>
</tr>
<tr>
<td>( x_i )</td>
<td>The (i)-th entry of vector ( x ).</td>
</tr>
<tr>
<td>( M^T )</td>
<td>Transpose of matrix ( M ).</td>
</tr>
<tr>
<td>|\cdot|_1</td>
<td>Nuclear norm, the sum of all the singular values.</td>
</tr>
<tr>
<td>|\cdot|_p</td>
<td>Frobenious norm, ( |M|<em>p = \sqrt{\sum</em>{i,j} M_{ij}^p} ).</td>
</tr>
<tr>
<td>|\cdot|_2</td>
<td>Vector Euclidean norm, ( |x|_2 = \sqrt{\sum_i x_i^2} ).</td>
</tr>
<tr>
<td>diag(\cdot)</td>
<td>The diagonal entries of a matrix.</td>
</tr>
</tbody>
</table>
(training and testing samples) are computed by disregarding the term with \( Q \) in (5), i.e., solving the following model:

\[
\min_{Z,E} \| Z \|_* + \| Z \|_1 + \lambda \| E \|_1,
\]

s.t. \( X = DZ + E. \) (6)

Zhang et al. \[7\] also proposed learn dictionary by setting \( \alpha = 0 \), i.e., removing the ideal representation matrix \( Q \) from (5). We called this method the LRRC method as it removed the structural information encoded in \( Q \).

Zhang et al. \[7\] reported good image classification results of the SLRRC and LRRC methods. However, using \( Q \) as the ideal representation matrix is questionable, because that the within-class coefficients are all ones is not reasonable. Moreover, the problem (5) is nonconvex. So its solution depends on initialization (it uses the KSVD \[20\] method to initialize \( D \)). Finally, it is also difficult to tune the three parameters.

Latent Low-Rank Representation (LatLRR) \[14\] is a recently proposed feature learning method, which is also based on low rank representation. 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. With Bayesian inference, the noiseless LatLRR model can be formulated as

\[
\min_{Z,L} \| Z \|_* + \| L \|_*, \quad \text{s.t.} \quad X = XZ + LX, \quad \text{(7)}
\]

where \( Z \) is the low-rank representation of \( X \) and \( L \) is a low-rank projection matrix. In their model, the term \( XZ \) is called “principal features” and the term \( LX \) is the “salient features” of \( X \) (Fig. 1), which can be used for image classification. If the data matrix \( X \) is noisy, sparse noise \( E \) is taken into account:

\[
\min_{Z,L} \| Z \|_* + \| L \|_* + \lambda \| E \|_1,
\]

s.t. \( X = XZ + LX + E, \quad \text{(8)} \)

where \( \lambda \) is a positive parameter.

However, recently Zhang et al. \[15\] doubted the effectiveness of LatLRR. They proved that the solution of noiseless LatLRR model \[7\] is non-unique. The non-uniqueness makes the recognition performance of LatLRR unpredictable.

All the aforementioned representation based methods perform feature learning and classification in two separate steps. Such a mechanism may not be optimal for recognition tasks as feature learning, which mainly exploits sparsity and low-rankness, is not closely related to classification error.

To overcome such a drawback, we present a novel feature learning method. We integrate the closed-form solutions of LatLRR with a ridge regression based classifier, where the regulated classification error is minimized for choosing the optimal linear transform \( L \). So the recognition accuracy can be significantly improved. In the next section, 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 of LatLRR with ridge regression and utilize the labels of data to learn discriminative feature on clean data. More often than not, data is corrupted (e.g., noise) in real applications. Then we extend our framework to handle corrupted data, whose validity is supported by theory. The features extracted by our method can be used for recognition directly.

#### A. Closed-Form Solutions of Noiseless LatLRR

To begin with, we quote the following theorem by Zhang et al. \[15\] on the complete closed-form solutions to the noiseless LatLRR model \[7\].

**Theorem 3.1 (\[15\]):** The complete solutions to problem (7) are as follows

\[
Z^* = V_X(I - S)V_X^T \quad \text{and} \quad L^* = U_XS U_X^T, \quad \text{(9)}
\]

where \( U_XS \) is the skinny SVD of \( X \) and \( S \) is any block diagonal matrix that satisfies two constraints: 1. its blocks are compatible with \( \Sigma_X \), i.e., if \((\Sigma_X)_{ii} \neq (\Sigma_X)_{jj}\), then \( s_{ij} = 0 \); 2. both \( S \) and \( I - S \) are positive semi-definite.

Although the non-uniqueness of solutions underlines the validity of LatLRR, it also brings us a benefit. Namely, we can choose among the solution set which is the most appropriate for the subsequent classification.

