Solving the face recognition problem using QR factorization

JIANQIANG GAO(a,b)
(a) Hohai University
College of Computer and
Information Engineering
Jiangning District, 210098, Nanjing
P.R. CHINA
gaojianqiang82@126.com

LIYA FAN(b)
(b) Liaocheng University
School of Mathematical Sciences
Hunan Road NO.1,
252059, Liaocheng
P.R. CHINA
fanliya63@126.com

LIZHONG XU(a)
(a) Hohai University
College of Computer and
Information Engineering
Jiangning District, 210098, Nanjing
P.R. CHINA
hhice@126.com

Abstract: Inspired and motivated by the idea of LDA/QR presented by Ye and Li, in addition, by the idea of WKDA/QR and WKDA/SVD presented by Gao and Fan. In this paper, we first consider computational complexity and efficacious of algorithm present a PCA/range(S_b) algorithm for dimensionality reduction of data, which transforms firstly the original space by using a basis of range(S_b) and then in the transformed space applies PCA. Considering computationally expensive and time complexity, we further present an improved version of PCA/range(S_b), denoted by PCA/range(S_b)-QR, in which QR decomposition is used at the last step of PCA/range(S_b). In addition, we also improve LDA/GSVD, LDA/range(S_b) and PCA by means of QR decomposition. Extensive experiments on face images from UCI data sets show the effectiveness of the proposed algorithms.

Key–Words: QR decomposition, PCA/range(S_b)-QR, PCA/GSVD-QR, LDA/range(S_b)-QR, PCA/QR, Algorithm

1 Introduction

Dimensionality reduction is important in many applications of data mining, machine learning and face recognition [1-5]. Many methods have been proposed for dimensionality reduction, such as principal component analysis (PCA) [2,6-10] and linear discriminant analysis (LDA) [2-4,11-15]. PCA is one of the most popular methods for dimensionality reduction in compression and recognition problem, which tries to find eigenvector of components of original data. LDA aims to find an optimal transformation by minimizing the within-class distance and maximizing the between-class distance, simultaneously. The optimal transformation is readily computed by applying the eigen-decomposition to the scatter matrices. An intrinsic limitation of classical LDA is that its objective function requires one of the scatter matrices being nonsingular. Classical LDA can not solve small size problems [16-18], in which the data dimensionality is larger than the sample size and then all scatter matrices are singular.

In recent years, many approaches have been proposed to deal with small size problems. We will review four important methods including PCA [2,6-10], LDA/range(S_b) [7,19], LDA/GSVD [19-21] and LDA/QR [22-23]. The difference of these four methods can be briefly described as follows: PCA tries to find eigenvectors of covariance matrix that corresponds to the direction of principal components of original data. It may discard some useful information. LDA/range(S_b) [7,19] first transforms the original space by using a basis of range(S_b) and then in the transformed space the minimization of within-class scatter is pursued. The key idea of LDA/range(S_b) is to discard the null space of between-class scatter S_b, which contains no useful information, rather than discarding the null space of within-class scatter S_w, which contains the most discriminative information. LDA/GSVD is based on generalized singular value decomposition (GSVD) [17,20-21,24-26] and can deal with the singularity of S_w. Howland et. al [27] established the equivalence between LDA/GSVD and a two-stage approach, in which the intermediate dimension after the first stage falls within a specific range and then LDA/GSVD is required for the second stage. Park et. al [19] presented an efficient algorithm for LDA/GSVD. Considering computationally expensive and time complexity, Ye and Li [22-23] presented a LDA/QR algorithm. LDA/QR is a two-stage method, the first stage maximizes the separation between different classes by applying QR decomposition and the second stage incorporates both between-class and within-class information by applying LDA to the "reduced" scatter matrices resulted from the first stage. Gao and Fan [32] presented WKDA/QR and WKDA/SVD algorithms which are used weighted function and kernel function by QR decomposition and singular value decomposition to solve small sample size problem.

In this paper, we first present an extension of PCA by means of the range of S_b, denoted by PCA/range(S_b). Considering computationally expensive and time complexity, we also study an improved version of PCA/range(S_b), in which QR decomposition is used at the last step of PCA. We denote the improved version by PCA/range(S_b)-QR. In addition, we will improve LDA/GSVD, LDA/range(S_b) and PCA with QR decomposition and obtain LDA/GSVD-QR, LDA/range(S_b)-QR-1, LDA/range(S_b)-QR-2, PCA/QR-1 and PCA/QR-2. The effectiveness of the presented methods is compared by large number of experiments with known ORL database and Yale database.

The rest of this paper is organized as follows. The review of previous approaches are briefly introduced and discussed in Section 2. The detailed descriptions about improvement of algorithms are presented in Section 3. In Section 4, Experiments and analysis are reported. Section 5 concludes the paper.

2 Review of previous approaches

In this section, we first introduce some important notations used in this paper. Given a data matrix $A = [a_1, \cdots, a_N] \in R^{n\times N}$, where $a_1, \cdots, a_N \in R^n$ are samples. We consider finding a linear transformation $G \in R^{n\times l}$ that maps each a_i to $y_i \in R^l$ with $y_i = G^T a_i$. Assume that the original data in A is partitioned into k classes as $A = [A_1, \cdots, A_k]$, where $A_i \in R^{n\times N_i}$ contains data points of the ith class and $\sum_{i=1}^{k} N_i = N$. In discriminant analysis, within-class, between-class and total scatter matrices are defined as follows [3]:

$$\begin{aligned} S_b &= \tfrac{1}{N}\textstyle\sum_{i=1}^{k} N_i(m_i-m)(m_i-m)^T, \\ S_w &= \tfrac{1}{N}\textstyle\sum_{i=1}^{k}\sum_{a\in A_i}(a-m_i)(a-m_i)^T, \\ S_t &= \tfrac{1}{N}\textstyle\sum_{i=1}^{N}(a_i-m)(a_i-m)^T, \end{aligned} \tag{1}$$

where $m_i = \frac{1}{N_i}\sum_{a\in A_i} a$ is the centroid of the ith class and $m = \frac{1}{N}\sum_{j=1}^{N} a_j$ is the global centroid of the training data set. If we assume

$$\begin{aligned} H_b &= \tfrac{1}{\sqrt{N}}[\sqrt{N_1}(m_1-m), \cdots, \sqrt{N_k}(m_k-m)], \\ H_w &= \tfrac{1}{\sqrt{N}}[A_1 - m_1 e_1^T, \cdots, A_k - m_k e_k^T], \\ H_t &= \tfrac{1}{\sqrt{N}}[a_1 - m, \cdots, a_N - m] = \tfrac{1}{\sqrt{N}}(A - me^T), \end{aligned} \tag{2}$$

then $S_b = H_b H_b^T, S_w = H_w H_w^T$ and $S_t = H_t H_t^T$, where $e_i = (1, \cdots, 1)^T \in R^{N_i}$ and $e = (1, \cdots, 1)^T \in R^N$. For convenience, we list these notations in Table 1.

