2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

Fast Orthogonal Linear Discriminant Analysis with Applications to Image Classification

Q. L. Ye, and N. Ye
College of Information Science and Technology
Nanjing Forestry University
Nanjing, China
{yening@njfu.edu.cn; yqlcom@njfu.edu.cn}

H. F. Zhang, and C. X. Zhao
College of Computer Science and Technology
Nanjing University of Science and Technology
Nanjing, China
{zhanghf@mail.njust.edu.cn; zhaocx@njust.edu.cn}

***Abstract*—Orthogonalized variant of *Linear Discriminant Analysisis* (LDA) is an effective statistical learning tool for dimension reduction. However, existing orthogonalized LDA algorithms suffer from various drawbacks, including the requirement for expensive computing time. This paper develops an efficient algorithm for dimension reduction, referred to as *Fast Orthogonal Linear Discriminant Analysis* (FOLDA), which adopts an iterative procedure to extract the orthogonal projection vectors. Different from previous efforts, this new approach applies QR decomposition and regression to solve for a new projection vector in each time of iterations, leading to the by far cheaper computational cost. FOLDA can achieve comparable recognition rates to existing orthogonal LDA algorithms. Experimental results on image databases, such as MNIST, COIL20, MEPG-7, and OUTEX, show the effectiveness and efficiency of FOLDA.**

Keywords—linear discriminant analysis; orthogonal linear discriminant analysis; orthogonal projection vectors; QR decomposition.

I. INTRODUCTION

Dimension reduction (DR) is a way of transforming large volumes of high-dimensional data into a meaningful low-dimensional space to further facilitate the underlying recognition tasks, such as face recognition, and text classification. Simply speaking, techniques for dimension reduction can be performed to seek for such an optimal low-dimensional space that is helpful for mitigating the so-called "curse of dimensionality".

Linear dimension reduction finds a meaningful lower-dimensional subspace which provides a compact representation of higher-dimensional data when the data structure is linear in the input space [1, 2]. Two most notable linear dimension reduction techniques are *Principal Component Analysis* (PCA) [1] and *Linear Discriminant Analysis* (LDA) [3] that have gained wide applications in computer vision and pattern recognition, because of their relative simplicity and effectiveness [1, 4, 5, 6]. Many comparative studies between LDA and PCA were made by numerous researchers [3, 5, 6,7], in which the results demonstrated that in the terms of recognition rates LDA outperformed PCA significantly [8], implying that it is important for satisfactory design of any classifier to incorporate supervised information into dimension reduction. Thus, LDA can be applied to a family of pattern recognition problems [1, 6, 9].

The central idea of classical LDA is to find the optimal projection or transformation that better separates different classes. This optimal projection is obtained by maximizing the between-class projection distance and simultaneously minimizing the within-class projection distance, thus achieving the discrimination between classes. The objective function in classical LDA is a trace ratio problem, in which the optimal projection can be computed by a generalized eigenvalue problem. Due to the good discrimination of images from different classes, LDA has a direct connection to classification. Despite the effectiveness and applicability, there are some serious limitations in classical LDA, resulting in many extensions and improvements (we can only cite the most significant ones). Among the most well-known is the undersampled or singularity problem, such as face recognition [3] where the dimension of feature space is much larger than the size of training set. Over the past decade, many algorithms have been proposed to solve this problem. In the research [3], Belhumeur et al. proposed to perform LDA after PCA. The authors in [11] performed LDA after *Singular Value Decomposition* (SVD). A common aspect of these two approaches is to perform LDA after another stage of dimension reduction. Since the rank of the within-class scatter matrix $\mathbf{S}_w$ is upper bounded by m-c, the maximum dimension of the PCA (or SVD) should be reduced to m-c, where m is the size of training set and c denotes the size of classes. However, there is a serious problem in PCA+LDA, which is that the most discriminant information may be lost [12]. To mitigate this problem, there are researchers who suggest keeping the most energy in the PCA stage [8, 13]. Another way to solve the singularity problem in classical LDA is to add the positive constants to the diagonal elements of $\mathbf{S}_w$ [14]. These algorithms, like classical LDA, transform the trace ratio problems into the ratio trace problems, leading to a non-optimal solution [15].

In [16], Duchene *et al*. proposed *Orthogonal Linear Discriminant Analysis* (OLDA). OLDA enforces an orthogonality relationship between the discriminant projection vectors to eliminate the redundant information,

thus achieving the more powerful discriminant projection vectors than the classical ones in the terms of discriminant ratio and mean error probability. They adopt a well-designed iterative procedure, and solve the primal problem of LDA under such an imposed constraint that a new projection vector is orthogonal to all the previously-obtained ones at each iteration. Since the projection vectors are independent of size of classes, there is no limitation on the number of projection vectors available (the number of projection vectors is limited to *c*-1). Similar to OLDA, Xiang *et al*. extended LDA to *Recursive Fisher Linear Discriminant* (RFLD) [8]. Different from OLDA which directly solves a Rayleigh Quotient (RQ) problem with an orthogonality constraint by employing the Lagrange multiplier method, RFLD firstly rewrites the LDA eigenvalue problem as the following generalized eigen-equation $\mathbf{S}_b\mathbf{w}_k = \lambda\mathbf{S}_w\mathbf{w}_k$ and then combines the orthogonality constraint with this equation at each iteration, where $\mathbf{S}_w$ and $\mathbf{S}_b$ denote the within-class scatter matrix and the between-class scatter matrix, respectively. Eventually, in each time of iterations, RFLD still solves a generalized eigen-equation problem, and furthermore, it also guarantees that the samples always involve the newest information by eliminating the old information represented by previously-computed projection vectors. Despite the effectiveness of RFLD, it, like OLDA, is expensive computationally, due to that each iteration involves eigen-decomposition, many operations of matrix inverses as well as matrix multiplications. Still there is an orthogonal LDA algorithm, called *Maximum Margin Criterion* (MMC) [17], which casts the RQ formulation of the classical LDA as a difference formulation. In addition to establishing the orthogonality relationship between projection vectors, MMC can avoid the singularity problem. However, it, like LDA, can only extract at most $c-1$ meaningful features [18]. Both OLDA and RFLD permit to define a best discriminant vector, orthogonal to a set of previously-computed vectors, without using any statistical property of this set [16], which is in contrast to MMC. Furthermore, when the dimensionality in the input space is large, it is not infeasible to apply MMC due to the expensive computation resulting from the solution to the formulated large-scale eigenvalue problem.

In this paper, we develop a novel algorithm for discriminant analysis, referred to as *Fast Orthogonal Linear Discriminant Analysis* (FOLDA), which is essentially based on RFLD [8]. Like RFLD [8], the new approach seeks for the orthogonal projection vectors, iteratively. According to some unique properties of matrix, the solution is empirically obtained. Then, the spectral regression [19] is used to obtain a new orthogonal projection vector at each iteration. The process of solution does not involve eigen-decomposition, multiple matrix inverses, and multiplications, leading to the less computational cost than RFLD. FOLDA does not use any statistical property of the previously-obtained projection vectors and is permitted to define a "best discriminant" vector, orthogonal to them. Therefore, there is no limitation on the number of projection vectors available from FOLDA, which is in contrast to MMC. We also demonstrate the efficiency of FOLDA by analyzing and comparing the time complexities of existing orthogonal algorithms. The experiment, tried out on four image databases, such as MEPG-7, COIL20, MNIST, and OUTEX indicates both the effectiveness and efficiency of our proposed FOLDA algorithm.