#### 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 minimizing for \( L \) by minimizing the classification error. So 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

\[
\min_{L,W} \sum_{i=1}^{m} \varphi(h_i, f(Lx_i, W)) + \alpha \| W \|_F^2, \quad \text{(10)}
\]

s.t. \( L = U_XS U_X^T \),

where \( x_i \in \mathbb{R}^d \) is the \( i \)-th sample of \( X \in \mathbb{R}^{d \times m} \), \( d \) is the dimension of feature vectors, and \( m \) is the number of samples. \( U_X \in \mathbb{R}^{d \times r} \) and \( S \in \mathbb{R}^{r \times r} \) satisfy the constraint of Theorem 3.1.
where $r$ is the rank of $X$. $W \in \mathbb{R}^{c \times N}$ is the parameters of classifier $f(x,W)$, where $c$ is the number of categories. $\varphi$ is the classification loss function, $h_i = [0, 0, \ldots, 1, \ldots, 0]^T \in \mathbb{R}^c$ is the label vector of the $i$th sample, where the position of 1 indicates the class of $x_i$, and $\alpha$ 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 [21]. For other classifiers, the optimization can still be performed but is more involved. So we leave it a future work. By our choice, the optimization problem (10) can be obtained.

C. Solving the Optimization Problem

To solve problem (11) more easily, we do some simplifications. First, we observe that usually the singular values of the data matrix $X$ are distinct from each other, i.e., $(\Sigma_X)_{ii} \neq (\Sigma_X)_{jj}$ when $i \neq j$. So the $S$ in the solution (13) 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 semi-definite 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 nonzero. Let

$$\begin{pmatrix} S_{11} & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & S_{22} & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & S_{rr} & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \end{pmatrix} \in \mathbb{R}^{d \times d},$$

then we have

$$L = U_X S U_X^T = U D U^T.$$

As $U U^T = I$, $U^T U = I$ and $V V^T = I$, we can deduce the following:

$$\begin{align*}
\| H - W L X \|^2_F + \alpha \| W \|^2_F &= \| H - W U D U^T X \Sigma V^T \|^2_F + \alpha \| W \|^2_F \\
&= \| H V - W U D \Sigma \|^2_F + \alpha \| W \|^2_F \\
&= \| H V - W U D \|^2_F + \alpha \| W U \|^2_F.
\end{align*}$$

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

$$\begin{align*}
\| H - W L X \|^2_F + \alpha \| W \|^2_F &= \| \tilde{H} - \tilde{W} D \Sigma \|^2_F + \alpha \| \tilde{W} \|^2_F \\
&= \sum_{i=1}^r \| \tilde{H}_i - S_{ii} \sigma_i \tilde{W}_i \|^2_2 + \sum_{i=r+1}^m \| \tilde{H}_i \|^2_2 \\
&\quad + \alpha \sum_{i=1}^r \| \tilde{W}_i \|^2_2 + \alpha \sum_{i=r+1}^m \| \tilde{W}_i \|^2_2.
\end{align*}$$

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

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

$$\begin{align*}
\min_{S_{ii}, \tilde{W}_i} \sum_{i=1}^r \| \tilde{H}_i - S_{ii} \sigma_i \tilde{W}_i \|^2_2 + \alpha \| \tilde{W}_i \|^2_2 \\
\text{s.t.} \quad S_{ii} \geq 0, \quad i = 1, 2, \ldots, r.
\end{align*}$$

But optimization problem (15) 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. Let $g = [S_{11}\sigma_1, \ldots, S_{rr}\sigma_r]^T$ and $Q = [S_{11}\sigma_1 \tilde{W}_1, \ldots, S_{rr}\sigma_r \tilde{W}_r]^T$, problem (15) is reformulated as the following problem

$$\begin{align*}
\min_{g, Q} \sum_{i=1}^r \| \tilde{H}_i - g_i \|^2_2 + \alpha \frac{g_i^2}{\sigma_i} \| Q_i \|^2_2 \\
\text{s.t.} \quad \sum_{i=1}^r g_i = C, \quad g_i \geq 0, \quad i = 1, 2, \ldots, r.
\end{align*}$$

The optimization problem (16) is not jointly convex with respect to $(Q, g)$. Therefore, we solve it by alternative minimization.