Table 1: Notation Description

Notation	Description
A	data matrix
N	number of training data points
k	number of classes
H_b	precursor of between-class scatter
H_w	precursor of within-class scatter
H_t	precursor of total scatter
S_b	between-class scatter matrix
S_w	within-class scatter matrix
S_t	total scatter matrix or covariance matrix
n	number of dimension
G	transformation matrix
l	number of retained dimensions
A_i	data matrix of the i-th class
m_i	centroid of the i-th class
N_i	number of data points in the i-th class
m	global centroid of the training data set
K	number of nearest neighbors in KNN

2.1. Principal component analysis (PCA)

PCA is a classical feature extraction method widely used in the area of face recognition to reduce the dimensionality. The goal of PCA is to find eigenvectors of the covariance matrix S_t, which correspond to the directions of the principal components of the original data. However, these eigenvectors may eliminate some discriminative information for classification.

Algorithm 1:	PCA
Input :	Data matrix $A \in R^{n\times N}$
Output :	Reduced data matrix A^L
1.	Compute $H_t \in R^{n\times N}$ according to (2);
2.	$S_t \leftarrow H_t H_t^T$;
3.	Compute W from the EVD of S_t: $S_t = W\Sigma W^T$;
4.	Assign the first p columns of W to G, where p=rank(S_t);
5.	$A^L = G^T A$.

2.2. Classical LDA

Classical LDA aims to find the optimal transformation G such that the class structure of the original high-dimensional space is preserved in the low-dimensional space. From (1), we can easily show that $S_t = S_b + S_w$ and see that $\mathrm{trace}(S_b) = \frac{1}{N}\sum_{i=1}^{k} N_i\|m_i - m\|_2^2$ and $\mathrm{trace}(S_w) = \frac{1}{N}\sum_{i=1}^{k}\sum_{x\in A_i}\|x - m_i\|_2^2$ measure the closeness of the vectors within the classes and the separation between the classes, respectively.

In the low-dimensional space resulted from the linear transformation G, the within-class, between-class and total scatter matrices become $S_b^L = G^T S_b G, S_w^L = G^T S_w G$ and $S_t^L = G^T S_t G$, respectively. An optimal transformation G would maximize traceS_b^L and minimize traceS_w^L. Common optimizations in classical LDA include (see [3,20]):

$$\max_G \operatorname{trace}\{(S_w^L)^{-1}S_b^L\} \text{ and } \min_G \operatorname{trace}\{(S_b^L)^{-1}S_w^L\}. \tag{3}$$

The optimization problems in (3) are equivalent to finding the generalized eigenvectors satisfying $S_b x = \lambda S_w x$ with $\lambda \neq 0$. The solution can be obtained by applying the eigen-decomposition to the matrix $S_w^{-1}S_b$ if S_w is nonsingular or $S_b^{-1}S_w$ if S_b is nonsingular. It was shown in [3,20] that the solution can also be obtained by computing the eigen-decomposition on the matrix $S_t^{-1}S_b$ if S_t is nonsingular. There are at most $k-1$ eigenvectors corresponding to nonzero eigenvalues since the rank of the matrix S_b is bounded from above by $k-1$. Therefore, the number of retained dimensions in classical LDA is at most $k-1$. A stable way to compute the eigen-decomposition is to apply SVD on the scatter matrices. Details can be found in [11].

2.3. LDA/range(S_b)

In this subsection, we recall a two-step approach proposed by Yu and Yang [7] to handle small size problems. This method first transforms the original space by using a basis of range(S_b) and then in the transformed space the minimization of within-class scatter is pursued. An optimal transformation G would maximize traceS_b^L and minimize traceS_w^L. Common optimization in LDA/range(S_b) is (see [19]):

$$\min_G \operatorname{trace}\{(S_b^L)^{-1}S_w^L\}. \tag{4}$$

The problem (4) is equivalent to finding the generalized eigenvectors satisfying $S_b x = \lambda S_w x$ with $\lambda \neq 0$. The solution can be obtained by applying the eigen-decomposition to the matrix $S_b^{-1}S_w$ if S_b is nonsingular. If S_b is singular, consider EVD of S_b:

$$S_b = U_b\Sigma_b U_b^T = \begin{bmatrix} U_{b1} & U_{b2} \end{bmatrix} \begin{bmatrix} \Sigma_{b1} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} U_{b1}^T \\ U_{b2}^T \end{bmatrix}, \tag{5}$$

where U_b is orthogonal, $U_{b1} \in R^{n\times q}$, rank(S_b) $= q$ and $\Sigma_{b1} \in R^{q\times q}$ is a diagonal matrix with non-increasing positive diagonal components. We can show that range(S_b)=span(U_{b1}) and $\Sigma_{b1}^{-1/2}U_{b1}^T S_b U_{b1}\Sigma_{b1}^{-1/2} = I_q$. Let $\widetilde{S}_b = \Sigma_{b1}^{-1/2}U_{b1}^T S_b U_{b1}\Sigma_{b1}^{-1/2} = I_q, \widetilde{S}_w = \Sigma_{b1}^{-1/2}U_{b1}^T S_w U_{b1}\Sigma_{b1}^{-1/2}$ and $\widetilde{S}_t = \Sigma_{b1}^{-1/2}U_{b1}^T S_t U_{b1}\Sigma_{b1}^{-1/2}$. Consider the EVD of $\widetilde{S}_w$: $\widetilde{S}_w = \widetilde{U}_w\widetilde{\Sigma}_w\widetilde{U}_w^T$, where $\widetilde{U}_w \in R^{q\times q}$ is orthogonal and $\widetilde{\Sigma}_w \in R^{q\times q}$ is a diagonal matrix. It is evident that

$$\begin{aligned} \widetilde{U}_w^T\Sigma_{b1}^{-1/2}U_{b1}^T S_b U_{b1}\Sigma_{b1}^{-1/2}\widetilde{U}_w = I_q, \\ \widetilde{U}_w^T\Sigma_{b1}^{-1/2}U_{b1}^T S_w U_{b1}\Sigma_{b1}^{-1/2}\widetilde{U}_w = \widetilde{\Sigma}_w. \end{aligned} \tag{6}$$