II. Related works

In this section, we briefly review two orthogonalized extensions of LDA, such as RFLD [8] and MMC [17]. We consider the problem of representing all of the vectors in a set $\mathbf{X}=[\mathbf{x}_1,\mathbf{x}_2,...,\mathbf{x}_n]$, $\mathbf{x}_i \in \mathrm{R}^d$, where $\mathbf{X}$, n, and d denote the data matrix, the sample size, and the dimensionality, respectively. The class label of the sample $\mathbf{x}_i$ is from the set $\{1,2,...,c\}$, where c is the number of classes. Define n_l as the number of labeled samples from the lth class. Let $\mathbf{W}^{(k-1)} = [\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_{k-1}] \in \mathrm{R}^{d\times(k-1)}$ be a set of the previously-computed $k-1$ orthogonal projection vectors. Denote by $\mathbf{z} \in \mathrm{R}^d (1 \le r \le d)$ a low-dimensional representation of a high-dimensional sample $\mathbf{x}$ in the original input space, where r is the dimensionality of the reduced space. The purpose of DR is to seek for a transformation matrix $\mathbf{W}$, such that a lower representation $\mathbf{z}$ of the sample $\mathbf{x}$ can be calculated as $\mathbf{z} = \mathbf{W}^T\mathbf{x}$, where "$T$" denotes the transpose. Define $\overline{\mathbf{X}}$ as the centered data matrix. Letting $\mathbf{W}^{(k-1)} = [\mathbf{w}_1, ..., \mathbf{w}_{k-1}]$ and

$$\mathbf{V} = \begin{bmatrix} \mathbf{V}^{(1)} & 0 & ... & 0 \\ 0 & \mathbf{V}^{(2)} & ... & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & ... & \mathbf{V}^{(c)} \end{bmatrix} \in \mathrm{R}^{n\times n} \tag{1}$$

Here, $\mathbf{V}^{(l)}$ is a $n_l \times n_l$ matrix with the entries equal to $1/n_l$.

$$\mathbf{w}_1 = \arg\max \frac{\mathbf{w}^T\overline{\mathbf{X}}\mathbf{V}\overline{\mathbf{X}}^T\mathbf{w}}{\mathbf{w}^T\overline{\mathbf{X}}\overline{\mathbf{X}}^T\mathbf{w}} \tag{2}$$

$$\text{and} \quad \mathbf{w}_k = \arg\max \frac{\mathbf{w}^T\overline{\mathbf{X}}^{(k)}\mathbf{V}\left(\overline{\mathbf{X}}^{(k)}\right)^T\mathbf{w}}{\mathbf{w}^T\overline{\mathbf{X}}^{(k)}\left(\overline{\mathbf{X}}^{(k)}\right)^T\mathbf{w}} \tag{3}$$

subject to $\mathbf{w}_k^T\mathbf{w}_1 = \mathbf{w}_k^T\mathbf{w}_2 = \cdots = \mathbf{w}_k^T\mathbf{w}_{k-1} = 0$, where

$$\overline{\mathbf{X}}^k = \overline{\mathbf{X}} - \mathbf{W}^{(k-1)}(\mathbf{W}^{(k-1)})^T\overline{\mathbf{X}} \tag{4}$$

is the updated sample matrix, which can eliminate the old information represented by the previously-computed orthogonal projection vectors. In practice, if there is no orthogonal constraint, the projection vector $\mathbf{w}_k$ can be computed as the eigenvector corresponding to the largest eigenvalue of

$$\overline{\mathbf{X}}^{(k)}\mathbf{V}\left(\overline{\mathbf{X}}^{(k)}\right)^T\mathbf{w} = \lambda\overline{\mathbf{X}}^{(k)}\left(\overline{\mathbf{X}}^{(k)}\right)^T\mathbf{w} \tag{5}$$

However, to make $\mathbf{w}_k$ orthogonal to the previously-computed projection vectors, RFLD combines the orthogonal constraint in (3) with (5), leading to

solving $\mathbf{C}^k\mathbf{w}=\lambda\mathbf{B}^k\mathbf{w}$, in which $\mathbf{B}^k=(\mathbf{X}_v,\mathbf{w}_1,...,\mathbf{w}_{k-1})^T\in \mathrm{R}^{(d+k-1)\times d}$ ($\mathbf{X}_v=\overline{\mathbf{X}}^{(k)}\mathbf{V}(\overline{\mathbf{X}}^{(k)})^T$) and $\mathbf{C}^k=\left(\overline{\mathbf{X}}^{(k)}\left(\overline{\mathbf{X}}^{(k)}\right)^T,0,...,0\right)^T\in \mathrm{R}^{(d+k-1)\times d}$. Since $\mathbf{B}^k$ is of full rank, one finally computes $\mathbf{w}_k$ as the eigenvector corresponding to the largest eigenvalue of $(\mathbf{B}^k)^T\mathbf{C}^k\mathbf{w}=\lambda\left((\mathbf{B}^k)^T\mathbf{B}^k\right)\mathbf{w}$.

MMC [17] is a very simple algorithm for orthogonal dimension reduction. It transforms the objective function in the classical LDA into

$$\mathbf{W}=\arg\max_{\mathbf{W}^T\mathbf{W}=\mathbf{I}} tr\left(\mathbf{W}^T\overline{\mathbf{X}}\mathbf{V}\overline{\mathbf{X}}^T\mathbf{W}-\mathbf{W}^T\overline{\mathbf{X}}\overline{\mathbf{X}}^T\mathbf{W}\right) \qquad (6)$$

The MMC objective function is formulated in a difference form of the classical LDA. The projection vectors are selected as the eigenvectors corresponding to the first r largest eigenvalues of $\mathbf{S}_b-\mathbf{S}_w$. There is no singular problem in MMC, which is clear. The number of the most discriminant projection vectors employed to constitute the transformation matrix is limited to $c-1$ [18].

III. The Proposed Algorithm

In this section, we introduce our FOLDA. The theoretical justification of our algorithm will be presented in next section.

Denote by $\overline{\mathbf{X}}$ the centered data matrix. The algorithmic procedure of FOLDA is stated as follows:

1) PCA projection. We throw away the components corresponding to zeroes eigenvalues of $\mathbf{S}_t$ to project the sample $\overline{\mathbf{x}}_i$ into the PCA subspace. Denote by $\mathbf{W}_{\mathrm{PCA}}$ the transformation matrix of PCA. Note that the rank of the new formed data matrix is equal to the number of features.