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

$$Q_i = \arg \min_{Q_i} \| \tilde{H}_i - Q_i \|^2_2 + \alpha \frac{g_i^2}{\sigma_i} \| Q_i \|^2_2$$

$$g_i = \frac{g_i^2}{g_i^2 + \alpha} \tilde{H}_i, \quad i = 1, 2, \ldots, r.$$ (17)

However, when $Q$ is fixed the update of $g$ needs a little more effort:

$$g = \arg \min_{g_i = 1, g_i \geq 0} \sum_{i=1}^r \frac{\alpha}{g_i^2} \| Q_i \|^2_2.$$ (18)

We use the method of Lagrange multiplier to solve for $g$. The Lagrangian function of (18) is

$$L(g, \beta) = \sum_{i=1}^r \frac{\alpha}{g_i^2} \| Q_i \|^2_2 + \beta \left( \sum_{i=1}^r g_i - t \right).$$ (19)

Then, we compute the derivative of $L$ with respect to $g$:

$$\frac{\partial L}{\partial g_i} = -\frac{2\alpha}{g_i^2} \| Q_i \|^2_2 + \beta.$$ (20)
Combining \( \sum_{i=1}^{r} g_i = t \) and \( \frac{g_i c}{g_i} = 0 \), we can obtain the following solution:

\[
g_i = \frac{t \| Q_i \|_2^2}{\sum_{i=1}^{r} \| Q_i \|_2^2}.
\]  

(21)

The detailed optimization procedure is presented in Algorithm 1.

**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 parameter \( \alpha > 0 \), \( \varepsilon > 0 \), and the constant \( t > 0 \).

**Initialize:** Conduct the singular value decomposition (SVD) of \( X_{tr} \):

\[
X_{tr} = U \Sigma V^T
\]

and obtain the rank \( r \) of \( X_{tr} \), \( \tilde{H} = HV \), and \( \tilde{W} = WU \). Set \( g^0_i = \frac{t}{r} \), \( Q^0 = 0 \), 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 \( g^{k+1} \):

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

end while

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

5. Compute the extracted features \( Z_{tr} = LX_{tr}, Z_{ts} = LX_{ts} \), and normalize all column vectors of \( Z_{tr} \) and \( Z_{ts} \).

**Output:** Discriminative features \( Z_{tr} \) and \( Z_{ts} \).

**D. Corrupted Data**

Now we consider the situation that data \( X \) is corrupted. In this case, the LatLRR model is defined as problem (8). When the data is noisy, LatLRR (8) uses the contaminated data as the dictionary (the term \( XZ \)) and also extracts features from noisy \( X \) (the term \( LX \)). However, Wei et al. (22) 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 limited. To overcome such a limitation, authors of (22), (23), (24) all propose to denoise \( X \) first and then apply noiseless LRR or LatLRR to the denoised data. This leads to the following model (24):

\[
\begin{align*}
\min_{Z,L,E} & \quad \| Z \|_* + \| L \|_* + \lambda \| E \|_1, \\
\text{s.t.} & \quad X - E = (X - E)Z + L(X - E).
\end{align*}
\]

(22)

Zhang et al. (24) proved that solving the above problem is equivalent to denoising \( X \) with Robust PCA (16) first to obtain \( (A, E) \) and then solving noiseless LatLRR (7) with \( X \) replaced by \( A \). They proved the following theorem.

**Theorem 3.2 (24):** Let the pair \( (A^*, E^*) \) be any optimal solution to the Robust PCA problem. Then the new noisy LatLRR model (22) has minimizers \( (Z^*, L^*, E^*) \), where

\[
Z^* = V_A(I - S)V_A^T, \quad L^* = U_A S U_A^T,
\]

in which \( U_A, \Sigma_A, V_A, \) and \( S \) satisfy the conditions in Theorem 3.1 with \( X \) replaced by \( A \).

So solving (22) simply reduces to solving Robust PCA, and thus the computation cost is greatly reduced. Our feature learning method in the case of noise is described in Algorithm 2.

**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 parameter \( \lambda, \alpha, \varepsilon \), and the constant \( t > 0 \).

1. With the parameters \( \lambda, \alpha, \varepsilon \), conduct Robust PCA on the data matrix \( X = [X_{tr}, X_{ts}] \) and obtain the clean data \( A^* = [A^*_t, A^*_s] \), where \( A^*_t \) and \( A^*_s \) correspond to \( X_{tr} \) and \( X_{ts} \), respectively.

2. With input \( A^*_t, A^*_s, H, \alpha, \varepsilon \) and \( t \), use Algorithm 1 to obtain discriminative feature \( Z_{tr} \) and \( Z_{ts} \), where the SVD of \( A^*_t \) need not be recomputed as it can be output by Robust PCA (25).