In most applications, rank(S_w) is greater than rank(S_b) and $\widetilde{\Sigma}_w$ is nonsingular. From (6), it follows that

$$(\widetilde{\Sigma}_w^{-1/2}\widetilde{U}_w^T\Sigma_{b1}^{-1/2}U_{b1}^T)S_b(U_{b1}\Sigma_{b1}^{-1/2}\widetilde{U}_w\widetilde{\Sigma}_w^{-1/2}) = \widetilde{\Sigma}_w^{-1},$$
$$(\widetilde{\Sigma}_w^{-1/2}\widetilde{U}_w^T\Sigma_{b1}^{-1/2}U_{b1}^T)S_w(U_{b1}\Sigma_{b1}^{-1/2}\widetilde{U}_w\widetilde{\Sigma}_w^{-1/2}) = I_q.$$

The optimal transformation matrix proposed in [7] is $G = U_{b1}\Sigma_{b1}^{-1/2}\widetilde{U}_w\widetilde{\Sigma}_w^{-1/2}$ and the algorithm as follows:

Algorithm 2:	LDA/range(S_b)
Input :	Data matrix $A \in R^{n\times N}$
Output :	Reduced data matrix A^L
1.	Compute $H_b \in R^{n\times k}$ and $H_w \in R^{n\times N}$ according to (2);
2.	$S_b \leftarrow H_b H_b^T$, $S_w \leftarrow H_w H_w^T$
3.	Compute the EVD of S_b : $S_b = \begin{bmatrix} U_{b1} & U_{b2} \end{bmatrix} \begin{bmatrix} \Sigma_{b1} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} U_{b1}^T \\ U_{b2}^T \end{bmatrix}$;
4.	Compute the EVD of $\widetilde{S}_w$ $\widetilde{S}_w = \Sigma_{b1}^{-1/2}U_{b1}^T S_w U_{b1}\sum_{b1}^{-1/2}$: $\widetilde{S}_w = \widetilde{U}_w\widetilde{\Sigma}_w\widetilde{U}_w^T$;
5.	Assign $U_{b1}\Sigma_{b1}^{-1/2}\widetilde{U}_w\widetilde{\Sigma}_w^{-1/2}$ to G;
6.	$A^L = G^T A$.

2.4. LDA/GSVD

A recent work on overcoming singularity problem in LDA is the use of generalized singular value decomposition (GSVD) [17,20-21,24-26]. The corresponding algorithm is named LDA/GSVD. It computes the solution exactly because the inversion of the matrix S_w can be avoided. An optimal transformation G obtained by LDA/GSVD would maximize traceS_b^L and minimize traceS_w^L. Common optimization is (see [19,21]):

$$\max_G \operatorname{trace}\{(S_w^L)^{-1}S_b^L\}. \tag{7}$$

The solution of the problem (7) can be obtained by applying the eigen-decomposition to the matrix $S_w^{-1}S_b$ if S_w is nonsingular. If S_w is singular, we have the following efficient algorithm (see [19]):

Algorithm 3:	An efficient method for LDA/GSVD
Input :	Data matrix $A \in R^{n\times N}$
Output :	Reduced data matrix A^L
1.	Compute $H_b \in R^{n\times k}$ and $H_t \in R^{n\times N}$ according to (2);
2.	$S_b \leftarrow H_b H_b^T$, $S_t \leftarrow H_t H_t^T$;

3. Compute the EVD of S_t:
$S_t = \left[\begin{array}{cc} U_{t1} & U_{t2} \end{array}\right] \left[\begin{array}{cc} \Sigma_{t1} & 0 \\ 0 & 0 \end{array}\right] \left[\begin{array}{c} U_{t1}^T \\ U_{t2}^T \end{array}\right]$;
4. Compute the EVD of $\widetilde{S}_b$;
$\widetilde{S}_b = \sum_{t1}^{-1/2} U_{t1}^T S_b U_{t1} \Sigma_{t1}^{-1/2}$:
$\widetilde{S}_b = \widetilde{U}_b \widetilde{\Sigma}_b \widetilde{U}_b^T$;
5. Assign the first $k-1$ columns of $\widetilde{U}_b$ as $\widetilde{U}_{b1}$;
6. $G \leftarrow U_{t1} \sum_{t1}^{-1/2} \widetilde{U}_{b1}$;
7. $A^L = G^T A$.

2.5. LDA/QR

Ye and Li [22-23] presented a novel LDA implementation method, namely LDA/QR. LDA/QR contains two stages, the first stage is to maximize separability between different classes and thus has similar target as OCM [28], the second stage incorporates the within-class scatter information by applying a relaxation scheme to W.

Specifically, we would like to find a projection matrix G such that $G = QW$ for any matrix $W \in R^{k\times k}$. This means that the problem of finding G is equivalent to computing W. Due to $G^T S_b G = W^T(Q^T S_b Q)W$ and $G^T S_w G = W^T(Q^T S_w Q)W$, we have the following optimization problem:

$$W = \arg\max_{W} \operatorname{trace}(W^T \widetilde{S}_b W)^{-1}(W^T \widetilde{S}_w W),$$

where$\widetilde{S}_b = Q^T S_b Q$ and $\widetilde{S}_w = Q^T S_w Q$. An efficient algorithm for LDA/QR can be found in [22-23]:

Algorithm 4:	LDA/QR
Input :	Data matrix $A \in R^{n\times N}$
Output :	Reduced data matrix A^L
1.	Compute $H_b \in R^{n\times k}$ and $H_t \in R^{n\times N}$ according to (2);
2.	Apply QR decomposition to H_b as $H_b = QR$, where $Q \in R^{n\times p}$, $R \in R^{p\times k}$, and p=rank(H_b);
3.	$\widetilde{S}_b \leftarrow RR^T$;
4.	$\widetilde{S}_w \leftarrow Q^T H_w H_w^T Q$;
5.	Compute W from the EVD of $\widetilde{S}_b^{-1}\widetilde{S}_w$ with the corresponding eigenvalues sorted in nondecreasing order;
6.	$G \leftarrow QW$, where $W = [W_1, \cdots, W_q]$ and q=rank(S_w);
7.	$A^L = G^T A$.