2) Generate a set of response vectors of $\mathbf{V}$ $\mathbf{A}=[\mathbf{a}_1,\mathbf{a}_2,...,\mathbf{a}_{c-1}]$. Let $\mathbf{A}=\mathbf{W}^T\overline{\mathbf{X}}$. According to [19], the LDA eigenvalue problem can be transformed into $\mathbf{VA}=\lambda\mathbf{A}$. For this eigenvalue problem, there are c eigenvectors of $\mathbf{V}$ associated with non-zero eigenvalues, which are

$$\mathbf{a}_k=[\underbrace{0,...,0}_{\sum_{k=1}^{l-1}n_k},\underbrace{1,...,1}_{n_k},\underbrace{0,...,0}_{\sum_{k=l+1}^{c}n_k}]^T,\quad k=1,2,...,c. \qquad (7)$$

It follows from [19] that we generate the set of response vectors of $\mathbf{V}$ $\mathbf{A}$.

3) Compute the projection vector $\mathbf{w}_1$. Let us solve $\mathbf{w}_1$ by optimizing $\mathbf{w}_1=\arg\min_{\mathbf{w}}(\overline{\mathbf{X}}^T\mathbf{w}-\mathbf{a}_1)^2$.

4) Compute the projection vector $\mathbf{w}_k$ ($c>k>1$). The projection vector $\mathbf{w}_k$, orthogonal to previously computed $k-1$ vectors represented by $\mathbf{W}^{(k-1)}(=[\mathbf{w}_1,...,\mathbf{w}_{k-1}])$, can be computed as follows:

- Update the training set by subtracting the old information represented by the previously $k-1$ projection vectors, i.e. $\overline{\mathbf{X}}^k=\overline{\mathbf{X}}-\mathbf{W}^{(k-1)}(\mathbf{W}^{(k-1)})^T\overline{\mathbf{X}}$.
- Perform QR decomposition of $\mathbf{W}^{(k-1)}$ to obtain two matrices $\mathbf{Q}^{(k-1)}$ and $\mathbf{R}^{(k-1)}$, where $\mathbf{Q}^{(k-1)}$ is an orthonormal matrix and $\mathbf{R}^{(k-1)}$ an order $k-1$ upper triangular matrix.
- Define $\mathbf{w}=\mathbf{Q}^{(k-1)}\mathbf{z}$, $\mathbf{G}^{(k-1)}=\overline{\mathbf{X}}^k\mathbf{Q}^{(k-1)}$, and $\mathbf{G}^{(k-1)}\mathbf{z}=\mathbf{a}_k$. We calculate $\mathbf{z}_k$ by optimizing $\mathbf{z}_k=\arg\min((\mathbf{G}^{(k-1)})^T\mathbf{z}-\mathbf{a}_k)^2$.
- Compute $\mathbf{w}_k$ as

$$\mathbf{w}_k=\mathbf{Q}^{(k-1)}\mathbf{C}\mathbf{z}_k \qquad (8)$$

where $\mathbf{C}=[0,...,\mathbf{I}_{d-k+1}]^T$ in which $\mathbf{I}_{d-k+1}$ is an identity matrix of dimensions $(d-k+1)\times(d-k+1)$.

5) If needed, go through the iteration from step 4) again to extract more feature vectors. Here, it should be pointed out that only there are $c-1$ response vectors generated. When the number of feature vectors to be extracted are more than $c-1$, we reuse these $c-1$ response vectors.

6) FOLDA Embedding. Denote $\mathbf{W}_{\mathrm{FOLDA}}=[\mathbf{w}_1,\mathbf{w}_2,...,\mathbf{w}_k]$. The embedding is as follows

$$\mathbf{x}\to\mathbf{p}=\left(\mathbf{W}_{\mathrm{PCA}}\mathbf{W}_{\mathrm{FOLDA}}\right)\mathbf{x} \qquad (9)$$

where $\mathbf{p}$ is a $r-$ dimensional vector representing the new coming sample $\mathbf{x}$ in the reduced low-dimensional space and $\mathbf{W}_{\mathrm{FOLDA}}$ is the transformation matrix of FOLDA.

IV. Justification

A. Optimal Orthogonal Embedding

Our proposed algorithm finds a set of orthogonal projection vectors $\mathbf{w}_1,\mathbf{w}_2,...,\mathbf{w}_k$ which minimizes the following function

$$f(\mathbf{w})=\frac{\mathbf{w}^T\overline{\mathbf{X}}\mathbf{V}\overline{\mathbf{X}}^T\mathbf{w}}{\mathbf{w}^T\overline{\mathbf{X}}\overline{\mathbf{X}}^T\mathbf{w}}.$$

Therefore, $\mathbf{w}_1,\mathbf{w}_2,...,\mathbf{w}_k$ are orthogonal projection vectors which minimizes $f(\mathbf{w})$ subject to the constraint $\mathbf{w}_k^T\mathbf{w}_1=\mathbf{w}_k^T\mathbf{w}_2=\cdots=\mathbf{w}_k^T\mathbf{w}_{k-1}=0.$

Just as RFLD, the objective function of our FOLDA aims to solve

$$\mathbf{w}_1=\arg\max\frac{\mathbf{w}^T\overline{\mathbf{X}}\mathbf{V}\overline{\mathbf{X}}^T\mathbf{w}}{\mathbf{w}^T\overline{\mathbf{X}}\overline{\mathbf{X}}^T\mathbf{w}} \qquad (10)$$

and

$$\mathbf{w}_k=\arg\max\frac{\mathbf{w}^T\overline{\mathbf{X}}\mathbf{V}\overline{\mathbf{X}}^T\mathbf{w}}{\mathbf{w}^T\overline{\mathbf{X}}\overline{\mathbf{X}}^T\mathbf{w}} \qquad (11)$$

$$\text{subject to } \left[\mathbf{W}^{(k-1)}\right]^T \mathbf{w}_k = 0. \tag{12}$$

Next, we use an efficient method to solve for the set of orthogonal projection vectors $\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_k$.

Existing orthogonal LDA algorithms [8, 16] solve the problem in (10) by using a traditional matrix eigen-decomposition technique, such as *Singular Value Decomposition* (SVD) which leads to the high computing complexity. To address this problem, we apply *Spectral Regression* (SR) [19]. To solve the problem (10) efficiently, we first use the following theorem.

Theorem 1. Let $\mathbf{a}$ be the eigenvector of eigen-equation $\mathbf{V}\mathbf{a} = \lambda\mathbf{a}$ with eigenvalue λ. If $\mathbf{X}^T\mathbf{w}_1 = \mathbf{a}$, then $\mathbf{w}_1$ is the eigenvector of eigen-problem in (10) with the same eigenvalue λ.

The proof of theorem 1 is similar as that in [19]. According to the theorem 1, we can directly obtain $\mathbf{a}$ by solving $\mathbf{V}\mathbf{a} = \lambda\mathbf{a}$. There is only one non-zero eigenvalue of matrix $\mathbf{V}^{(l)}$, since the rank of $\mathbf{V}^{(l)}$ is one. Therefore, there are c eigenvectors with the same eigenvalue 1, which are

$$\mathbf{a}_k = [\underbrace{0,...,0}_{\sum_{k=1}^{l-1} n_k}, \underbrace{1,...,1}_{n_k}, \underbrace{0,...,0}_{\sum_{k=l+1}^{c} n_k}], \quad k = 1,2,...,c$$

According to [19], we can generate $c-1$ exact response vectors of $\mathbf{V}$. When $\mathbf{a}$ is obtained, we find $\mathbf{w}_1$ which satisfies $\mathbf{X}^T\mathbf{w}_1 = \mathbf{a}_1$ by solving a least squares problem

$$\mathbf{w}_1 = \arg\min (\mathbf{X}^T\mathbf{w} - \mathbf{a}_1)^2 \tag{13}$$

Many efficient algorithms, such as LSQR [21], can handle very large scale least squares problems.