**Output:** Discriminative features \( Z_{tr} \) and \( Z_{ts} \).

**E. Classification**

When (11) is solved, we obtained both the optimal matrices \( L \) and \( W \) for computing the discriminative features and constructing a linear classifier, respectively. Given a test sample \( y \), we extract its feature as \( z = Ly \) and its label is assigned as:

???

**F. Parameter Settings**

At a first glance, our algorithm has three parameters to tune, \( \lambda, \alpha, \) and \( t \). In reality, we only need to tune one parameter \( \lambda \) in the Robust PCA problem (3)). Indeed, the parameter \( \alpha \) is a regularization parameter in the optimization problem (16) and our method is insensitive to it, which can be seen in Fig. 6. In all of our experiments, we set \( \alpha = 10^{-4} \). As for the parameter \( t \), we simply set \( t = d \). After computing \( S \), we normalize the extracted features (see Algorithm 1). So only the parameter \( \lambda \) needs to be tuned. Fortunately, Candès et al. (16) has provided a suggested value \( 1/\sqrt{\max(d,m)} \), which provides good reference on the order of magnitude when we tune \( \lambda \).
IV. Fast Algorithm for Robust PCA

The major computation of our algorithm is the step 1 in Algorithm 2. It requires performing Robust PCA, whose complexity is \(O(rd\text{dim})\) at each iteration \([25]\), where \(r\) is the rank of data matrix and \(d \times m\) is the size of data matrix. When the size of dataset is large, it is a very expensive computation task. In \([26]\), Liu et al. present an algorithm called \(\ell_1\)-filtering, which is a fast randomized algorithm for solving Robust PCA. We sketch it below.

\(\ell_1\)-filtering first randomly samples an \(s_r \times s_c\) submatrix \(X^s\) from \(X\), where \(s_r > 1\) and \(s_c > 1\) are the row and column oversampling rates, respectively. For simplicity, we assume that \(X^s\) is at the top left corner of matrix \(X\). Then accordingly \(X, A,\) and \(E\) is partitioned into:

\[
X = \begin{bmatrix} X^s & X^c \\ X^r & X^s \end{bmatrix}, \quad A = \begin{bmatrix} A^s & A^c \\ A^r & A^s \end{bmatrix}, \quad E = \begin{bmatrix} E^s & E^c \\ E^r & \hat{E}^s \end{bmatrix},
\]

(24)

where \(A\) is the low rank matrix we need to recover and \(E\) is the sparse error matrix.

Then \(\ell_1\)-filtering recovers \(A^s\), called the seed matrix, from \(X^s\) by solving a small-sized Robust PCA problem, e.g., via principal component pursuit \([3]\). Since \(s_r\) and \(s_c\) are both small as compared with \(m\) and \(n\), the computation of recovering \(A^s\) is much cheaper than recovering the whole \(A\).

Next, as the rank of \(A\) and \(A^s\) are both \(r\), there must exist matrix \(Q\) and \(P\) satisfying the following equations \([25]\):

\[
A^c = A^sQ, \quad A^r = P^TA^s.
\]

(25)

Since the matrix \(E\) is sparse, the matrices \(E^c\) and \(E^r\) are also sparse. So we can find matrix \(Q\) and \(P\) by minimizing the following problems:

\[
\min_{E^c, Q} \|E^c\|_1, \quad \text{s.t. } X^c = A^sQ + E^c,
\]

(26)

and

\[
\min_{E^r, P} \|E^r\|_1, \quad \text{s.t. } X^r = P^TA^s + E^r,
\]

(27)

respectively. For these two problems, using the alternating direction method (ADM) \([27]\) to solve them is efficient. So we can get \(P\) and \(Q\), thus \(A^c\) and \(A^r\) can be also obtained. Finally, only \(A^s\) needs to be computed. By the low-rankness of \(A\), we can obtain

\[
\hat{A}^s = P^TA^sQ.
\]

(28)

In summary, the matrix \(A\) can be recovered with a complexity of \(O(r^2(d + m))\) \([26]\), much lower than \(O(rd\text{dim})\) when \(d\) and \(m\) are large. When \(r/\min(d, m)\) is sufficiently small, with high probability the \(\ell_1\)-filtering produces the same solution as by solving the full-scale Robust PCA directly \([26]\). However, if this condition is not satisfied, using \(\ell_1\)-filtering for solving Robust PCA may cause a degraded recognition rate.

V. Experiments

In this section, we evaluate our Integrated Low Rank Discriminative Feature Learning (ILRDFL) method on three widely used face databases: Extended YaleB \([28]\), AR \([29]\), and PIE \([30]\). It should be pointed out that the difficulty of these three face databases are not the same. As shown in Fig. 2, Extended YaleB is relatively simple. For each individual, it has around 64 near frontal images under different illuminations. The challenge of AR database is that it contains different facial expressions, illumination conditions, and occlusions (sun glasses and scarf). The PIE database is taken under different pose, expression, and illumination. Compared with the first two databases, it is more difficult to identify.