3 Improvement of algorithms

3.1. PCA/range(S_b)

In this subsection, we will present a new method, namely PCA/range(S_b), to handle small size problems. This method first transforms the original space by using a basis of range(S_b) and then in the transformed space applies PCA. Similar 2.3, we first consider the EVD (5) of S_b and let $\widetilde{S}_b = \Sigma_{b1}^{-1/2} U_{b1}^T S_b U_{b1} \Sigma_{b1}^{-1/2} = I_q, \widetilde{S}_w = \Sigma_{b1}^{-1/2} U_{b1}^T S_w U_{b1} \Sigma_{b1}^{-1/2}$ and $\widetilde{S}_t = \Sigma_{b1}^{-1/2} U_{b1}^T S_t U_{b1} \Sigma_{b1}^{-1/2}$. Then, we consider the EVD of $\widetilde{S}_t$: $\widetilde{S}_t = \widetilde{U}_t \widetilde{\Sigma}_t \widetilde{U}_t^T$, where $\widetilde{U}_t \in R^{q\times q}$ is orthogonal and $\widetilde{\Sigma}_t \in R^{q\times q}$ is a diagonal matrix, and get an optimal transformation matrix $G = U_{b1}\Sigma_{b1}^{-1/2}\widetilde{U}_t$. An efficient algorithm for PCA/range(S_b) is listed as follows:

Algorithm 5:	PCA/range(S_b)
Input :	Data matrix $A \in R^{n\times N}$
Output :	Reduced data matrix A^L
1.	Compute $H_b \in R^{n\times k}$ and $H_t \in R^{n\times N}$ according to (2), respectively;
2.	$S_b \leftarrow H_b H_b^T$, $S_t \leftarrow H_t H_t^T$;
3.	Compute the EVD of S_b $S_b = \left[\begin{array}{cc} U_{b1} & U_{b2} \end{array}\right] \left[\begin{array}{cc} \Sigma_{b1} & 0 \\ 0 & 0 \end{array}\right] \left[\begin{array}{c} U_{b1}^T \\ U_{b2}^T \end{array}\right]$;
4.	Compute the EVD of $\widetilde{S}_t$ $\widetilde{S}_t = \Sigma_{b1}^{-1/2} U_{b1}^T S_t U_{b1} \Sigma_{b1}^{-1/2}$: $\widetilde{S}_t = \widetilde{U}_t \widetilde{\Sigma}_t \widetilde{U}_t^T$;
5.	$G \leftarrow U_{b1}\Sigma_{b1}^{-1/2}\widetilde{U}_t$;
6.	$A^L = G^T A$.

3.2. PCA/range(S_b)-QR, PCA/QR-1 and PCA/QR-2

In order to improve computationally expensive and low effectively classification, we consider QR decomposition of matrices and introduce an extension version of PCA/range(S_b), namely PCA/range(S_b)-QR. This method is different from PCA/range(S_b) in the second stage. Detailed steps see Algorithm 6:

Algorithm 6:	PCA/range(S_b)-QR
Input :	Data matrix $A \in R^{n\times N}$
Output :	Reduced data matrix A^L
1.	Compute $H_b \in R^{n\times k}$ and $H_t \in R^{n\times N}$ according to (2), respectively;
2.	$S_b \leftarrow H_b H_b^T$, $S_t \leftarrow H_t H_t^T$;
3.	Compute the EVD of S_b $S_b = \left[\begin{array}{cc} U_{b1} & U_{b2} \end{array}\right] \left[\begin{array}{cc} \Sigma_{b1} & 0 \\ 0 & 0 \end{array}\right] \left[\begin{array}{c} U_{b1}^T \\ U_{b2}^T \end{array}\right]$;
4.	Compute the QR decomposition of $\widetilde{H}_t = U_{b1}^T H_t$: $\widetilde{H}_t = \widetilde{Q}_t \widetilde{R}_t$;
5.	$G \leftarrow U_{b1}\widetilde{Q}_t$;
6.	$A^L = G^T A$.

If the EVD of S_t in Algorithm 1 is replaced by the QR decomposition of H_t, we can obtain two improve versions of PCA, namely PCA/QR-1 and PCA/QR-2.

Algorithm 7:	PCA/QR-1
Input :	Data matrix $A \in R^{n\times N}$
Output :	Reduced data matrix A^L
1.	Compute $H_t \in R^{n\times N}$ according to (2);
2.	Compute W from the QR decomposition of H_t;
3.	Assign the first p columns of W to G where p=rank(S_t);
4.	$A^L = G^T A$.

Algorithm 8:	PCA/QR-2
Input :	Data matrix $A \in R^{n\times N}$
Output :	Reduced data matrix A^L
1.	Compute $H_t \in R^{n\times N}$ according to (2);
2.	Compute W from the QR decomposition of H_t;
3.	Assign the first q columns of W to G where q=rank(S_b);
4.	$A^L = G^T A$.

3.3. LDA/range(S_b)-QR-1, LDA/range(S_b)-QR-2, and LDA/GSVD-QR

Let $\widetilde{H}_b = \Sigma_{b1}^{-1/2} U_{b1}^T H_b \in R^{q\times k}$, where Σ_{b1} and U_{b1} are same as in (5), and consider the QR decomposition of $\widetilde{H}_b$: $\widetilde{H}_b = \widetilde{Q}_b \widetilde{R}_b$. If the EVD of $\widetilde{S}_t$ in Algorithm 5 is replaced by the QR decomposition of $\widetilde{H}_b$, we can get an improve version of LDA/range(S_b), namely LDA/range(S_b)-QR-1.

Algorithm 9:	LDA/range(S_b)-QR-1
Input :	Data matrix $A \in R^{n\times N}$
Output :	Reduced data matrix A^L
1.	Compute $H_b \in R^{n\times k}$ according to (2);
2.	$S_b \leftarrow H_b H_b^T$;
3.	Compute the EVD of S_b $S_b = \left[\begin{array}{cc} U_{b1} & U_{b2}\end{array}\right] \left[\begin{array}{cc}\Sigma_{b1} & 0\\ 0 & 0\end{array}\right]\left[\begin{array}{c} U_{b1}^T \\ U_{b2}^T\end{array}\right]$;
4.	Compute $\widetilde{Q}_b$ from the QR decomposition of $\widetilde{H}_b = \sum_{b1}^{-1/2} U_{b1}^T H_b$: $\widetilde{H}_b = \widetilde{Q}_b\widetilde{R}_b$;
5.	$G \leftarrow U_{b1} \sum_{b1}^{-1/2} \widetilde{Q}_b$;
6.	$A^L = G^T A$.