Now, we aim to compute $\mathbf{w}_k$ ($k > 1$). Similar to RFLD, we update the data matrix first to keep the newest but useful formation in training set by $\overline{\mathbf{X}}^k = \overline{\mathbf{X}} - \mathbf{W}^{(k-1)}(\mathbf{W}^{(k-1)})^T\overline{\mathbf{X}}$. The QR decomposition of $\mathbf{W}^{(k-1)}$ gives two matrices $\mathbf{Q}^{(k-1)}$ and $\mathbf{R}^{(k-1)}$, such that we have $\mathbf{W}^{(k-1)} = \mathbf{Q}^{(k-1)}\mathbf{R}^{(k-1)}$, where $\mathbf{Q}^{(k-1)}$ is an orthonormal matrix and $\mathbf{R}^{(k-1)}$ an order $k-1$ upper triangular matrix. Note that $\left[\mathbf{Q}^{(k-1)}\right]^T\mathbf{Q}^{(k-1)} = \mathbf{I}$. Since $\mathbf{W}^{(k-1)}$ is an orthogonal set, it is of full rank. Naturally, we can write $\left[\mathbf{Q}^{(k-1)}\right]^T\mathbf{W}^{(k-1)} = \begin{bmatrix}\mathbf{R}^{(k-1)} \\ \mathbf{0}\end{bmatrix}$, where $\mathbf{0}$ is a zero matrix of dimensions $(d-k+1)\times(k-1)$. Let $\mathbf{w}_k = \mathbf{Q}^{(k-1)}\mathbf{z}_k = \mathbf{Q}^{(k-1)}\begin{bmatrix}\mathbf{y}_k \\ \mathbf{v}_k\end{bmatrix}$, where $\mathbf{y}_k$ is a vector of the first $k-1$ components of $\mathbf{z}_k$ and $\mathbf{v}_k$ of the last $d-k+1$ components of $\mathbf{z}_k$. Thus, we have

$$\left[\mathbf{W}^{(k-1)}\right]^T\mathbf{w}_k = \left[(\mathbf{R}^{(k-1)})^T \quad \mathbf{0}^T\right]\begin{bmatrix}\mathbf{y}_k \\ \mathbf{v}_k\end{bmatrix} = \mathbf{0} \tag{14}$$

and hence the following theorem is obtained

Theorem 2. The vector $\mathbf{y}_k$ in (14) is a zero vector.

Proof: Denote by $(\mathbf{R}^{(k-1)})_i^T$, $i = 1,...,k-1$ the ith row of $(\mathbf{R}^{(k-1)})^T$. Let $(\mathbf{y}_k)_i$ $i = 1,...,k-1$ the ith element of $\mathbf{y}_k$. It follows from (14) that $(\mathbf{R}^{(k-1)})^T\mathbf{y}_k = \mathbf{0}$, which means that $(\mathbf{R}^{(k-1)})_i^T\mathbf{y}_k = 0$. Since $\mathbf{R}^{(k-1)}$ is an order $k-1$ upper triangular matrix, $(\mathbf{R}^{(k-1)})^T$ is an order $k-1$ lower triangular matrix. It is obvious that $(\mathbf{R}^{(k-1)})_1^T\mathbf{y}_k = 0$ if and only if $(\mathbf{y}_k)_1 = 0$. For $(\mathbf{R}^{(k-1)})_2^T$, it has at most two nonzero elements. Since $(\mathbf{y}_k)_1 = 0$ and $(\mathbf{R}^{(k-1)})_2^T\mathbf{y}_k = 0$,we have $(\mathbf{y}_k)_2 = 0$. The rest can be also deduced by analogy. Finally, we have $\mathbf{y}_k = \mathbf{0}$.■□

Theorem 2 tells us that if we set $\mathbf{y}_k$ equal to zero, $\mathbf{W}^{(k-1)}$ and $\mathbf{w}_k$ must satisfy orthogonality condition. Defining $\mathbf{w}_k = \mathbf{Q}^{(k-1)}\mathbf{z}_k$, it is possible to reformulate the objective function in (11) as

$$\mathbf{z}_k = \arg\max \frac{\mathbf{z}^T\left[\mathbf{Q}^{(k-1)}\right]^T\overline{\mathbf{X}}^k\mathbf{V}\left(\overline{\mathbf{X}}^k\right)^T\mathbf{Q}^{(k-1)}\mathbf{z}}{\mathbf{z}^T\left[\mathbf{Q}^{(k-1)}\right]^T\overline{\mathbf{X}}^k\left(\overline{\mathbf{X}}^k\right)^T\mathbf{Q}^{(k-1)}\mathbf{z}} \tag{15}$$

Letting $\mathbf{G}^{(k-1)} = \left(\overline{\mathbf{X}}^k\right)^T\mathbf{Q}^{(k-1)}$, the problem (15) becomes

$$\mathbf{z}_k = \arg\max \frac{\mathbf{z}^T\left[\mathbf{G}^{(k-1)}\right]^T\mathbf{V}\mathbf{G}^{(k-1)}\mathbf{z}}{\mathbf{z}^T\left[\mathbf{G}^{(k-1)}\right]^T\mathbf{G}^{(k-1)}\mathbf{z}} \tag{16}$$

The problem (16) contains $k-1$ orthogonal projection vectors. We can obtain the solution to (16) by solving $\left[\mathbf{G}^{(k-1)}\right]^T\mathbf{V}\mathbf{G}^{(k-1)}\mathbf{z} = \lambda\left[\mathbf{G}^{(k-1)}\right]^T\mathbf{G}^{(k-1)}\mathbf{z}$ with eigenvalue λ. However, expensive computing cost is required in solving this eigenvalue problem. It is easy to observe that the equations in (16) and (10) have a very similar formulation. Therefore, the spectral regression technique can be directly used.

Then, we can find $\mathbf{z}_k$, which should satisfy $\mathbf{G}^{(k-1)}\mathbf{z}_k = \mathbf{a}_k$, by solving the following least squares problem

$$\mathbf{z}_k = \arg\min (\mathbf{G}^{(k-1)}\mathbf{z} - \mathbf{a}_k)^2 \tag{17}$$

Finally, we compute the components of $\mathbf{z}_k$, solution of the problem (11) with constraints (12):

$$\mathbf{w}_k = \mathbf{Q}^{(k-1)}\mathbf{C}\mathbf{z}_k \tag{18}$$

where $\mathbf{C} = [\ 0,....,\mathbf{I}_{d-k+1}]^T$ in which $\mathbf{I}_{d-k+1}$ is an identity matrix of dimensions $(d-k+1)\times(d-k+1)$. Recall that there is no limitation on the number of projection vectors in RFLD. In solving $\mathbf{w}_k$, we have to use the response vector of $\mathbf{V}$ $\mathbf{a}_k$. However, there are only $c-1$ exact response vectors from $\mathbf{V}$, leading to the limitation on the number of

projection vectors. To eliminate this problem, we re-use all these response vectors, when the desired subspace dimension is greater than $c-1$. Note that similar idea in solving $\mathbf{a}_k$ was also discussed in the prior work [22] for binary classification problems. Our algorithm is an extension of this idea in dimension reduction.

