ORTHOGONAL DISCRIMINANT NEIGHBORHOOD PRESERVING EMBEDDING FOR FACIAL EXPRESSION RECOGNITION

Shuai Liu, Qiuqi Ruan, Rongrong Ni

Institute of Information Science, Beijing Jiaotong University, 100044 Beijing, P. R. China

ABSTRACT

In this paper, a new manifold learning algorithm called Orthogonal Discriminant Neighborhood Preserving Embedding (ODNPE) is proposed for facial expression recognition. The ODNPE pursues orthogonal projections vectors to preserve the local manifold within same classes and keep the separability between different classes. The obtained orthogonal projections vectors can keep the metric structure of the manifold embedded in high dimensional space such that the intrinsic dimensions of the manifold can be well learned. Furthermore, we design a novel penalty graph to describe the separability between pair-wise different classes. The proposed algorithm is compared with some other algorithms on two facial expression databases, and the experimental results show its effectivity.

Index Terms— facial expression recognition, dimensionality reduction, manifold learning, orthogonal discriminant neighborhood preserving embedding (ODNPE)

1. INTRODUCTION

Dimensionality reduction is a key problem in many fields such as computer vision and pattern recognition. There are two kinds of algorithms: the linear and the nonlinear. The linear algorithms mainly include principal component analysis (PCA) [3] and linear discriminant analysis (LDA) [5]. PCA finds the projection directions that maximize the global variations. LDA minimizes the intraclass scatter and meanwhile maximizes the interclass scatter. However, linear algorithms only discover the linear structure in Euclidean space. The manifold learning algorithms, which aim to find meaningful low dimensional manifold structure embedded in the high dimensional space, effectively solve the nonlinear problem. The popular manifold learning algorithms include Local Linear Embedding (LLE) [1] with its linearization version Neighborhood Preserving Embedding (NPE) [9], and Laplacian Eigenmaps [7] with its linearization version Locality Preserving Projections (LPP) [6]. Recently, as a variation of LLE, the Discriminant Locality Linear Embedding (DLLE) is proposed [8], which incorporates a penalty graph into LLE according to the graph embedding framework [2], such that it can preserve the local properties described by LLE and meanwhile separate nearby points among different classes.

However, the projection vectors obtained by the linearized DLLE (DLLE/L) are not orthogonal; therefore, it can't preserve the metric structure of the high-dimensional space and may suffer from the problem of dimensionality estimation [4]. On the other hand, the penalty graph adopted in DLLE only describes the separability between one class and all the other classes rather than the pair-wise different classes. Such a penalty graph may not sufficiently characterize the interclass separability for multi-classes.

In this paper, motivated by the success of Orthogonal Laplacianfaces (OLPP) [4] and based on DLLE\L, we propose the Orthogonal Discriminant Neighborhood Preserving Embedding (ODNPE), which finds orthogonal projection vectors to better preserve the local intraclass manifold and designs a novel penalty graph to describe the separability between pair-wise different classes. The experimental results on two facial expression databases show the effectivity of our algorithm for facial expression recognition.

In this paper, matrices are denoted by italic uppercase symbols such as *X*, *Y*, *W*; vectors are denoted by italic bold lowercase symbols such as ***x***, ***y***, ***w***; scalars are denoted by normal symbols such as i, h, α, N.

2. REVIEW OF DLLE

Given input sample set $X=[\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_{\mathrm{N}}]$, $\boldsymbol{x}_{\mathrm{i}}\in\mathbb{R}^{\mathrm{M}}$, the dimensionality reduction algorithms aim to find low-dimensional vectors $\{\boldsymbol{y}_1,\boldsymbol{y}_2,\cdots,\boldsymbol{y}_{\mathrm{N}}\}$, $\boldsymbol{y}_{\mathrm{i}}\in\mathbb{R}^{\mathrm{L}}$ (L<M) , satisfying some statistic or topological properties, to effectively represent X . The graph embedding framework [2] unifies most of the popular dimensionality reduction algorithms (including but not limited to [1][3][5][6][7][9]) into the following objective function:

$$Y=\underset{Tr(YBY^T)=\mathrm{c}}{\arg\min}\sum\nolimits_{\mathrm{i}\neq\mathrm{j}}\left\|\boldsymbol{y}_{\mathrm{i}}-\boldsymbol{y}_{\mathrm{j}}\right\|^2 S_{\mathrm{i,j}}\ , \qquad (1)$$

$$=\arg\min Tr(YLY^T)/Tr(YBY^T)$$

, where $Y=[\boldsymbol{y}_1,\boldsymbol{y}_2,\cdots,\boldsymbol{y}_N]$; $Tr(\bullet)$ denotes the trace of a matrix; S is the similarity matrix of an affinity graph G which describes the similarities to be preserved among the input samples; $L=D-S$ (D is a diagonal matrix with $D_{\mathrm{ii}}=\sum\nolimits_{\mathrm{j}}S_{\mathrm{ij}}$) is the laplacian matrix of *G*; *B* may be simply

978-1-4244-7994-8/10/$26.00 ©2010 IEEE

a diagonal matrix for scale normalization or the laplacian matrix of a penalty graph G' which describes the unwanted similarities among input samples that should be suppressed; c is a constant.

DLLE is proposed based on (1). DLLE constructs the similarity matrix S of G by:

$$S_{\mathrm{ij}} = (M + M^T - M^T M)_{\mathrm{ij}}, \mathrm{i} \neq \mathrm{j}; 0, \mathit{otherwise}\,. \tag{2}$$

M in (2) is the coefficient matrix of LLE [1] obtained by:

$$\min \sum\nolimits_{\mathrm{j} \in \aleph_{\mathrm{k}_1}(\mathrm{i})} \left\| \boldsymbol{x}_{\mathrm{i}} - M_{\mathrm{ij}} \boldsymbol{x}_{\mathrm{j}} \right\|^2, \sum\nolimits_{\mathrm{j}} M_{\mathrm{ij}} = 1 \; \forall \mathrm{i}\,, \tag{3}$$

, where $\aleph_{\mathrm{k}_1}(\mathrm{i})$ means the k_1 nearest neighbors of $\boldsymbol{x}_{\mathrm{i}}$.

DLLE gives B in (1) as the laplacian matrix of G', i.e., $B = L^p = D^p - S^p$, D^p is a diagonal matrix with $D^p_{\mathrm{ii}} = \sum\nolimits_{\mathrm{j}} S^p_{\mathrm{ij}}$, and S^p is the similarity matrix of G' constructed by:

$$S^p_{\mathrm{ij}} = 1/\mathrm{k}_2, \mathit{if}\ \mathrm{j} \in \aleph^{l_{\mathrm{i}}}_{\mathrm{k}_2}\ \mathit{or}\ \mathrm{i} \in \aleph^{l_{\mathrm{j}}}_{k_2} (l_{\mathrm{i}} \neq l_{\mathrm{j}}); 0, \mathit{otherwise} \tag{4}$$

, where l_{i} is the class label of $\boldsymbol{x}_{\mathrm{i}}$, and $\aleph^{l_{\mathrm{i}}}_{\mathrm{k}_2}$ means the first k_2 nearest neighbors between class l_{i} and other classes. Finally, DLLE optimizes the following trace difference form objective function (5) which is adapted from (1):

$$\begin{aligned} Y^* &= \arg\max(\sum\nolimits_{\mathrm{i} \neq \mathrm{j}} \left\| (y_{\mathrm{i}} - y_{\mathrm{j}}) \right\|^2 S^p_{\mathrm{ij}} - \mathrm{h} \sum\nolimits_{\mathrm{i} \neq \mathrm{j}} \left\| (y_{\mathrm{i}} - y_{\mathrm{j}}) \right\|^2 S_{\mathrm{ij}}) \\ &= \arg\max_{Y} (Tr(YL^pY^T) - \mathrm{h}Tr(YLY^T)) \end{aligned} \tag{5}$$

, where $Tr(YL^pY^T)$ reflects the interclass separability, while $Tr(YLY^T)$ reflect the intraclass manifold structure depicted by LLE; the parameter h is used to balance these two terms. The objective function (5) endeavors to shorten the distance between the connected points in G and to enlarge the distance between the connected points in G'.