Another improve version of LDA/range(S_b), namely LDA/range(S_b)-QR-2, is a two-step method. Firstly, consider the QR decomposition of H_b : $H_b = Q_b R_b$ and let

$$\widetilde{S}_w = Q_b^T S_w Q_b = Q_b^T H_w H_w^T Q_b = \widetilde{H}_w \widetilde{H}_w^T,$$
$$\widetilde{S}_b = Q_b^T S_b Q_b = Q_b^T H_b H_b^T Q_b = R_b R_b^T,$$

where $\widetilde{H}_w = Q_b^T H_w$, and then consider the QR decomposition of $\widetilde{H}_w$: $\widetilde{H}_w = \widetilde{Q}_w \widetilde{R}_w$. We can obtain the transformation matrix $G = Q_b \widetilde{Q}_w$, that is,

Algorithm 10:	LDA/range(S_b)-QR-2
Input :	Data matrix $A \in R^{n\times N}$
Output :	Reduced data matrix A^L
1.	Compute $H_b \in R^{n\times k}$ and $H_w \in R^{n\times N}$ according to (2) respectively;
2.	Compute the QR decomposition of H_b $H_b = Q_b R_b$, where $Q_b \in R^{n\times q}$, $R_b \in R^{q\times k}$ and $q = \mathrm{rank} H_b$;
3.	Compute the QR decomposition of $\widetilde{H}_w = Q_b^T H_w$: $\widetilde{H}_w = \widetilde{Q}_w \widetilde{R}_w$, where $\widetilde{Q}_w \in R^{q\times q}$ and $\widetilde{R}_w \in R^{q\times N}$;
4.	$G \leftarrow Q_b \widetilde{Q}_w$;
5.	$A^L = G^T A$.

Next, we apply the QR decomposition of matrices to improve Algorithm 3 and get an improve version of LDA/GSVD, namely LDA/GSVD-QR. Firstly, consider the QR decomposition of H_t : $H_t = Q_t R_t$ and let $\widetilde{S}_b = Q_t^T S_b Q_t = (Q_t^T H_b)(Q_t^T H_b)^T = \widetilde{H}_b \widetilde{H}_b^T$, where $\widetilde{H}_b = Q_t^T H_b$. Then, consider the QR decomposition of $\widetilde{H}_b$: $\widetilde{H}_b = \widetilde{Q}_b \widetilde{R}_b$ and obtain the transformation matrix $G = Q_t \widetilde{Q}_b$. Detail steps are listed in Algorithm 11.

Algorithm 11:	LDA/GSVD-QR
Input :	Data matrix $A \in R^{n\times N}$
Output :	Reduced data matrix A^L
1.	Compute $H_b \in R^{n\times k}$ and $H_t \in R^{n\times N}$ according to (2), respectively;
2.	Compute the QR decomposition of H_t $H_t = Q_t R_t$;
3.	Compute the QR decomposition of $\widetilde{H}_b = Q_t^T H_b$: $\widetilde{H}_b = \widetilde{Q}_b \widetilde{R}_b$;
4.	$G \leftarrow Q_t \widetilde{Q}_b$;
5.	$A^L = G^T A$.

4 Experiments and analysis

In this section, in order to show the classification effectiveness of the proposed methods, we make a series of experiments with 6 different data sets taken from ORL faces database [29-30] and Yale faces database [18,29,31]. Data sets 1-3 are taken from ORL face database, which consists of 400 images of 40 different people with 10304 attributes for each image and has 40 classes. Data set ORL1 chooses from 222-th to 799-th dimension, ORL2 from 1901-th to 2400-th dimension and ORL3 from 2301-th to 2800-th dimension. Data sets 4-6 come from Yale database, which contains 165 face images of 15 individuals with 1024 attributes for each image and has 10 classes. We randomly take 150 images in Yale database with 10 images for each person to make up a database, in which data set Yale1 chooses from 1-th to 300-th dimension, Yale2 from 301-th to 600-th dimension and Yale3 from 501-th to 800-th dimension. All data sets are split to a train set and a test set with the ratio 4:1. Experiments are repeated 5 times to obtain mean prediction error rate. The K-nearest-neighbor algorithm (KNN) with K=3,4,5,6,7 is used as a classifier for all date sets. All experiments are performed on a PC (2.40GHZ CPU, 2G RAM) with MATLAB 7.1.

4.1. Comparison PCA/range(S_b) and PCA/range(S_b)-QR with PCA and LDA/range(S_b)

In this subsection, in order to demonstrate the effectiveness of PCA/range(S_b) and PCA/range(S_b)-QR, we compare them with PCA and LDA/range(S_b) on the six data sets. The experiment results are listed in Table 2: For convenience, this paper call LDA/range(S_b), LDA/range(S_b)-QR-1, LDA/range(S_b)-QR-2, PCA/range(S_b), PCA/range(S_b)-QR, PCA/QR-1, PCA/QR-2, LDA/GSVD and LDA/GSVD-QR methods LRSb, LRSbQ1, LRSbQ2, PRSb, PRSbQ, PQ1, PQ2, LGD and LGSQ for short, respectively.

Table 2: The error rate of PCA, PCA/range(S_b), PCA/range(S_b)-QR and LDA/range(S_b)

Data set	K	PCA	PRSb	PRSbQ	LRSb
ORL1	3	97.25	34.00	23.25	43.50
	4	97.00	35.75	24.00	45.25
	5	97.25	39.75	24.75	46.75
	6	97.00	40.50	26.00	48.25
	7	97.25	41.50	30.00	48.75
ORL2	3	94.75	35.00	20.25	39.00
	4	94.75	38.50	21.00	41.75
	5	94.75	38.75	24.00	45.50
	6	95.00	39.75	25.00	47.75
	7	96.00	42.75	25.50	48.75
ORL3	3	96.50	31.50	18.25	40.00
	4	97.00	36.50	20.25	42.00
	5	97.00	40.00	23.50	46.00
	6	97.75	43.00	24.50	49.25
	7	98.25	43.00	26.50	49.25
Yale1	3	95.30	27.30	32.00	28.70
	4	95.30	29.30	32.00	25.30
	5	96.00	28.70	34.70	27.30
	6	96.00	29.30	31.30	30.00
	7	96.00	31.30	32.00	29.30
Yale2	3	94.70	30.70	27.30	31.30
	4	93.30	31.30	29.30	31.30
	5	93.30	32.70	28.70	31.30
	6	93.30	33.30	34.00	31.30
	7	95.30	35.30	34.00	34.70
Yale3	3	90.70	36.00	38.70	35.30
	4	90.70	33.30	34.00	36.00
	5	90.70	33.30	33.30	34.00
	6	90.70	33.30	36.70	38.70
	7	94.00	37.30	38.00	42.00

From Table 2, we can see that PCA/range(S_b) and PCA/range(S_b)-QR are generally better than PCA and LDA/range(S_b) for classification accuracy. For data sets ORL1-ORL3 and Yale2 PCA/range(S_b)-QR is obviously better than PCA/range(S_b) and for data sets Yale1 and Yale3 PCA/range(S_b) is better than PCA/range(S_b)-QR.

