Boosting Performance for 2D Linear Discriminant Analysis via Regression

Nam Nguyen, Wanquan Liu and Svetha Venkatesh
Department of Computing, Curtin University)
{Thanh.Nguyen, W.Liu, S. Venkatesh}@curtin.edu.au

Abstract

Two Dimensional Linear Discriminant Analysis (2DLDA) has received much interest in recent years. However, 2DLDA could make pairwise distances between any two classes become significantly unbalanced, which may affect its performance. Moreover 2DLDA could also suffer from the small sample size problem. Based on these observations, we propose two novel algorithms called Regularized 2DLDA and Ridge Regression for 2DLDA (RR-2DLDA). Regularized 2DLDA is an extension of 2DLDA with the introduction of a regularization parameter to deal with the small sample size problem. RR-2DLDA integrates ridge regression into Regularized 2DLDA to balance the distances among different classes after the transformation. These proposed algorithms overcome the limitations of 2DLDA and boost recognition accuracy. The experimental results on the Yale, PIE and FERET databases showed that RR-2DLDA is superior not only to 2DLDA but also other state-of-the-art algorithms.

1 Introduction

Face recognition has attracted much attention in recent years. Well-known algorithms including Eigenface [12] and Fisherface [2] work on vector representation of image and need to compute the eigenvectors of high dimensional covariance matrix in order to find the optimal linear transformation. When the size of an image is large, these algorithms may have computing problem in eigen-decomposition. To avoid this problem, Ye *et al.* proposed Two Dimensional Linear Discriminant Analysis (2DLDA) [13], which works directly on a matrix representation of images. However, 2DLDA could suffer from the small sample size problem [10]. Moreover, the transformation in 2DLDA could make the pairwise distances of any two distinct classes significantly unbalanced, which may reduce the recognition accuracy.

Based on these observations, in this paper we propose two novel algorithms to boost the performance of 2DLDA, one called Regularized 2DLDA and the other called Ridge Regression 2DLDA (RR-2DLDA). Regularized 2DLDA works directly on a matrix representation of images with low computational costs inherited from 2DLDA. A regularization parameter is introduced in Regularized 2DLDA to deal with the small sample size problem. RR-2DLDA is a further extension of Regularized 2DLDA aiming to balance the pairwise distances among distinct classes. In order to do so, we first define a set of mapping points that are distributed evenly in a low-dimensional space. Then an optimal transformation that maps the images of each class to a specific point is obtained via ridge regression as reported in [1]. Because these mapping points have been pre-defined evenly in the reduced space, RR-2DLDA can overcome the unbalance problem, making RR-2DLDA superior to Regularized 2DLDA. Moreover, RR-2DLDA avoids eigen-decomposition for a high dimensional covariance matrix and uses the advantage of 2DLDA, thus having low computational cost.

The contributions of this paper are as follows: (1) Regularized 2DLDA for face recognition, (2) integration of ridge regression into Regularized 2DLDA in a novel framework, (3) a strategy to learn parameters for the proposed framework, and (4) experiments on three benchmark data sets (the Yale, PIE and FERET databases) to demonstrate that RR-2DLDA is superior not only to 2DLDA but also other state-of-the-art algorithms such as PCA [12], Fisherface [2], Locality Preserving Projection (LPP) [7] and Spectral Regression (SR) [4].

2 Regularized 2DLDA

2.1 2DLDA

Assume that $\mathbf{X}_1, \ldots, \mathbf{X}_m$ ($\mathbf{X}_i \in \mathbb{R}^{n_1 \times n_2}$) are training images belonging to C classes $\Pi_1, \ldots, \Pi_C$ and m_c is the number of images in Π_c. 2DLDA [13] aims to find two optimal matrices $\mathbf{L} \in \mathbb{R}^{n_1 \times L_1}$ and $\mathbf{R} \in \mathbb{R}^{n_2 \times L_2}$ to project a face image $\mathbf{X} \in \mathbb{R}^{n_1 \times n_2}$ to $f(\mathbf{X}) = \mathbf{L}^T\mathbf{X}\mathbf{R} \in \mathbb{R}^{L_1 \times L_2}$, where L_1 and L_2 are the reduced height and width parameters, respectively. Let $\mathbf{M}_c = (1/m_c)\sum_{\mathbf{X}\in\Pi_c}\mathbf{X}$ and $\mathbf{M} = (1/m)\sum_{i=1}^{m}\mathbf{X}_i$ be the mean of each class Π_c and the mean of all training images, respectively. The within-class distance and