DLLE/L is the linearization of DLLE, which computes the low-dimensional representation for the samples out of X though linear projection. For linearization, the original high-dimensional samples are usually first projected into the PCA subspace by:

$$\tilde{X} = W^T_{PCA} X\,, \tag{6}$$

where W_{PCA} is the projection matrix of PCA. Then DLLE/L gets its linear projection matrix W^* by optimizing the objective function:

$$W^* = \arg\max_{W} Tr(W^T \tilde{X} L^p \tilde{X}^T W - \mathrm{h} W^T \tilde{X} L \tilde{X}^T W)\,. \tag{7}$$

The overall projection matrix is $W_o = W_{PCA} W^*$, and for any input sample $\boldsymbol{x}^*$, DLLE/L gets the low- dimensional representation as $\boldsymbol{y}^* = W^T_o \boldsymbol{x}^*$.

3. OUR ODNPE ALGORITHM

In this section, we propose our ODNPE algorithm, which improves the penalty graph of DLLE, and finds the orthogonal projection vectors for DLLE\L.

3.1. A novel penalty graph

First, we observe the penalty graph G' of DLLE defined in (4) can't ensure the marginal points between all pairs of different classes are connected. In the case of multi-classification, there may exist two classes with no edge connected between them in G', thus the nearby marginal points between them may not be well separated. To solve this problem, we design a novel penalty graph $\tilde{G}'$ in this way: for each pair of different classes, connect the nearest k pairs of marginal points between them. We denote the similarity matrix of $\tilde{G}'$ by $\tilde{S}^p$ which is defined as:

$$\tilde{S}^p_{\mathrm{ij}} = 1/\mathrm{k}, \mathrm{i} \in l, \mathrm{j} \in l', l \neq l', \mathit{and}(\mathrm{i},\mathrm{j}) \in \aleph^{ll'}_{\mathrm{k}}; 0, \mathit{otherwise}\,, \tag{8}$$

where l and l' represent any two different classes, and $\aleph^{ll'}_{\mathrm{k}}$ means the first k nearest pairs of marginal points between class l and class l'.

3.2. ODNPE

The orthogonal projection vectors preserve the metric structure of the original space, and are testified to have more locality preserving power and more discriminating power than the non-orthogonal ones [4]. Motivated by that, we propose the ODNPE, which finds orthogonal projection vectors for DLLE/L.

The objective function of our ODNPE is as follows:

$$W^* = \arg\max_{W} Tr(W^T X \tilde{L}^p X^T W) / Tr(W^T XLX^T W) \tag{9}$$

s.t. $\boldsymbol{w}^T_{\mathrm{p}} XLX^T \boldsymbol{w}_{\mathrm{p}} = 1$, $\boldsymbol{w}^T_{\mathrm{p}} \boldsymbol{w}_{\mathrm{q}} = 0$ $(\mathrm{p},\mathrm{q} = 1 \cdots \mathrm{n}, \mathrm{p} \neq \mathrm{q})$

where $W = [\boldsymbol{w}_1, \boldsymbol{w}_2, \cdots, \boldsymbol{w}_{\mathrm{n}}]$ is the projection matrix with the columns as the projection vectors, $\tilde{L}^p$ is the laplacian matrix of the novel penalty graph $\tilde{G}'$ defined by (8), and L is the laplacian matrix of the affinity graph G defined by (2). There is no rule to select the balancing parameter for the trace difference form objective function as (7), so here in (9) we still adopt the trace ratio form as (1). The constraints $\boldsymbol{w}^T_{\mathrm{p}} \boldsymbol{w}_{\mathrm{q}} = 0$ $(\mathrm{p},\mathrm{q} = 1 \cdots \mathrm{n}, \mathrm{p} \neq \mathrm{q})$ ensure the projection vectors are orthogonal; and the constraints $\boldsymbol{w}^T_{\mathrm{p}} XLX^T \boldsymbol{w}_{\mathrm{p}} = 1$ $(\mathrm{p},\mathrm{q} = 1 \cdots \mathrm{n}, \mathrm{p} \neq \mathrm{q})$ normalize the scale of the projection vectors. To avoid the singularity problem, if the dimensionality of the original sample space is too high vis-a-vis the number of the training samples, we first substitute X with the PCA projections by (6), thus we can ensure that XLX^T is positive definite. We use the method adopted in [4] to resolve (9). First, for p=1, there is no orthogonal constraint to $\boldsymbol{w}_1$, so $\boldsymbol{w}_1$ can be computed as the generalized eigenvector of $X\tilde{L}^p X^T$ and XLX^T corresponding to the largest eigenvalue. Assume $\boldsymbol{w}_1, \boldsymbol{w}_2, \cdots, \boldsymbol{w}_{\mathrm{p}-1}$ (p>1) are fixed, then $\boldsymbol{w}_{\mathrm{p}}$ can be obtained by resolving the following optimization problem:

$$\boldsymbol{w}_{\mathrm{p}} = \arg\max_{\boldsymbol{w}_{\mathrm{p}}} (\boldsymbol{w}^T_{\mathrm{p}} X \tilde{L}^p X^T \boldsymbol{w}_{\mathrm{p}}) / (\boldsymbol{w}^T_{\mathrm{p}} XLX^T \boldsymbol{w}_{\mathrm{p}}) \tag{10}$$

s.t. $\boldsymbol{w}_{\mathrm{p}}^T XLX^T \boldsymbol{w}_{\mathrm{p}} = 1, \boldsymbol{w}_{\mathrm{p}}^T \boldsymbol{w}_{\mathrm{q}} = 0 \; (\mathrm{q} = 1, \cdots, \mathrm{p}-1)$

Then construct the Lagrange multiplier function $g(\boldsymbol{w}_{\mathrm{p}})$ as:

$$\begin{aligned} g(\boldsymbol{w}_{\mathrm{p}}) = \boldsymbol{w}_{\mathrm{p}}^T (X\tilde{L}^p X^T)\boldsymbol{w}_{\mathrm{p}} - \alpha(\boldsymbol{w}_{\mathrm{p}}^T XLX^T \boldsymbol{w}_{\mathrm{p}} - 1) \\ -\beta_1 \boldsymbol{w}_{\mathrm{p}}^T \boldsymbol{w}_1 - \beta_2 \boldsymbol{w}_{\mathrm{p}}^T \boldsymbol{w}_2 \cdots - \beta_{\mathrm{p-1}} \boldsymbol{w}_{\mathrm{p}}^T \boldsymbol{w}_{\mathrm{p-1}} \end{aligned} \tag{11}$$

Let $\partial g(\boldsymbol{w}_{\mathrm{p}}) / \partial \boldsymbol{w}_{\mathrm{p}} = 0$, we have:

$$2(X\tilde{L}^p X^T)\boldsymbol{w}_{\mathrm{p}} - 2\alpha XLX^T \boldsymbol{w}_{\mathrm{p}} - \beta_1 \boldsymbol{w}_1 - \cdots - \beta_{\mathrm{p-1}} \boldsymbol{w}_{\mathrm{p-1}} = 0 \tag{12}$$

Multiplying (12) by $\boldsymbol{w}_{\mathrm{p}}^T$, we get

$$\alpha = (\boldsymbol{w}_{\mathrm{p}}^T X\tilde{L}^p X^T \boldsymbol{w}_{\mathrm{p}}) / (\boldsymbol{w}_{\mathrm{p}}^T XLX^T \boldsymbol{w}_{\mathrm{p}}) .$$

It can be seen that α coincides with the objective function in (10), thus the optimization problem (10) is equivalent to maximizing α in (12). Multiplying (12) successively by $\boldsymbol{w}_{\mathrm{q}}^T (XLX^T)^{-1}$ (q=1,...,p-1), we get the following (p-1) equations:

$$\begin{cases} \beta_1 \boldsymbol{w}_1^T (XLX^T)^{-1} \boldsymbol{w}_1 + \cdots + \beta_{\mathrm{p-1}} \boldsymbol{w}_1^T (XLX^T)^{-1} \boldsymbol{w}_{\mathrm{p-1}} = 2\boldsymbol{w}_1^T (XLX^T)^{-1} (X\tilde{L}^p X^T)\boldsymbol{w}_{\mathrm{p}} \\ \vdots \\ \beta_1 \boldsymbol{w}_{\mathrm{p-1}}^T (XLX^T)^{-1} \boldsymbol{w}_1 + \cdots + \beta_{\mathrm{p-1}} \boldsymbol{w}_{\mathrm{p-1}}^T (XLX^T)^{-1} \boldsymbol{w}_{\mathrm{p-1}} = 2\boldsymbol{w}_{\mathrm{p-1}}^T (XLX^T)^{-1} (X\tilde{L}^p X^T)\boldsymbol{w}_{\mathrm{p}} \end{cases}$$

Let $\boldsymbol{\beta}^{(\mathrm{p-1})} = [\beta_1, \beta_2, \cdots, \beta_{\mathrm{p-1}}]^T$, $W^{(\mathrm{p-1})} = [\boldsymbol{w}_1, \cdots, \boldsymbol{w}_{\mathrm{p-1}}]$ and $H^{(\mathrm{p-1})} = [W^{(\mathrm{p-1})}]^T (XLX^T)^{-1} W^{(\mathrm{p-1})}$, then the above (p-1) equations can be written as:

$$\boldsymbol{\beta}^{(\mathrm{p-1})} = 2[H^{(\mathrm{p-1})}]^{-1} [W^{(\mathrm{p-1})}]^T (XLX^T)^{-1} (X\tilde{L}^{\mathrm{p}} X^T)\boldsymbol{w}_{\mathrm{p}} \tag{13}$$

Again, multiplying (12) by $(XLX^T)^{-1}$, we get

$$2(XLX^T)^{-1}(X\tilde{L}^p X^T)\boldsymbol{w}_{\mathrm{p}} - 2\alpha \boldsymbol{w}_{\mathrm{p}} - (XLX^T)^{-1} W^{(\mathrm{p-1})} \boldsymbol{\beta}^{(\mathrm{p-1})} = 0 \tag{14}$$

Replacing $\boldsymbol{\beta}^{(\mathrm{p-1})}$ in (14) with (13), we have:

$$\begin{aligned} &(I - (XLX^T)^{-1} W^{(\mathrm{p-1})} [H^{(\mathrm{p-1})}]^{-1} [W^{(\mathrm{p-1})}]^T) \\ &(XLX^T)^{-1} (X\tilde{L}^p X^T)\boldsymbol{w}_{\mathrm{p}} = \alpha \boldsymbol{w}_{\mathrm{p}} \end{aligned} \tag{15}$$

, where I is the identity matrix. Then the optimization problem (10) is transferred to find $\boldsymbol{w}_{\mathrm{p}}$ in (15) such that α is maximized. Obviously, α and $\boldsymbol{w}_{\mathrm{p}}$ can be calculated as the largest eigenvalue and the corresponding eigenvector of $(I - (XLX^T)^{-1} W^{(\mathrm{p-1})} [H^{(\mathrm{p-1})}]^{-1} [W^{(\mathrm{p-1})}]^T)(XLX^T)^{-1}(X\tilde{L}^p X^T)$.

In the same way, $\boldsymbol{w}_{\mathrm{p+1}}, \cdots, \boldsymbol{w}_{\mathrm{n}}$ (p>1) in (10) are resolved. The complete procedure of ODNPE is described in Table 1.