4.2. Comparison LDA/GSVD-QR, PCA/QR-1 and PCA/QR-2 with LDA/GSVD and LDA/QR

The experiments in this subsection can demonstrate the effectiveness of LDA/GSVD-QR, PCA/QR-1 and PCA/QR-2 by comparing them with LDA/GSVD and LDA/QR. The experiment results can be found in Table 3:

Table 3: The error rate of LDA/GSVD, LDA/QR, LDA/GSVD-QR, PCA/QR-1 and PCA/QR-2

Data set	K	LGD	LGSQ	PQ1	PQ2	LDA/QR
ORL1	3	95.00	23.25	23.00	22.25	33.50
	4	93.75	24.00	22.50	22.75	34.25
	5	92.75	24.75	26.25	23.75	36.75
	6	93.75	26.00	27.75	25.25	38.25
	7	94.75	30.00	28.25	27.00	39.75
ORL2	3	94.50	20.25	20.00	19.00	34.50
	4	95.00	21.00	20.75	20.25	36.00
	5	94.75	24.00	24.00	23.25	39.25
	6	94.25	25.00	24.75	23.25	41.00
	7	94.75	25.50	28.75	25.25	42.00
ORL3	3	92.00	18.25	19.50	20.50	31.50
	4	93.50	20.25	21.25	21.25	33.50
	5	94.00	23.50	23.75	24.00	36.75
	6	93.00	24.50	25.00	24.25	40.00
	7	92.75	26.50	26.75	26.00	41.50
Yale1	3	90.00	32.00	31.30	31.30	24.70
	4	90.70	32.00	31.30	31.30	24.70
	5	88.70	34.70	32.70	33.30	28.00
	6	90.00	31.30	31.30	36.70	27.30
	7	90.70	32.00	29.30	33.30	28.00
Yale2	3	85.30	27.30	23.30	32.00	30.70
	4	84.70	29.30	22.70	33.30	31.30
	5	86.00	28.70	28.00	32.00	34.00
	6	84.70	34.00	26.00	33.30	33.30
	7	82.00	34.00	30.00	34.70	34.00
Yale3	3	87.30	38.70	31.30	39.30	32.70
	4	86.70	34.00	30.00	39.30	31.30
	5	85.30	33.30	32.00	39.30	32.00
	6	84.00	36.70	28.70	38.00	34.00
	7	86.00	38.00	31.30	36.70	34.00

From Table 3, we can see that LDA/GSVD-QR, PCA/QR-1 and PCA/QR-2 are generally better than LDA/GSVD and LDA/QR for classification accuracy. For data sets ORL1 and ORL2 PCA/QR-2 is better than LDA/GSVD-QR and PCA/QR-1 and for data sets Yale1-Yale3 PCA/QR-1 is better than LDA/GSVD-QR and PCA/QR-2. For Yale1 LDA/QR is the best in five algorithms and for Yale3 PCA/QR-1 is the best.

4.3. Comparison LDA/range(S_b)-QR-1 and LDA/range(S_b)-QR-2 with LDA/range(S_b)

In this subsection, in order to demonstrate the effectiveness of LDA/range(S_b)-QR-1 and LDA/range(S_b)-QR-2, we compare them with LDA/range(S_b). The experiment results are as follows:

Table 4: The error rate of LDA/range(S_b), LDA/range(S_b)-QR-1 and LDA/range(S_b)-QR-2

Data set	K	LRSb	LRSbQ1	LRSbQ2
ORL1	3	43.50	34.00	23.25
	4	45.25	35.75	24.00
	5	46.75	39.75	24.75
	6	48.25	40.50	26.00
	7	48.75	41.50	30.00
ORL2	3	39.00	35.00	20.25
	4	41.75	38.50	21.00
	5	45.50	38.75	24.00
	6	47.75	39.75	25.00
	7	48.75	42.75	25.50
ORL3	3	40.00	31.50	18.25
	4	42.00	36.50	20.25
	5	46.00	40.00	23.50
	6	49.25	43.00	24.50
	7	49.25	43.00	26.50
Yale1	3	28.70	27.30	32.00
	4	25.30	29.30	32.00
	5	27.30	28.70	34.70
	6	30.00	29.30	31.30
	7	29.30	31.30	32.00
Yale2	3	31.30	30.70	27.30
	4	31.30	31.30	29.30
	5	31.30	32.70	28.70
	6	31.30	33.30	34.00
	7	34.70	35.30	34.00
Yale3	3	35.30	36.00	38.70
	4	36.00	33.30	34.00
	5	34.00	33.30	33.30
	6	38.70	33.30	36.70
	7	42.00	37.30	38.00

From Table 4, we can see that for data sets ORL1-ORL3 LDA/range(S_b)-QR-1 is obviously better than LDA/range(S_b) and LDA/range(S_b)-QR-2 is obviously better than LDA/range(S_b)-QR-1. LDA/range(S_b)-QR-2 is the worst in three algorithms for Yale1, but the best for Yale2 with K=3,4,5,7. For Yale3 LDA/range(S_b)-QR-1 is better than LDA/range(S_b) and LDA/range(S_b)-QR-2 for k=4,5,6,7.

4.4. Visual comparison of different approaches

In this subsection, In order to illustrate the effectiveness of the QR decomposition outperforms previous approaches on classification accuracy rate, we further compare the improvement methods on the ORL and Yale database with different K values. The experiment results are shown from Fig.1 to Fig.4.

Fig.1 and Fig.2 are show the deformed version of PCA, Fig.3 and Fig.4 are show the deformed version of LDA.