In some steps of our algorithm, we have to compute some regression problems. However, these regressions involve a singularity problem. It is observed that RFLD and classical LDA work under the assumption that the matrix $\overline{\mathbf{X}}\overline{\mathbf{X}}^T$ or $\overline{\mathbf{X}}^{(k)}\left(\overline{\mathbf{X}}^{(k)}\right)^T$ is nonsingular. However, this assumption does not hold in many real applications where sometimes the number of samples in the training set tends to be much smaller than that number of features in each sample. To overcome this problem, we can use the way in [3], which suggests projecting the sample set to the PCA space in the first stage and then performing the LDA or RFLD dimension reduction. Still there are two considerations for the use of PCA. First, some useless information, such as noise, can be eliminated by keeping most of energy. Second, the computational efficiency can be improved. Therefore, we use PCA to throw away the components corresponding to zeros eigenvalues of each of $\overline{\mathbf{X}}\overline{\mathbf{X}}^T$ and $\overline{\mathbf{X}}^{(k)}\left(\overline{\mathbf{X}}^{(k)}\right)^T$ to guarantee the non-singularity in our problem and make fair comparisons. It should be pointed that in this work the within-class scatter $\mathbf{S}_w$ replaced by $\overline{\mathbf{X}}\overline{\mathbf{X}}^T$, such that there will be no information loss in the PCA step. Similar discussion can be found in [20].

B. Time Complexity Comparison

Our algorithm FOLDA and traditional orthogonal DR methods RFLD and MMC all generate the orthogonal projection vectors. However, they are different in the terms of solution, leading to different time complexities. We now analyze their time costs.

MMC directly solves the standard eigenvalue problem of $\left(\overline{\mathbf{X}}\mathbf{V}\overline{\mathbf{X}}^T-\overline{\mathbf{X}}\overline{\mathbf{X}}^T\right)\mathbf{w}=\lambda\mathbf{w}$. To solve this eigenvector problem, we need first to compute matrices $\overline{\mathbf{X}}\mathbf{V}\overline{\mathbf{X}}^T$ and $\overline{\mathbf{X}}\overline{\mathbf{X}}^T$. The time complexities for calculating them are $\mathrm{O}(dn^2+nd^2)$ and $\mathrm{O}(nd^2)$, respectively. To project the data points into a r-dimensional subspace, we must compute the first r largest eigenvectors of matrix $\overline{\mathbf{X}}\mathbf{V}\overline{\mathbf{X}}^T-\overline{\mathbf{X}}\overline{\mathbf{X}}^T$ with dimensions $d\times d$, whose time complexity is $\mathrm{O}(rd^2)$. Thus, the total time complexity of the MMC algorithm is

$$\mathrm{O}(dn^2+(n+r)d^2) \tag{19}$$

In RFLD, the first step is to find $\mathbf{w}_1$ by solving the generalized eigenvalue problem $\overline{\mathbf{X}}\mathbf{V}\overline{\mathbf{X}}^T\mathbf{w}=\lambda\overline{\mathbf{X}}\overline{\mathbf{X}}^T\mathbf{w}$. Similar to MMC, it requires computing $\overline{\mathbf{X}}\mathbf{V}\overline{\mathbf{X}}^T$ and $\overline{\mathbf{X}}\overline{\mathbf{X}}^T$, whose time complexity is $\mathrm{O}(dn^2+nd^2)$. To solve the above generalized eigenvalue problem, we need first to take the time complexity of $\mathrm{O}(d^3)$ to compute the SVD of the matrix $\overline{\mathbf{X}}\overline{\mathbf{X}}^T$ [23]. To achiever $\mathbf{w}_1$, we need to compute the first largest eigenvector of a $d\times d$ matrix, leading to the time complexity of $\mathrm{O}(d^2)$. Therefore, the cost of the first step of RFLD is $\mathrm{O}(dn^2+nd^2+d^3)$. In the kth step of RFLD ($k\geq 2$) to compute the kth projection vector, it needs to compute the generalized eigenvalue problem $(\mathbf{B}^k)^T\mathbf{C}^k\mathbf{w}=\lambda\left((\mathbf{B}^k)^T\mathbf{B}^k\right)\mathbf{w}$, where both $\mathbf{C}^k$ and $\mathbf{B}^k$ are of dimensions $(d+k-1)\times d$. To compute this generalized eigenvalue problem, we need first to compute the components of $\mathbf{B}^k$ and $\mathbf{C}^k$, i.e., $\overline{\mathbf{X}}^{(k)}\mathbf{V}\left(\overline{\mathbf{X}}^{(k)}\right)^T$ and $\overline{\mathbf{X}}^{(k)}\left(\overline{\mathbf{X}}^{(k)}\right)^T$. For the computation of these two matrices, the time complexity is $\mathrm{O}(dn^2+nd^2)$. Then, we compute the matrices $(\mathbf{B}^k)^T\mathbf{C}^k$ and $(\mathbf{B}^k)^T\mathbf{B}^k$, whose time complexities are all $\mathrm{O}((d+k-1)d^2)$. The other computation of $\mathbf{w}_1$ is the same as the step1, whose time complexity is $\mathrm{O}(d^2+d^3)$. Thus, the complexity of the kth step is $\mathrm{O}(dn^2+(n+d+k)d^2+d^3)$. This time complexity is analyzed under the assumption that the first $k-1$ projection vectors are known. Therefore, to get the r-dimensional subspace, the time complexity becomes $\mathrm{O}((r-1)dn^2+(p+(r-1)n)d^2+(r-1)d^3)$, in which $p=r\times(r+1)/2-1$. Finally, the total complexity of RFLD is

$$\mathrm{O}(rdn^2+(p+rn)d^2+rd^3) \tag{20}$$

It is observed from (20) that the time complexity depends on the number of samples, the number of dimension in input space, and the dimension of reduced space.