Table 1. ODNPE

```
Input: $X = [x_1, x_2, \cdots, x_N]$ , the dimensions of PCA subspace m (optional), the final dimensions n.
1. Perform PCA on $X$, then update $X$ with PCA coefficients, i.e. $X = W^T_{\mathrm{PCA}} X$ (optional).
2. Construct the affinity graph $G$ by (2) and the penalty graph $\tilde{G}'$ by (8); compute the laplacian matrix $L$ and $\tilde{L}^p$ .
3. Calculate $w_1$ as the eigenvector of $(XLX^T)^{-1} X\tilde{L}^{\mathrm{p}} X^T$ associated with the largest eigenvalue.
For p=2 to n do:
    Update $W^{(\mathrm{p-1})} = [w_1, \cdots, w_{\mathrm{p-1}}]$ ,
    $H^{(\mathrm{p-1})} = [W^{(\mathrm{p-1})}]^T (XLX^T)^{-1} W^{(\mathrm{p-1})}$ ,
    $\Gamma^{(\mathrm{p}-1)} = (I - (XLX^T)^{-1} W^{(\mathrm{p-1})} [H^{(\mathrm{p-1})}]^{-1} [W^{(\mathrm{p-1})}]^T)$
        $(XLX^T)^{-1} (X\tilde{L}^p X^T)$
    Calculate $w_{\mathrm{p}}$ as the eigenvector of $\Gamma^{(\mathrm{p}-1)}$ associated with the largest eigenvalue.
End.
Output: The projection matrix $W = [w_1, w_2, \cdots, w_{\mathrm{n}}]$ .
```

4. EXPIRIMENTS

In this section we apply our ODNPE to facial expression recognition and compare it with some predominant and related algorithms, i.e., LDA, LPP, OLPP, NPE and DLLE/L. All the algorithms are conducted in the supervised mode (class labels of training samples are known). The Euclidean distance and the nearest neighbor classifier are adopted. Two facial expression databases, the JAFFE database and the Cohn-Kanade database are used, both of which contain 6 basic expressions: happiness, anger, fear, sadness, disgust and surprise, and all the gray level images are cropped and normalized to 64×64 pixels (see Fig.1 and Fig.2).

4.1. Experiments on JAFFE database

The JAFFE database contains 213 static facial expression images captured from 10 Japanese females. We select 25 images per expression as the total set. Then for each expression we leave one sample of the total set for testing with the rest for training and repeat it 25 times until all the samples in the total set are tested. We take the average recognition rates over 25 times' testing. It can be seen that our ODNPE performs better than the other algorithms across a wide range of dimensions (Fig.3). And ODNPE gets the highest top recognition rate (shown in Table 2), although the corresponding dimension is relative high. The involved parameters for these algorithms are set as follows: the nearest neighbors' number is uniformly set to 5 when constructing the intra-class affinity graph for LPP, OLPP, NPE, DLLE/L and ODNPE; the interclass nearest neighbors' number k_2 for the old penalty graph defined in (4) is 100; the balancing parameter h in (7) is 1; the nearest neighbors' number k between pair-wise classes for our novel penalty graph defined in (8) is 20.

4.2. Experiments on Cohn-Kanade database

The Cohn-Kanade database contains about 500 image sequences from 100 subjects with each sequence recording a series of dynamic expressions. For each expression, we select 160 images as the total set, and then randomly chose r (=20, 40, 60, 80) images from the total set for training and leave the rest for testing. The top average results over 10 random splitting of the total set are list in Table 3. We can see that ODNPE always achieves higher recognition rates than DLLE/L, LDA, NPE and LPP in spite of the training

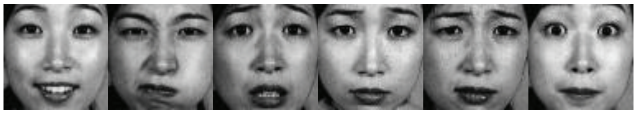

Fig.1. The six expressions of one person in JAFFE database

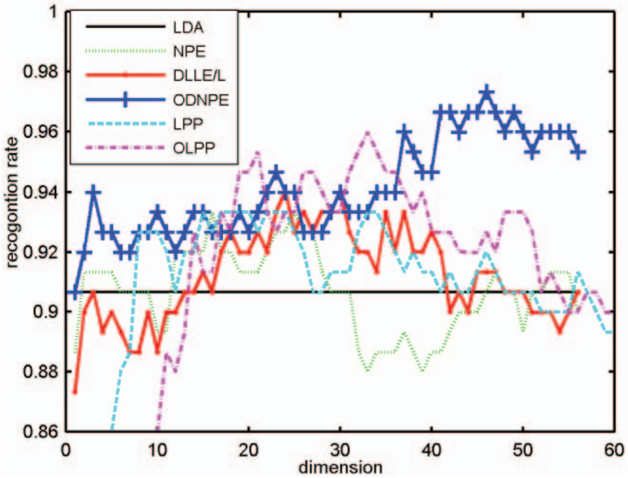

Fig. 3. Recognition rates on JAFFE database[1].