We also test our method on two more different types of databases: Fifteen Scene Categories \([1]\), for scene classification, and UCF50 \([31]\), for action recognition. In our experiments, since the feature dimensions of Fifteen Scene Categories and UCF50 are too high, PCA is applied to reduce their dimensions to 3,000.

In all the above recognition tasks, we compare our method with SRC \([4]\), CRC \([11]\), the locality-constrained linear coding method (LLC) \([5]\), LRC \([6]\), LRSIC \([6]\), LRRC \([7]\), and SLRRC \([7]\). In each specific task, we further compare with other state-of-the-art methods for that task. Our code will be published online if our paper is accepted.

A. Face Recognition

In the face recognition task, besides SRC etc. that we have mentioned above we further compare with Fisherfaces \([32]\).

1) Extended YaleB: The Extended YaleB \([28]\) consists of 2,414 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 8 such that the down-sampled feature dimension \(d\) is 504. We randomly select 5, 10, 15, and 20 training images from each person and the others for testing. Every experiment runs 10 times.

As a common setting, Fisherfaces reduces the feature dimension to 37 \([?\?]\). When we evaluate SRC \([4]\), CRC \([11]\), LRC \([6]\), and LRSIC \([6]\), all training samples are used as the dictionary. The number of neighbors of LLC \([5]\) is set to 5, which is the same as in \([5]\). As \([7]\) did, the dictionary size for LRRC \([7]\) and SLRRC \([7]\) is 140, i.e., the trained dictionary has 5 atoms for each person. About the parameters of our method, we only need to consider one parameter \(\lambda\) and we set
The average computing time (seconds) for classifying a test image on Extended YaleB database

<table>
<thead>
<tr>
<th>Method</th>
<th>Average Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRC [4]</td>
<td>0.0915</td>
</tr>
<tr>
<td>LRC [6]</td>
<td>0.1074</td>
</tr>
<tr>
<td>LRSIC [6]</td>
<td>0.1132</td>
</tr>
<tr>
<td>LRRC [7]</td>
<td>0.0936</td>
</tr>
<tr>
<td>SLRRC [7]</td>
<td>0.1023</td>
</tr>
<tr>
<td>ILRDFL (our method)</td>
<td>0.1052</td>
</tr>
</tbody>
</table>

Robust PCA, the discriminative feature learning process (i.e., the Algorithm 1) does not take much time.

we then test the robustness of our method. In this experiment, all images are resized to 48 × 42 pixels. We randomly select 20 training samples per person and the remaining ones are used for testing. As [4] and [7] did, a percentage of randomly chosen pixels in training samples and testing samples are replaced with i.i.d. noise uniformly distributed on $[0, y_{\text{max}}]$, where $y_{\text{max}}$ is the largest possible pixel value. The percentage of corrupted pixels varies from 10% to 90%, Fig. 4 shows the recognition rates of our method and other seven competitors. Our method dramatically outperforms others at all levels of corruption. Up to 40% corruption, our method still performs almost perfectly, correctly recognizing over 97% of test images. Other methods are more sensitive to noise. Their recognition rates decrease rapidly when the percentage of corruption increases. At 50% corruption, our recognition rate is 91.3%, while none of others achieve higher than 80%. Even at 60% corruption, our recognition rate is still 80%. Compared with others methods, our method is much more robust to noise.

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

### Table II

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>78.0</td>
<td>84.2</td>
<td>83.5</td>
<td>77.8</td>
<td>86.1</td>
<td>86.6</td>
<td>80.3</td>
<td>81.3</td>
<td>92.1</td>
</tr>
<tr>
<td>15</td>
<td>84.3</td>
<td>88.8</td>
<td>89.6</td>
<td>87.0</td>
<td>88.9</td>
<td>91.5</td>
<td>84.8</td>
<td>86.7</td>
<td>95.6</td>
</tr>
<tr>
<td>20</td>
<td>87.0</td>
<td>93.1</td>
<td>93.0</td>
<td>89.9</td>
<td>93.1</td>
<td>94.1</td>
<td>88.9</td>
<td>89.5</td>
<td>98.1</td>
</tr>
</tbody>
</table>

Robust PCA, the discriminative feature learning process (i.e., the Algorithm 1) does not take much time.