In order to solve for the first projection vector $\mathbf{w}_1$, our algorithm FOLDA only needs to compute $\mathbf{w}_1$ by optimizing the regression problem $\mathbf{w}_1=\arg\min_{\mathbf{w}}(\overline{\mathbf{X}}^T\mathbf{w}-\mathbf{a}_1)^2$. The time complexity of solving this regression problem is $\mathrm{O}(ns)$ [19], where s is the average number of non-zero features for one sample in $\overline{\mathbf{X}}$. The time complexity for computing $\mathbf{w}_k$ is dominated by three parts: QR decomposition, a small number of matrix multiplications, and solving a least squares problem. For the QR decomposition of $\mathbf{W}^{(k-1)}$, the time complexity is $\mathrm{O}(d(k-1)^2)$ [24]. To solve the problem $\mathbf{z}_k=\arg\min(\mathbf{G}^{(k-1)}\mathbf{z}-\mathbf{a}_k)^2$, we first compute $\mathbf{G}^{(k-1)}$ $(=\left(\overline{\mathbf{X}}^k\right)^T\mathbf{Q}^{(k-1)})$, whose time complexity is $\mathrm{O}(d^2n)$. To get $\mathbf{z}_k$, FOLDA requires time complexity of $\mathrm{O}(ne)$ to solve the above regression problem [19], where

e is average number of non-zero elements of each row of $\mathbf{G}^{(k-1)}$. Finally, we compute $\mathbf{w}_k$ as $\mathbf{w}_k = \mathbf{Q}^{(k-1)}\mathbf{C}\mathbf{z}_k$. Since $\mathbf{C}$ is sparse and only the last $(d-k+1)$ diagonal elements are ones, the time complexity in this computation is around $O(d^2)$. Thus, the time complexity used for computing $\mathbf{w}_k$ is around $O(d(k-1)^2)+O(d^2n)+O(ne)$. Similar to RFLD, we must compute the first $k-1$ projection vectors before getting $\mathbf{z}_k$. Therefore, to get the r-dimensional subspace, the time complexity of computing $\mathbf{w}_k$ is $O(dp+(r-1)nd^2+(r-1)ne)$, in which $p=r(r-1)(2r-1)/6-1$. As a result, the total time complexity for FOLDA is

$$O(dp+(r-1)nd^2+(r-1)ne+ns) \quad (21)$$

Both e and s are equal to d in the worst case. In real cases, $r << d$.

Clearly, RFLD has square-time complexity while our algorithm has linear-time complexity with respect to n in computing $\mathbf{w}_1$. In solving the r-dimensional subspace, RFLD also has cubic-time complexity while our FOLDA has only square-time complexity with respect to d. Furthermore, FOLDA is better than RFLDA, in the terms of computational cost used for matrix multiplications. Clearly, FOLDA gains significant computational saving on time.

V. Experimental Result

We evaluate our algorithm on four image databases: a shape image database MPEG-7 [25], an object database COIL20 [26], a handwritten digit database MNIST [27], and a terrain database OUTEX [28]. Maybe, the readers are not familiar with the OUTEX database, thus we simply introduce it. The OUTEX database includes 20 outdoor scene images. In the experiment, four object classes are defined as grass, tree, sky and road, with considerable changes of illumination and shadow. Following [29, 30], the rotation-invariant operators LBP $riu2_{8,1+16,3}$ and 4 bin color histogram were used for extracting features of each object image. Therefore, each image is represented by a 314-dimensional vector. Table I describes the details for each database used in the experiments. For all the databases, a random subset with l (=20%, 30%, 40%, 50%) labeled samples per class are selected for training and the rest for testing.

TABLE I. Data Description. Four Image Databases, A Shape Image Database, An Object Database, A Terrain Database, And A Handwritten Digit Database Are Used In The Experiments.

Database	No. of samples	No. of classes	No. of dimensions
MPEG-7	1400	120	2000
COIL20	1440	20	1024
MNIST	4000	10	784

We investigate the recognition rates of LDA, MMC, RFLD, and FOLDA. All algorithms are implemented in MATLAB 7.1 and carried out experiments on a PC with Intel(R) Core2Duo processor (2.79GHz), 4GB RAM. According to [8], [20], the images are projected into the PCA subspace by throwing away the components corresponding to zero eigenvalues to avoid the singularity problem, for LDA, RFLDA, and FOLDA. Among them, LDA and MMC can extract at most $c-1$ meaningful features [18]. According to [17], MMC does not do any pre-processing due to that it has no singularity problem. We report the mean recognition rate ± standard deviation over 10 random splits on the test set in Table II. Fig.1. shows recognition rate versus the variation of dimensions on MPEG-7, COIL20, MNIST, and OUTEX. From the results, we can see the following main points. Firstly, when comparing LDA, MMC, RFLD, and FOLDA, we observe that both RFLD and FOLDA are comparable, in the terms of recognition rates, which indicates that FOLDA is very effective. Secondly, RFLD and FOLDA are better than LDA and MMC in most cases, in the terms of recognition rates, possibly because of the limitation on the number of discriminant projection vectors resulting from LDA and MMC. In the experiments, MNIST and OUTEX are two typical small-class databases. For MMC, still there is a possible reason resulting in its ineffectiveness. Although MMC also generates a set of orthogonal projection vectors, it uses the statistical property of this set. In contrast to MMC, both OLDA and RFLD permit to define a best discriminant vector, orthogonal to the set of previously-computed vectors, without using any statistical property of this set (similar discussion can be found in [16]).

In Table III, we report the mean computing time over 10 random splits at the optimal dimension. Fig.2 shows the computing time versus the variation of dimensions on MPEG-7, COIL20, MNIST, and OUTEX. From Table III and Fig.2, we can observe that in the terms of computing time, FOLDA is by far faster than RFLD. The computational costs of these two algorithms largely depend on the number of projection vectors. From the Table III, we also observe that the optimal dimension of RFLD is more than that of FOLDA in most cases. It is necessary to point out that FOLDA is by faster than RFLD, even if their optimal dimensions are equal (e.g. on MNIST). FOLDA is faster than MMC on MPEG-7 and COIL20, which is because MMC is time-consuming on high-dimensional datasets. It can be seen that the computing time of RFLD fiercely increases with the increase of the training set size. Our FOLD is contrast to RFLD.

VI. Conclusions

Enforcing the orthogonality relationship between projection vectors usually leads to the optimal discrimination for recognition or classification. However, existing effective orthogonal linear discriminant methods need high computing complexities. In this paper, we have proposed a very efficient linear discriminant analysis, called FOLDA. Different from previous work, FOLDA is inspired by Spectral Regression and QR decomposition, which decomposes the set of previously-computed projection vectors using QR, and then solves for the new projection vector by employing Spectral Regression. In entire process of solution, there is no need to perform eigen-decomposition, leading to the less computing cost. We have

listed the algorithmic procedure of FOLDA, justified this algorithm, and analyzed the time complexity. The experiments, carried out on four image datasets, indicated the effectiveness and efficiency of our algorithm. The idea behind our algorithm is very simple, which can be naturally extended to other dimension reduction methods, including supervised and unsupervised variants.

TABLE II. RECOGNITION RATE (MEAN RECOGNITION RATE± STANDARD DEVIATION %) OF LDA, MMC, RFLD AND FOLDA ON MPEG-7, COIL20, MNIST AND OUTEX. NOTE THE NUMBERS IN PARENTHESES ARE THE OPTIMAL DIMENSIONS AFTER DIMENSION REDUCTION.