Table 2. Top recognition rates on JAFFE database.

algorithms	recognition rate	dimension
LDA	0.9067	5
NPE	0.9333	29
LPP	0.9333	33
OLPP	0.9600	33
DLLE/L	0.9400	28
ODNPE	0.9733	50

samples' number; and OLPP is comparative with our algorithm. The parameters are set as the same as that in 4.1.

5. CONCLUSIONS

In this paper, we propose the Orthogonal Discriminant Neighborhood Preserving Embedding (ODNPE) algorithm, which improves the former DLLE/L by designing a novel penalty graph and pursuing the orthogonal projection vectors. The experimental results on two facial expression databases show the effectivity of the proposed algorithm.

6. REFERENCE

[1] S.T. Roweis, and L.K. Saul, "Nonlinear Dimensionality Reduction by Locally Linear Embedding". *Science*, 2000. 290: pp.2323-2326.

[2] S.C. Yan, D. Xu, B.Y. Zhang and H.J. Zhang, "Graph Embedding: A General Framework for Dimensionality Reduction", *In: Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition*, vol. 2, pp. 830- 837, 2005.

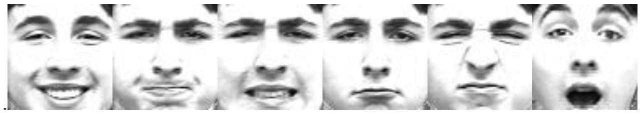

Fig.2. The six expressions of one person in Cohn-Kanade database.

Table 3. Top recognition rates on Cohn-Kanade database.

algorithms	recognition rate	dimension	training samples
LDA	0.7988	5	20
NPE	0.8024	5	
LPP	0.7644	10	
OLPP	0.8125	20	
DLLE/L	0.7024	10	
ODNPE	0.8179	7	
LDA	0.8847	5	40
NPE	0.8861	56	
LPP	0.8708	17	
OLPP	0.9057	46	
DLLE/L	0.8992	12	
ODNPE	0.8986	14	
LDA	0.9483	5	60
NPE	0.9483	10	
LPP	0.9265	23	
OLPP	0.9420	60	
DLLE/L	0.9567	13	
ODNPE	0.9650	11	
LDA	0.9563	5	80
NPE	0.9417	17	
LPP	0.9587	25	
OLPP	0.9665	38	
DLLE/L	0.9333	34	
ODNPE	0.9696	41	

[3] M.A. Turk, and A.P. Pentland, "Face recognition using eigenfaces", *In: Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pp. 586-591, 1991.

[4] D. Cai, X. He, and J. Han, "Orthogonal Laplacianfaces for Face Recognition", *IEEE transactions on image processing*, vol. 15, pp. 3609-3614, 2006.

[5] P.N. Belhumeur, J.P. Hespanha, and D.J. Kriegman, "Eigenfaces vs. Fisherfaces: recognition using class specific linear projection", *IEEE Trans. on PAMI*, vol. 19, pp. 711-720, 1997.

[6] X.F. He and P. Niyogi, "Locality preserving projections", *In: Proceedings of the Conference on Advances in Neural Information Processing Systems*, pp. 153-160, 2003.

[7] M. Belkin, and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation". *Neural Computation*, vol. 15, pp. 1373–1396, 2003.

[8] X.L. Li, S. Lin, S.C. Yan, and D. Xu, "Discriminant locally linear embedding with high order tensor data", *IEEE Trans. Syst., Man, Cybern.*, vol. 38, pp. 342–352, 2008.

[9] X.F. He, D. Cai, S.C. Yan and H.J. Zhang, "Neighborhood preserving embedding", *In: Proceedings of IEEE International Conference on Computer Vision*, vol. 2, pp. 1208 – 1213, 2005.

[1] We only compute the recognition rate of LDA corresponding to C-1 dimensions (C is the number of classes, i.e., 6 here).