978-1-4244-2175-6/08/$25.00 ©2008 IEEE

between-class distance of training images after projection are $\mathbf{D}_w = \sum_{c=1}^{C} \sum_{\mathbf{X} \in \Pi_c} \parallel \mathbf{L}^T(\mathbf{X} - \mathbf{M}_c)\mathbf{R} \parallel_F^2$ and $\mathbf{D}_b = \sum_{c=1}^{C} m_c \parallel \mathbf{L}^T(\mathbf{M}_c - \mathbf{M})\mathbf{R} \parallel_F^2$, respectively, where $\| \, . \, \|_F$ is the Frobenius norm [6]. Let us denote

$$\mathbf{S}_w^R = \sum_{c=1}^{C} \sum_{\mathbf{X} \in \Pi_c} (\mathbf{X} - \mathbf{M}_c)\mathbf{R}\mathbf{R}^T(\mathbf{X} - \mathbf{M}_c)^T \quad (1)$$

$$\mathbf{S}_b^R = \sum_{c=1}^{C} m_c(\mathbf{M}_c - \mathbf{M})\mathbf{R}\mathbf{R}^T(\mathbf{M}_c - \mathbf{M})^T \quad (2)$$

$$\mathbf{S}_w^L = \sum_{c=1}^{C} \sum_{\mathbf{X} \in \Pi_c} (\mathbf{X} - \mathbf{M}_c)^T\mathbf{L}\mathbf{L}^T(\mathbf{X} - \mathbf{M}_c) \quad (3)$$

$$\mathbf{S}_b^L = \sum_{c=1}^{C} m_c(\mathbf{M}_c - \mathbf{M})^T\mathbf{L}\mathbf{L}^T(\mathbf{M}_c - \mathbf{M}) \quad (4)$$

According to derivations in [13], we have $\mathbf{D}_w = \text{trace}(\mathbf{L}^T\mathbf{S}_w^R\mathbf{L}) = \text{trace}(\mathbf{R}^T\mathbf{S}_w^L\mathbf{R})$ and $\mathbf{D}_b = \text{trace}(\mathbf{L}^T\mathbf{S}_b^R\mathbf{L}) = \text{trace}(\mathbf{R}^T\mathbf{S}_b^L\mathbf{R})$. 2DLDA aims to obtain $\mathbf{L}$ and $\mathbf{R}$ by maximizing the Fisher's discriminant criterion $\mathbf{D}_b/\mathbf{D}_w$. This is a nonlinear optimization problem and an iterative algorithm is proposed in [13] to solve $\mathbf{L}$ and $\mathbf{R}$. In detail, starting with an initial $\mathbf{R} = [\mathbf{I}_{L_2}, 0]^T$, where $\mathbf{I}_{L_2}$is an $L_2 \times L_2$ identity matrix, $\mathbf{L}$ is obtained as $\mathbf{L} = \arg\max_{\mathbf{L}}(\text{trace}(\mathbf{L}^T\mathbf{S}_b^R\mathbf{L})/\text{trace}(\mathbf{L}^T\mathbf{S}_w^R\mathbf{L}))$. We then fix $\mathbf{L}$ and obtain $\mathbf{R}$ as $\mathbf{R} = \arg\max_{\mathbf{R}}(\text{trace}(\mathbf{R}^T\mathbf{S}_b^L\mathbf{R})/\text{trace}(\mathbf{R}^T\mathbf{S}_w^L\mathbf{R}))$. After obtaining $\mathbf{R}$, we fix $\mathbf{R}$ and compute $\mathbf{L}$ again, and so on. The convergence issue for such iterations has been discussed in [13]. The computational cost of 2DLDA is $\mathcal{O}(mdn_1n_2)$, where $d = \max(L_1, L_2)$.

2.2 Regularized 2DLDA

In practice, 2DLDA could suffer from the small sample size problem since the within-class scatter matrices $\mathbf{S}_w^R$ and $\mathbf{S}_w^L$ may not be estimated correctly when the number of training samples is small. This could reduce the performance of 2DLDA significantly. In order to solve this problem, we propose an algorithm called Regularized 2DLDA, in which a regularization term is introduced to the within-class scatter matrices

$$\tilde{\mathbf{S}}_w^R = \mathbf{S}_w^R + \alpha_{lda}\mathbf{I} \text{ and } \tilde{\mathbf{S}}_w^L = \mathbf{S}_w^L + \alpha_{lda}\mathbf{I} \quad (5)$$

where α_{lda} is a regularization parameter. The regularization term can reduce the bias level of the within-class scatter matrix estimation. With the introduction of the regularization term, $\mathbf{L}$ and $\mathbf{R}$ are obtained by solving the following optimization problems.

$$\mathbf{L} = \arg\max_{\mathbf{L}} \frac{\text{trace}(\mathbf{L}^T\mathbf{S}_b^R\mathbf{L})}{\text{trace}(\mathbf{L}^T\tilde{\mathbf{S}}_w^R\mathbf{L})} \quad (6)$$

$$\mathbf{R} = \arg\max_{\mathbf{R}} \frac{\text{trace}(\mathbf{R}^T\mathbf{S}_b^L\mathbf{R})}{\text{trace}(\mathbf{R}^T\tilde{\mathbf{S}}_w^L\mathbf{R})} \quad (7)$$

The same method in 2DLDA is adopted to obtain $\mathbf{L}$ and $\mathbf{R}$, while the parameter α_{lda} can be learned cross-validation [5].