We also compare the computation time of our method with those of SRC [4], LLC [5], LRC [6], LRSIC [6], LRRC [7], and SLRRC [7]. The time is computed by the following method. First we compute the total computation time, for our method, it contains the Robust PCA time, extracting feature time, training time and testing time, for SLRRC, it contains dictionary learning time, structured low-rank representation learning time, training time, and testing time. Second, we use the number of test sample divided by the total time to compute average computation time for a test image. (???) The average computing time is shown in Table III. Our method is roughly as fast as other methods. We also note that the major computation time of our method is the running time of

Fig. 4. Recognition rates under different percentages of random corruption on Extended YaleB.
we also down-sample all images. When testing the LLC algorithm, the down-sample rate is 2, while for other methods the down-sample rate is 3. The reason why we set different down-sample rates is that LLC encodes the SIFT features and we should maintain a certain amount of SIFT features. The number of neighbors of LLC is set to 5. 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 and SLRRC has 500 atoms. The parameter $\lambda$ of our method is set as 0.012. As [6] and [7] did, we consider the following three scenarios.

**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 the session 2.

**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 session 1 and session 2) and the unobscured images at session 2 are for testing.

**Sunglasses and Scarf:** This scenario takes all the images with sunglasses and scarf into account. We choose seven neutral images plus one with sunglasses and one with scarf at session 1 for training. All the remainder at session 1 and session 2 are used for testing. Namely, we use nine images for training and the remaining seventeen images for testing.

We repeat these three experiments 5 times and the average recognition rates are reported in Table IV. The performance of our method is better than Fisherfaces, SRC, CRC, LLC, LRC, LRSIC, LRRC, and SLRRC. Our method achieves about 5.6%, 8.3%, and 8.3% improvements for the sunglasses, the scarf, and the mixed scenarios, respectively. Compared with other methods, our method is very robust when there exist much noise in the data, thanks to the effectiveness of Robust PCA in removing corruptions.

We also present the extracted discriminative features on AR, as shown in Fig. 5. Our method can remove the principle feature and noise, only remain the discriminative feature, which has strong discriminative ability for recognition. Such as the two people in the Fig. 5, the two people’s feature face are easier to be identified. (???)

We also show the average computing time in Table V. Our method is slower than other methods. This is because the data are corrupted seriously. So Robust PCA has taken much time to remove the noise.

As stated earlier, our method is not sensitive to the regularization parameter $\alpha$ in the multivariate ridge regression (11). We verify this by testing the effect of the value of $\alpha$ on our algorithm on two datasets, Extended YaleB and AR. As can be seen in Fig. 6, when the value of $\alpha$ ranges from $10^{-2}$ to $10^{-8}$, both the recognition rates on Extended YaleB and AR datasets are stable. So our method is robust to the choice of $\alpha$.

![Fig. 5. Examples of discriminative features extracted from the AR database.](image)

![Fig. 6. The effects of parameter $\alpha$ on our method.](image)
TABLE VI
The recognition rates (%) on the PIE database

<table>
<thead>
<tr>
<th>#Training Samples per Person</th>
<th>10</th>
<th>15</th>
<th>20</th>
<th>25</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fisherfaces [32]</td>
<td>74.6</td>
<td>78.1</td>
<td>80.3</td>
<td>85.2</td>
</tr>
<tr>
<td>SRC [4]</td>
<td>77.3</td>
<td>87.2</td>
<td>90.9</td>
<td>93.3</td>
</tr>
<tr>
<td>CRC [11]</td>
<td>83.4</td>
<td>89.1</td>
<td>92.2</td>
<td>93.1</td>
</tr>
<tr>
<td>LLC [5]</td>
<td>77.1</td>
<td>85.5</td>
<td>89.9</td>
<td>93.0</td>
</tr>
<tr>
<td>LRC [6]</td>
<td>79.1</td>
<td>84.7</td>
<td>88.3</td>
<td>93.4</td>
</tr>
<tr>
<td>LRSIC [6]</td>
<td>82.4</td>
<td>87.7</td>
<td>90.9</td>
<td>93.5</td>
</tr>
<tr>
<td>LRRC [7]</td>
<td>79.8</td>
<td>85.2</td>
<td>89.1</td>
<td>91.3</td>
</tr>
<tr>
<td>SLRRC [7]</td>
<td>80.9</td>
<td>86.0</td>
<td>89.9</td>
<td>91.8</td>
</tr>
<tr>
<td>ILRDFL (our method)</td>
<td><strong>86.0</strong></td>
<td><strong>91.4</strong></td>
<td><strong>93.8</strong></td>
<td><strong>94.7</strong></td>
</tr>
</tbody>
</table>

TABLE VII
The average computing time (second) for classifying a test image in the PIE database