Result / Method	MPEG-7 Dataset				COIL20 Dataset			
	20% Train	*30% Train*	*40% Train*	*50% Train*	*20% Train*	*30% Train*	*40% Train*	*50% Train*
LDA	55.3±2.3(19)	59.2±2.4(69)	54.4±2.4(25)	61.2±2.3(69)	92.9±0.7 (19)	93.4±0.8(19)	94.2±0.8(19)	93.9±0.9(19)
MMC	71.5±0.8(69)	75.9±1.1(31)	79.7±1.1(69)	82.5±0.9(63)	94.4±1.3 (19)	96.8±0.6(19)	98.2±0.3(19)	99.1±0.2(19)
RFLD	72.5±0.8(65)	77.6±1.2(71)	80.9±0.5(73)	82.9±0.9(75)	95.9±0.9 (89)	97.7±0.8(13)	98.6±0.3(21)	99.4 ±0.4(15)
FOLDA	72.2±1.0(35)	76.1±1.3(31)	79.7±0.5(37)	82.1±.8 (39)	95.9±1.4 (21)	97.4±2.5(23)	98.6 ±0.4(21)	99.4±0.2(15)
Result / Method	**MNIST Dataset**				**OUTEX Dataset**			
	20% Train	*30% Train*	*40% Train*	*50% Train*	*20% Train*	*30% Train*	*40% Train*	*50% Train*
LDA	50.8±3.2(9)	63.4±2.8(9)	69.0±3.2(9)	72.9±1.4(9)	67.2±0.7(3)	67.3±0.7(3)	67.2±0.6 (3)	67.4±0.5(3)
MMC	67.6±1.0(9)	69.8±0.9(9)	71.3±1.0(9)	71.9±0.3(9)	60.0±0.8(3)	60.2±0.6(3)	59.9±0.5(3)	60.0±0.6(3)
RFLD	87.2±0.7(37)	89.1±0.5(39)	90.3±0.3(39)	90.9±0.4(29)	74.3±0.3(61)	75.5±0.6(87)	75.7±0.4(77)	76.5±0.6(87)
FOLDA	86.8±0.4(31)	89.0±0.5(33)	90.3±0.4(31)	90.9±0.5(31)	74.7±1.1(61)	75.6±0.8(29)	76.0±0.4(31)	76.4±0.7(29)

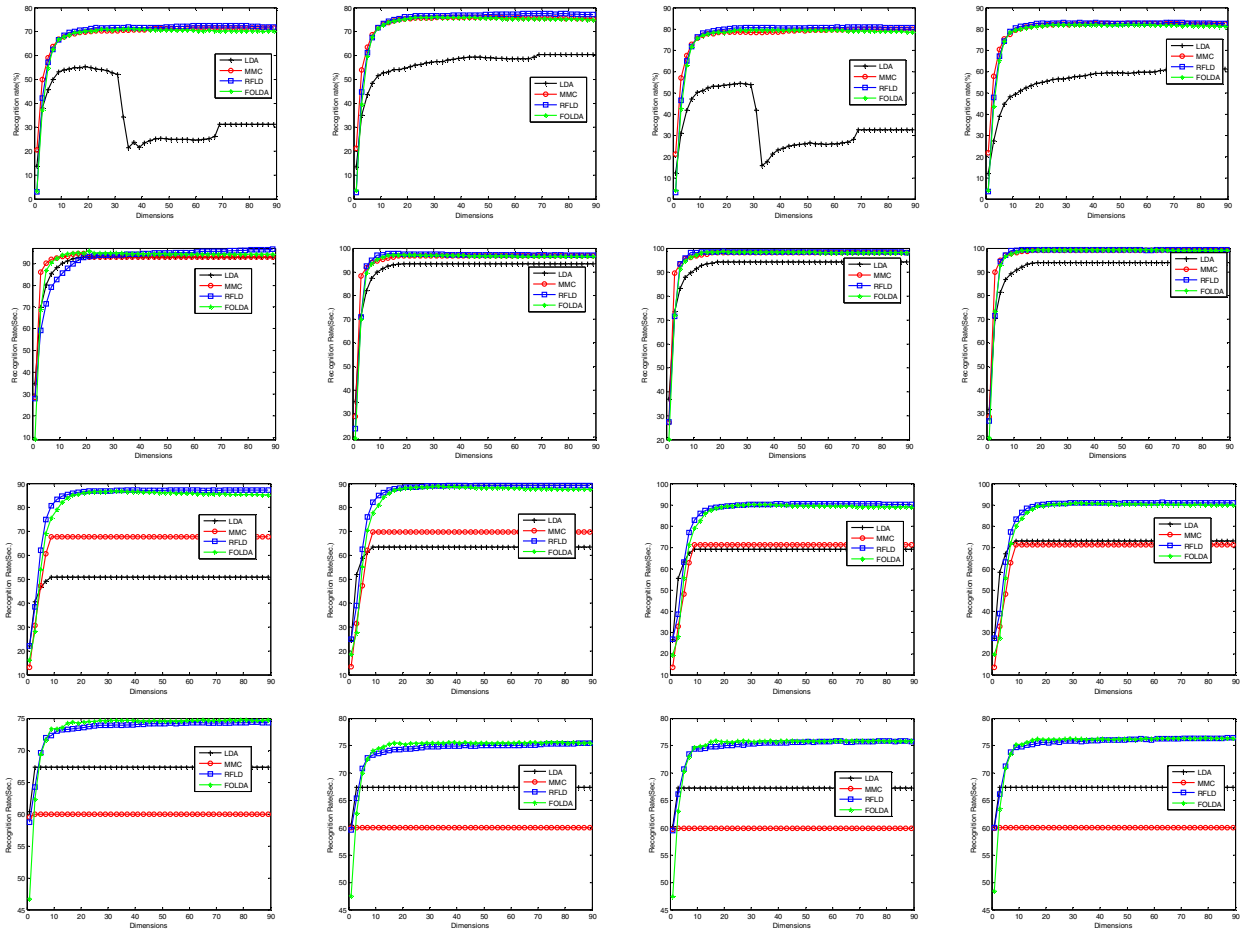

Fig.1. Recognition rate versus the variation of dimensions on MPEG-7, COIL20, MNIST, and OUTEX.

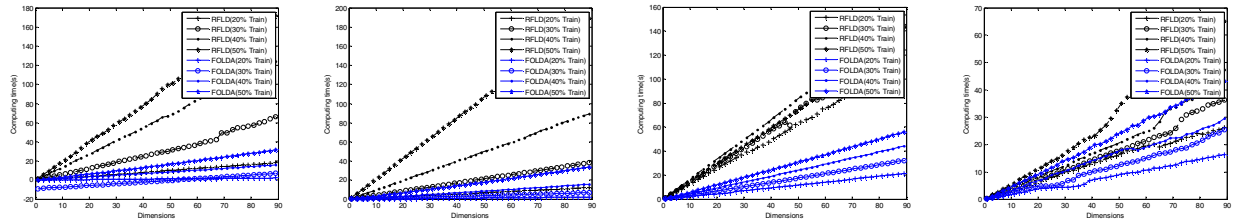

Fig.2. Computing time versus the variation of dimensions on MPEG-7, COIL20, MNIST, and OUTEX.

TABLE III. AVERAGE COMPUTING TIME (SEC.) OF LDA, MMC, RFLD, AND FOLDA ON MPEG, COIL20, MNIST AND OUTEX.