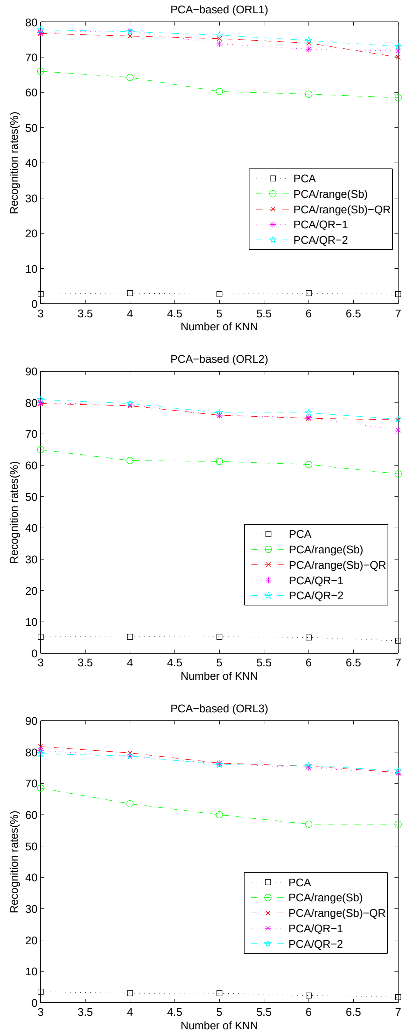


Fig.1 Recognition rates of PCA, PCA/range(Sb), PCA/range(Sb)-QR, PCA/QR-1 and PCA/QR-2 under ORL.

From Fig.1 to Fig.4 we clear to see that QR decomposition play an important role in classification accuracy rate.

5 Conclusion

In this paper, we introduce a new algorithm PCA/range(S_b) and five improve versions of three

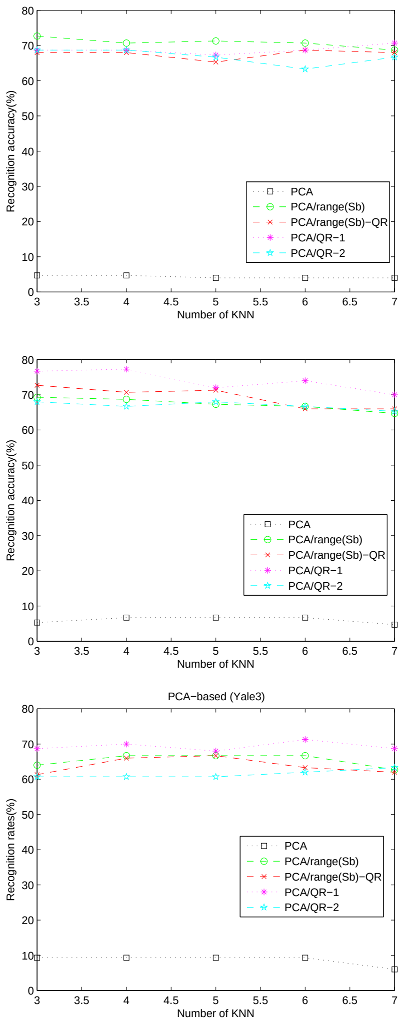
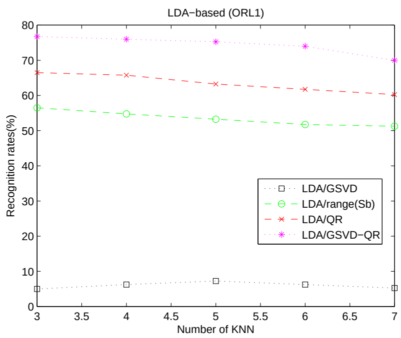


Fig.2 Recognition rates of PCA, PCA/range(Sb), PCA/range(Sb)-QR, PCA/QR-1 and PCA/QR-2 under Yale.



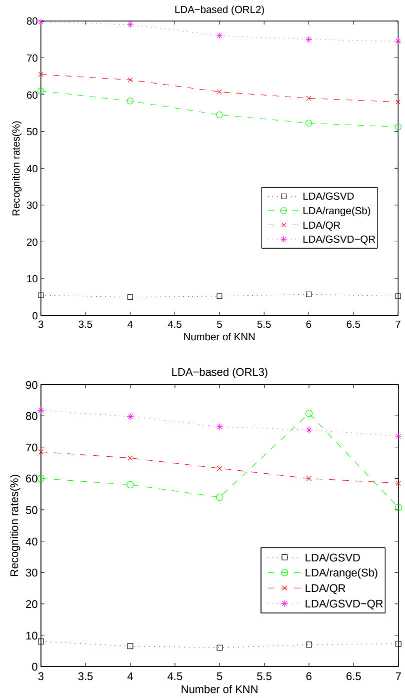
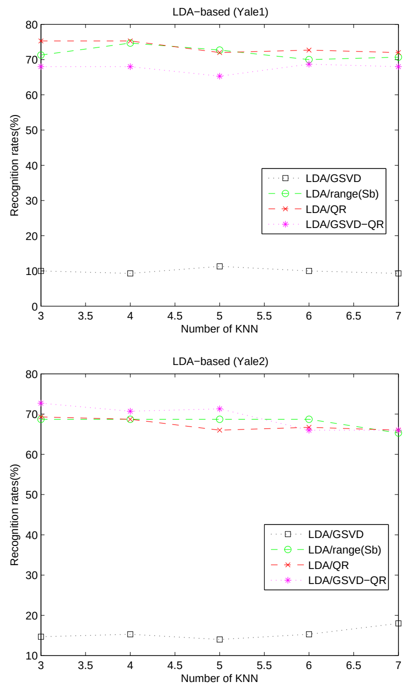


Fig.3 Recognition rates of LDA/GSVD, LDA/range(Sb), LDA/QR and LDA/GSVD-QR under ORL.



known algorithms and PCA/range(S_b) by means of QR decomposition for small size problems. In order to explain the classification effectiveness of presented methods, we perform a series of experiments with six data sets taken from ORL and Yale faces databases. The experiment results show that the presented algorithms are better than PCA, LDA/range(S_b), LDA/GSVD and LDA/QR in generally. From Tables 2-4, we can see that PCA products very poor classification result, but PCA/QR-1 and PCA/QR-2 are much better than PCA for the six data sets, which illustrates that QR decomposition can really improve classification accuracy for discriminant analysis. In 11 algorithms mentioned in this paper, PCA and LDA/GSVD are the worst algorithms, and PCA/QR-2

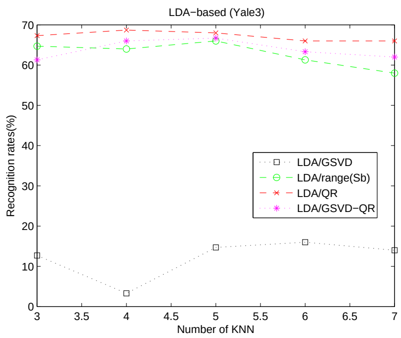


Fig.4 Recognition rates of LDA/GSVD, LDA/range(Sb), LDA/QR and LDA/GSVD-QR under Yale.

and LDA/range(S_b)-QR-2 are more better than others for classification accuracy. Visual comparison of different approaches show the effective of QR decomposition.