<table>
<thead>
<tr>
<th>Method</th>
<th>Average Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRC [4]</td>
<td>0.3804</td>
</tr>
<tr>
<td>LRC [6]</td>
<td>0.3978</td>
</tr>
<tr>
<td>LRSIC [6]</td>
<td>0.4000</td>
</tr>
<tr>
<td>LRRC [7]</td>
<td>0.2932</td>
</tr>
<tr>
<td>SLRRC [7]</td>
<td>0.3121</td>
</tr>
<tr>
<td>ILRDFL (our method)</td>
<td><strong>0.2053</strong></td>
</tr>
</tbody>
</table>

3) **PIE**: The PIE database [30] contains 41,368 images of 68 people, each being under 13 different poses, 43 different illumination conditions, and with 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 under different illuminations and expressions. So in our experiment, there are 170 images for each person. We evaluate our method with other state-of-the-art methods. When we evaluate the LLC method [5], each image is normalized to a size of 64 × 64 pixels. When we test 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. Namely, LLC encodes SIFT features and we need to provide a certain amount of SIFT features for LLC. All the training samples are used as the dictionary for SRC [4], CRC [11], LLC [5], LRC [6], LRSIC [6], LRRC [7], and SLRRC [7]. The size of learned dictionary for LRRC [7] and SLRRC [7] is 340. \( \lambda = 0.0075 \) is used in our method. We select different numbers of training samples per person to test these methods. The recognition rates are summarized in Table VII. Our method achieves good results and outperform the compared methods. When the training number of each person is 10, 15, 20, and 25, our method makes about 2.6%, 2.3%, 1.6%, and 1.2% improvement, respectively.

The average computing time for classifying one test image is also presented in Table VII. Our method is the fastest method among all compared method. It is approximately twice times faster than SRC, LRC, and LRSIC.

**B. Scene Classification**

We test scene classification with the Fifteen Scene Categories database [1]. It is a database of 15 natural scene categories that expands on the thirteen category database released by Li and Perona [34]. It contains 4,485 images falling into fifteen categories and each category has 200 to 400 images. The 15 categories vary from bedroom and kitchen to street and country scenes, as shown in Fig. 7.

The feature data of Fifteen Scene Categories is provided by [35], which can be downloaded from [http://www. umiacs. umd.edu/~zhuolin/projectlcksvd.html](http://www. umiacs. umd.edu/~zhuolin/projectlcksvd.html). First computing spatial pyramid feature with a four-level spatial pyramid and a SIFT-descriptor codebook with a size of 200, then PCA is applied to reduce the feature dimension to 3,000. (???) Following the same experiment settings of Lazebnik et al. [1], 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 VIII in which we compare our results with SRC [4], CRC [11], LLC [5], LRC [6], LRSIC [6], LRRC [7], SLRRC [7] and other state-of-the-art scene classification approaches [1], [30], [37], [39], [33], [35]. It should be pointed out that, as [35] did, LLC is the original LLC, which uses sparse coding to encode SIFT descriptors [33], while LLC* uses sparse coding to encode the spatial pyramid features. Our method, SRC, CRC, LR, and SLR all use the spatial pyramid feature. The dictionary size of SRC, CRC, LRC, LLC, LRRC, and SLRRC are all 450. LLC and LLC* both have 30 neighborhoods. We set the parameter \( \lambda = 0.5 \) in our method. As Table VIII shows, our method performs the best among all the compared methods and has about 4.7% improvement over the runner-up. The confusion matrix can be seen in Fig. 7 where the average recognition rates for each class are along the diagonal. There is no class that are classified badly and the worst recognition rate is as high as 91%.

Furthermore, we compare our method with SRC [4], LRC [6], LRSIC [6], LRRC [7], and SLRRC [7] in terms of average computing time for classifying a test image. The results are shown in Table IX. Our method is the fourth fastest method.

**C. Action Recognition**

Finally, we test our methods and related algorithms with action recognition, using the UCF50 database [31]. The UCF50 database is one of the largest action recognition database, consisting of realistic videos taken from Youtube. It contains
50 action categories with a total of 6,617 action videos and the categories are of Baseball Pitch, Basketball Shooting, Biking, Diving, Tennis Swing, etc. Some images from this database are shown in Fig. 9.

For this database, we use the action feature representations presented in [40], 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 PCA to reduce the feature dimension to 3,000, then take dimension reduced feature to evaluate our method. SRC [4], CRC [11], LLC [5], LRC [6], LRSIC [6], LRRC [7], and SLRRC [7]. We also compare with other state-of-the-art action recognition methods, such as [41], [42], [43], and [38]. Following the common experiment settings, we test these methods using the 5-fold group-wise cross-validation methodology. The dictionary sizes for SRC, CRC, LRC, and LRSIC are all 1,500, i.e., 30 dictionary atoms for each category. When we evaluate LLC*, we used the original LLC method to encode the action feature and the neighborhood number is 30. We set $\lambda = 0.5$ in our method.