Result Method	MPEG-7 Dataset				COIL20 Dataset			
	20% Train	*30% Train*	*40% Train*	*50% Train*	*20% Train*	*30% Train*	*40% Train*	*50% Train*
LDA	0.7544	3.2048	3.5716	9.7619	0.48247	1.91262	4.67463	8.1555
MMC	52.7750	50.3870	69.4756	67.5073	22.5714	24.3851	26.8392	29.4859
RFLD	13.2507	40.8480	99.9506	150.5965	1.9876	5.3927	20.5507	31.6138
FOLDA	1.7961	2.0884	6.2510	9.9547	0.4663	1.6144	3.3508	5.5577
Result Method	**MINST Dataset**				**OUTEX Dataset**			
	20% Train	*30% Train*	*40% Train*	*50% Train*	*20% Train*	*30% Train*	*40% Train*	*50% Train*
LDA	5.2258	6.0149	6.8347	7.6602	0.7105	0.8203	0.5910	1.0649
MMC	4.9844	5.3381	5.3381	5.9256	0.5628	0.5958	0.7189	0.7333
RFLD	45.0733	55.8924	64.2818	76.0427	19.8371	35.5080	36.5736	67.5130
FOLDA	7.0272	11.3364	16.6728	18.7547	11.4416	5.6886	15.1029	13.8071

ACKNOWLEDGMENT

The authors are extremely thankful to Scientific Research Foundation for Advanced Talents and Returned Overseas Scholars of Nanjing Forestry University, China National Funds for Distinguished Young Scientists (31125008), Jiangsu Qing Lan Project, Jiangsu talent peaks of six fields Project, Jiangsu Science Foundation (BK2012399), and National Science Foundations of China (61101197 and 61272220) for support.

REFERENCES

[1] M. Turk, and A.P. Pentland., "Face Recognition Using Eigenfaces", IEEE Conf. Computer Vision and Pattern Recognition, 1991.

[2] J. Yang, D. Zhang, J.-Y. Yang, and B. Niu, "Globally Maximizing, Locally Minimizing: Unsupervised Discriminant Projection with Applications to Face and Palm Biometrics", IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 29, no. 4, Apr. 2007, pp. 650-664,.

[3] P.N. Belhumeur, J.P. Hespanha, and D.J. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection", IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 19, no. 7, July 1997, pp. 711-720.

[4] K. Liu, Y.Q. Cheng, J.Y. Yang, X. Liu., "An efficient algorithm for Foley-Sammon optimal set of discriminant vectors by algebraic method", International Journal of Pattern Recognition and Artificial Intelligence, vol. 6, no. 1992, pp. 817–829.

[5] D.L. Swets, J. Weng, "Using discriminant eigenfeatures for image retrieval", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 18, no. 8, 1996, pp. 831–836.

[6] K. Etemad, R. Chellappa, "Discriminant analysis for recognition of human face images", Journal of the Optical Soceity of America, vol. 14, 1997, pp. 1724–1733.

[7] D. Cai, X.F. He, "Manifold Adaptive Experimental Design for Text Categorization", IEEE Transactions on Knowledge and Data Engineering, vol. 24, no.4, 2012, pp. 707-719.

[8] C. Xiang, X.A. Fan, T.H. Lee, "Face recognition using recursive fisher linear discriminant", IEEE Transactions on Image Process, vo. 15, no. 8, 2006, 2097–2105.

[9] C. Liu, H. Wechsler, "Robust coding schemes for indexing and retrieval from large face databases", IEEE Transactions on Image Process, vol.9, no.1, 2000, 132–137.

[10] T. Hastie, A. Buja, and R. Tibshirani, "Penalized discriminant analysis," Ann. Stat., vol. 23, no. 1,1995, pp. 73–102.

[11] K. Torkkola, "Linear discriminant analysis in document classification". In Proc. IEEE ICDM Workshop Text Mining, 2001.

[12] D.Q. Dai and P. C. Yuen, "Face recognition by regularized discriminant analysis," IEEE Trans. Syst., Man, Cybern. B, Cybern., vol. 37, no. 4,2007, pp. 1080–1085.

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

[14] J. H. Friedman, "Regularized discriminant analysis". Journal of the American Statistical Association, vol. 84, no.405,1989, pp. 165–175.

[15] H. Wang, S. Yan, D. Xu, X. Tang, and T. Huang, "Trace Ratio vs. Ratio Trace for Dimensionality Reduction," Prceedings of Conference on Computer Vision and Pattern Recognition (CVPR), 2007.

[16] J. Duchene and S. Leclercq, "An optimal transformation for discriminant and principal component analysis", IEEE Trans. on PAMI, vol.10, no. 6, 1988, 978–983.

[17] H. Li, T. Jiang, and K. Zhang, "Efficient and Robust Feature Extraction by Maximum Margin Criterion", IEEE Transactions on Neural Networks, vol. 17, no. 1, 2006, pp. 157–165.

[18] J. Liu, S. Chen and X. Tan. A Study on Three Linear Discriminant Analysis Based Methods in Small Sample Size Problem, Pattern Recognition, 41, no. 1, 2008, pp.102-116.

[19] D. Cai, "Spectral regression: A regression framework for efficient regularized subspace learning," Ph.D. dissertation, Dept. Comput. Sci., Univ. Illinois Urbana-Champaign, Urbana, May, 2009.

[20] D. Cai, X. He, and J. Han. Using Graph Model for Face Analysis, Department of Computer Science Technical Report No. 2636, University of Illinois at Urbana-Champaign (UIUCDCS-R-2005-2636), Sept. 2005.

[21] C. C. Paige and M. A. Saunders, "LSQR: An algorithm for sparse linear equations and sparse least squares", ACM Transactions on Mathematical Software, vol.8, no.1, 1982, pp. 43–71.

[22] Mario R. Guarracino, Danilo Abbate, Roberto Prevete, "Nonlinear knowledge in learning models", In Workshop on Prior Conceptual Knowledge in Machine Learning and Knowledge Discovery, European Conference on Machine Learning, 2007, pp. 29–40.

[23] X. F. He, D. Cai, and J.W. Han, "Learning a Maximum Margin Subspace for Image Retrieval", IEEE Transactions on Knowledge and Data Engineering, vol. 20, no. 2, 2008, pp.189-201.

[24] G.H. Golub and C.F. Van Loan, "Matrix Computations", third ed. Baltimore, M.D.: The Johns Hopkins Univ. Press, 1996.

[25] L. Latecki, R. Lakamper, and T. Eckhardt, "Shape descriptors for non-rigid shapes with a single closed contour," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., vol. 1. Hilton Head Island, SC, 2000, pp. 424–429.

[26] S. A. Nene, S. K. Nayar, and H. Murase. Columbia object image library (COIL-20), Columbia Univ., New York, Tech. Rep. CUCS-005-96, 1996.

[27] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proc. IEEE, vol. 86, no. 11, 1998, pp. 2278–2324.

[28] OUTEX Database. (online), /http://www.outex.oulu.fiS.

[29] P. Matti, N. Tom, M. Topi, T. Markus. "View-based recognition of real-world textures", Pattern Recognition, vol. 37, no. 2, 2005, pp. 313-323.

[30] M. J. Procopio, T. Strohmann, A. R. Bates, G. Grudic, and J. Mulligan. Using Binary Classifiers to Augment Stereo Vision for Enhanced Autonomous Robot Navigation, CU/Boulder Technical Report, April 2007.