Acknowledgements: The research is supported by NSF (grant No.10871226), NSF (grant No. ZR2009AL006) of Shandong Province and Young and Middle-Aged Scientists Research Foundation (grant No. BS2010SF004) of Shandong Province, P.R. China. Corresponding author: Liya Fan, fanliya63@126.com.

References:

[1] R. Bellmanna, Adaptive control processes: a guided tour, *princeton university press*, 1961.

[2] R. Duda, P. Hart, D. Stork, pattern classification, second ed., wiley, *New York*, 2000.

[3] K. Fukunaga, Introduction to statistical pattern classification academic press, *San Diego, California, USA*, 1990.

[4] T. Hastie, R. Tibshirani, J.H. Friedman, The elements of statistical learning: *data mining, inference, and prediction, Springer*, 2001.

[5] I.T Jolliffe, Principal Component Analysis, Second ed., *Springer-Verlag, New York*, 2002.

[6] I.T. Jolliffe, Principal component analysis, *Springer-Verlag, New York*, 1986.

[7] H. Yu, J. yang, A direct LDA algorithm for high-dimensional data with application to face recognition, *pattern recognition*, 34, 2001, pp. 2067–2070.

[8] A. Basilevsky, Statistical factor analysis and related methods, theory and applications, *Wiley, New York*, 1994.

[9] C. Chatfield, Introduction to multivariate analysis, chapman and hall, *london, New York*, 1980.

[10] A.C. Rencher, Methods of multivariate analysis, *Wiley, New York*, 1995.

[11] D.L. Swets, J. Weng, Using discriminant eigenfeatures for image retrieval, *IEEE Trans Pattern Anal Mach Intell*,18, 1996, (8), pp. 831–836.

[12] M. Loog, R.P.W. Duin, R. Hacb-Umbach, Multiclass linear dimensuin reduction by weighted pairwise fisher criteria, *pattern recognition and machine intelligence*, 23, 2001, pp. 762–766.

[13] A. Jain, D. Zongker, Feature slection: application and small sample performance, *IEEE transations on pattern analysis and machine intelligence*, 19, 1997, (2), pp. 153–158.

[14] H.M. Lec, C.M. Chen, Y.L. Jou, An efficient fuzzy classifier with feature selection based on fuzzy entropy, *IEEE transations on systems, man, and cybernetics B*, 31, 2001, (3), pp. 426–432.

[15] N.R. Pal, V.K. Eluri, Two efficient connectionist schemes for structure preserving dimensuinality reduction, *IEEE transations on neural networks*, 9, 1998, (6), pp. 1142–1154.

[16] W.J. Krzanowski, P. Jonathan, W.V. McCarthy, M.R. Thomas, Discriminant analysis with singular covariance matrices: methods and applications to speetroscopic data, *applied statistic*, 44, 1995, pp. 101–115.

[17] J. Ye, R. Janardan, C.H. Park, and H. Park, An optimization criterion for generalized discriminant analysis on undersampled problem, *IEEE trans,pattern analysis and machine intelligence*, 26, 2004, (8), pp. 982–994.

[18] P. Belhumeur, J.P. Hespanha, D.J. Kriegman, Eigenfaces va. fisherface recognition using class specific linear projection, *pattern recognition and machine intelligence*, 19, 1997, (7), pp. 711–720.

[19] C.H. Park, H. Park, A comparison of generalized linear discriminant analysis algorithms, *Pattern Recognition*, 41, 2008, pp. 1083–1097.

[20] P. Howland, M. Jeon, H. Park, structure preserving dimension reduction for clustered text data based on the generalized singular value decomposition, *SIAM J. matrix analysis and applications*, 25, 2003, (1), pp. 165–179.

[21] P. Howland, H. Park, Generalizing discriminant analysis using the generalized singular value decomposition, *IEEE trans., pattern anal., mach. intell.*, 26, 2004, (8), pp. 995–1006.

[22] J. Ye, Q. Li, LDA/QR: An efficient and effective dimension reduction algorithm and its theoretical foundation, *pattern recognition*, 37, 2004, pp. 851–854.

[23] J. Ye and Q. Li, A two-stage linear discriminant analysis via QR-decomposition, *IEEE transaction on pattern analysis and machine intelligence*, 27, 2005, (6), pp. 929–941.

[24] C.F. Vanloan, Generalizing the singular value decomposition,*SIAM J. numerical analysis*, 13, 1976, pp. 76–83.

[25] G.H. Golub, C.F. Vanloan, Matrix computations, third ed. *the johns hopkins univ. press*, 1996.

[26] C.C. Paige, M.A. Saunders, Towards a generalized singular value decomposotion, *SIAM J. numer anal.*, 18, 1981, (3), pp. 398–405.

[27] P. Howland, H. Park, Equivalence of several two-stage methods for linear discriminant analysis, in: *proceedings of the fourth SIAM international conference on data mining*, 4, 2004, pp. 69–77.

[28] H. Park, M. Jeon, J. Rosen, lower dimensional representation of text data based on centroids and least spuares, *BIT*, 43, 2003, (2) pp. 1–22.

[29] L.M. Zhang, L.S. Qiao, S.C. Chen, Graph-optimized locality preserving projections, *pattern recognition*, 43, 2010, (6), pp. 1993–2002.

[30] ORL database <http://www.uk.research.att.com/facedatabase.html>, 2002.

[31] Yale database <http://cvc.yale.edu/projects/yalefaces/yalefaces.html>, 2002.

[32] JianQiang Gao, Liya Fan, Kernel-based weighted Discriminant Analysis with QR Decom- position and Its Application Face Recognition. *WSEAS Transactions on Mathematics*,10, 2011, (10), pp. 358–367.