Table X presents the detailed comparison results, in which we note that our method outperforms other methods and makes about 4.2% improvement over the second best. Our confusion matrix, which is shown in Fig. 10 shows a dominant diagonal with no stand-out confusion among the classes. Only two categories (Pizza Tossing and Skijet) obtain relatively bad classification rates. The other categories are all classified well.

The average computing times for classifying a test image are also compared in Table XI. Compared with SRC, LRC, LRSIC, LRRC, and SLRRC, our method is the fastest. It is about four times faster than SRC and is approximately seven times faster than the SLRRC method. Again, this is because there exists little noise in data. So Robust PCA only runs a few dozens iteration to converge.

### D. Speed up with the Fast Algorithm

In the previous experiments, we just solve the full-sized Robust PCA for our discriminative feature learning method.
Most of the time our method is faster than other representation based methods. In this subsection, we show the effectiveness of speeding up our method by solving the Robust PCA problem with the $\ell_1$-filtering algorithm, called Fast Integrated Low Rank Discriminative Feature Learning (F-ILRDFL), when handling large scale databases. We still use Extended YaleB [28], AR [29], PIE [30], and UCF50 [31], but we do not down-sample these databases. The experimental settings are as follows.

Extended YaleB database: The size of data matrix is $37,600 \times 2,600$. We randomly select 20 training images per person and the remaining for testing. SRC, LRC, and LRSIC use all the training samples as the dictionary. The trained dictionary for LRRC and SLRRC has 20 dictionary atoms for every person. In the $\ell_1$-filtering used in our method, the size of seed matrix is $760 \times 1,140$ and we set $\lambda = 0.03$ when we apply Robust PCA to recover the seed matrix.

PIE database: The size of data matrix is $4,096 \times 11,560$ pixels. We randomly select 20 training images per person and the remaining for testing. All the training samples are used as the dictionary for SRC, LRC, and LRSIC. For LRRC and SLRRC, we also train a dictionary with 20 atoms for every person. We set the size of the seed matrix as $2,040 \times 2,720$ and $\lambda = 0.012$.

UCF50 database: The size of data matrix is $14,965 \times 6,617$. We use the 5-fold group-wise cross-validation methodology to evaluate these methods. The dictionary size for SRC, LRC, LRSIC, LRRC, and SLRRC are all 1,500, i.e., 30 dictionary atoms for each person. We set the size of seed matrix as $1,000 \times 1,500$ and $\lambda = 0.038$.

The experimental results are summarized in Table XII. Compared with SRC, LRC, LRSIC, LRRC, and SLRRC, our F-ILRDFL method is the fastest. It is several times faster than these compared methods. When the scale of data matrix is large, the speed of SRC drops dramatically. Though LRC and LRSIC apply SRC to do classification, it does not take much time, since the feature dimension has been reduced by PCA. However, these two methods use Robust PCA to remove noise from training samples class by class. This process is computationally expensive for high dimension data. When the size of data is large, LRRRC and SLRRC are not easy to converge because these two methods have to solve a non-convex problem with a lot of parameters. We also note that though the data matrix is large, its rank is low. So with high probability, the $\ell_1$-filtering in our method produces the same solution as by solving full-scale Robust PCA directly. From Table XII our fast method achieves the best recognition rates.

In conclusion, our F-ILRDFL method not only runs fast, but also achieves the best performance.

### VI. Conclusions and Future Work

We proposed a novel supervised low rank based discriminative feature learning method. Unlike other representation based feature learning methods that separate the feature learning process and the subsequent classification into two steps and optimize the sparsity or the low rankness to extract features, our method learns discriminative features by integrating LatL-RR with ridge regression to minimize the classification error rate directly. By this way, the extracted features is directly related to the recognition rate. We also adopt the $\ell_1$-filtering algorithm to speed up the computation of Robust PCA, which we use for denoising data robustly. Finally, our method has only one parameter in effect, making 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 datasets, by adopting the $\ell_1$-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

Pan Zhou and Chao Zhang are supported by National Key Basic Research Project of China (973 Program) 2011CB302400 and National Nature Science Foundation of China (NSFC, grant no. 61071156). Zhouchen Lin is supported by NSFC (grant nos. 61272341, 61231002, 61121002).

### REFERENCES