3 Regression for 2DLDA

Regularized 2DLDA may have unbalanced pairwise distances between two distinct classes, and thus have reduced performance. In order to tackle this unbalanced distance problem, we apply ridge regression reported in [1] for 2DLDA and abbreviate this proposed algorithm as RR-2DLDA. The training stage of RR-2DLDA has two phases. The first phase is as the same as Regularized 2DLDA and it finds optimal $\mathbf{L}$ and $\mathbf{R}$ by solving the optimization problems in Eq. 6. The training images $\mathbf{X}_1, \ldots, \mathbf{X}_m$ are then projected into $\mathbb{R}^{L_1 \times L_2}$ space as $\tilde{\mathbf{X}}_i = f(\mathbf{X}) = \mathbf{L}_i^T\mathbf{X}_i\mathbf{R}$. In the second phase, RR-2DLDA first defines a set of mapped points for all classes such that these mapped points are distributed evenly in the reduced space. Then, RR-2DLDA finds an optimal compound mapping f_A to minimize the sum of the Frobenius distances from $f_A(\tilde{\mathbf{X}}_1), \ldots, f_A(\tilde{\mathbf{X}}_m)$ to the corresponding mapped points. The mapping f_A can be obtained by solving a ridge regression problem. In practice, we are mainly interested in f_A as a linear transformation. With this linear constraint, if the mapped points are vectors, $\tilde{\mathbf{X}}$ is required to be converted from a matrix to a vector. If the mapped points are matrices, no conversion is required for $\tilde{\mathbf{X}}$. Corresponding to these two cases, we have two algorithms called 1D-RR-2DLDA and 2D-RR-2DLDA, respectively. However, in our experiments, the 2D-RR-2DLDA does not perform better than 1D-RR-2DLDA. Thus, in the remaining of the paper, we focus on discussion of 1D-RR-2DLDA.

3.1 1D-RR-2DLDA

In 1D-RR-2DLDA, the mapped points $\mathbf{y}_1, \ldots, \mathbf{y}_C$ are vectors in the mapping space. We select the mapping space as $\mathbb{R}^C$, where C is the number of classes. The predefined mapping points $\mathbf{y}_1, \ldots, \mathbf{y}_C$ in $\mathbb{R}^C$ can be as: $\mathbf{y}_1 = [1, 0, \ldots, 0]^T$, $\mathbf{y}_2 = [0, 1, \ldots, 0]^T$,..., $\mathbf{y}_C = [0, 0, \ldots, 1]^T$. The Euclidean distance between $\mathbf{y}_i$ and $\mathbf{y}_j$ are $\parallel \mathbf{y}_i - \mathbf{y}_j \parallel_2 = \sqrt{2}$, implies that $\mathbf{y}_1, \ldots, \mathbf{y}_C$ are distributed evenly in $\mathbb{R}^C$. It can be proved that this selection is equivalent to the selection of a simplex in [1]. In order to use vector representation in ridge regression, $\tilde{\mathbf{X}}_i$ is converted from matrix to a vector in $\mathbb{R}^{L_1L_2}$ as: $\tilde{\mathbf{x}}_i = \text{vectorize}(\tilde{\mathbf{X}}_i)$. 1D-RR-2DLDA aims to find a linear transformation $f_A(\tilde{\mathbf{x}}) = \mathbf{A}^T\tilde{\mathbf{x}}$, such that $\mathbf{A}^T\tilde{\mathbf{x}}_i = \mathbf{y}_{c_i} (\forall i = 1, \ldots, m)$, where c_i is the class that $\mathbf{X}_i$ belongs to. In reality, such $\mathbf{A}$ might not exist. Thus, $\mathbf{A}$

Algorithm 1 1D-RR-2DLDA

Input: training images $\mathbf{X}_1, \ldots, \mathbf{X}_m$; training classes $\Pi_1, \ldots, \Pi_C$; Regularized parameters α_{lda}, α_r; reduced height L_1, reduced width L_2

Output: transformation matrices $\mathbf{L}, \mathbf{R}, \mathbf{A}$; projected images $\tilde{\mathbf{y}}_1, \ldots, \tilde{\mathbf{y}}_m$

Algorithm:

1. Obtain $\mathbf{L}$ and $\mathbf{R}$ by using Regularized 2DLDA.
2. Project training images using $\mathbf{L}$ and $\mathbf{R}$: $\tilde{\mathbf{X}}_1 = \mathbf{L}^T\mathbf{X}_1\mathbf{R}, \ldots, \tilde{\mathbf{X}}_m = \mathbf{L}^T\mathbf{X}_1\mathbf{R}$.
3. Vectorize projected images: $\tilde{\mathbf{x}}_i$ = vectorize$(\tilde{\mathbf{X}}_i)$, $\forall i = 1, \ldots, m$.
4. Define mapped points $\mathbf{y}_1, \ldots, \mathbf{y}_C \in \mathbb{R}^C$.
5. Obtain transformation matrix $\mathbf{A}$ from Eq. 9.
6. Compute final projected images: $\tilde{\mathbf{y}}_1 = \mathbf{A}^T\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{y}}_m = \mathbf{A}^T\tilde{\mathbf{x}}_m$.

Testing: The class of a test image $\mathbf{X}$ is obtained as follows:

1. Compute: $\tilde{\mathbf{X}}$=$\mathbf{L}^T\mathbf{X}\mathbf{R}$, $\tilde{\mathbf{x}}$ = vectorize$(\tilde{\mathbf{X}})$, $\mathbf{y} = \mathbf{A}^T\tilde{\mathbf{x}}$.
2. Compare $\mathbf{y}$ with $\tilde{\mathbf{y}}_1, \ldots, \tilde{\mathbf{y}}_m$ to find the class of $\mathbf{X}$ using the nearest center classifier.

can be obtained by solving the following ridge regression problem:

$$\mathbf{A} = \arg\min_{\mathbf{A}}(\sum_{i=1}^{m} \parallel \mathbf{A}\tilde{\mathbf{x}}_i - \mathbf{y}_{c_i} \parallel_2^2 + \alpha_r \parallel \mathbf{A} \parallel_F^2) \quad (8)$$

where α_r is the regularization parameter. Let $\mathbb{X} = [\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_m]$ and $\mathbb{Y} = [\mathbf{y}_{c_1}, \ldots, \mathbf{y}_{c_m}]$. Taking derivatives on the right block of Eq. 8 and setting them equal to zero, we have

$$\mathbf{A} = (\mathbb{X}\mathbb{X}^{\mathrm{T}} + \alpha_r\mathbf{I}_{L_1L_2})^{-1}\mathbb{X}\mathbb{Y}^{\mathrm{T}} \quad (9)$$

where $\mathbf{I}_{L_1L_2}$ is an $L_1L_2 \times L_1L_2$ identity matrix. The computational cost of 1D-RR-2DLDA equals the sum of the computational costs of solving a ridge regression problem in Eq. 8 and Regularized 2DLDA. The details of 1D-RR-2DLDA is presented in Algorithm 1. It should be remembered that ridge regression is taken on the reduced images of Regularized 2DLDA, which can be solved more easily than the case in [1], which is taken on the original images.

3.2 Selection of Parameters

The parameters in RR-2DLDA include L_1, L_2, α_{lda} in Regularized 2DLDA and α_r in ridge regression. These parameters can be selected by leave-one-out cross-validation [5]. Usually, the computational cost to learn L_1, L_2, α_{lda} and α_r is high because there are a huge number of possible values for these of parameters. We propose two steps to make the computational cost less expensive:

1. Discretize the search space of the parameters.
2. Learn the parameters L_1 and L_2 first, then learn the parameters α_{lda} and α_r.

We assume that the optimal L_1 and L_2 for 2DLDA are also the optimal L_1 and L_2 for RR-2DLDA. Thus, L_1 and L_2 can be learned using cross-validation for 2DLDA. Then, we fix L_1, L_2 and learn α_{lda} and α_r using cross-validation for RR-2DLDA. With this strategy, we can select parameters L_1, L_2, α_{lda} and α_r for RR-2DLDA.

4 Experimental results

4.1 Experiments on the Yale face database

The Yale face database[1] has 165 face images of 15 people with each person having 11 images. These images were resized to 32×32. Three experiments (2-train, 3-train and 4-train) were considered, where in the $i-$train experiment, i images of each person were used for training and the remaining images are used for testing. For each experiment, 30 random splits (train images, test images) of the Yale face database were created. We tested the proposed algorithms on these splits and take the average of the results.

We discretized the parameter search space of 1D-RR-2DLDA to reduce the computational cost with the following options: $L_1 \in \{5, 10, 15, 20, 25, 30\}$, $L_2 \in \{5, 10, 15, 20, 25, 30\}$, $\alpha_{lda} \in \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^0\}$ and $\alpha_r \in \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^0\}$. We obtained the parameters for 1D-RR-2DLDA using the described cross-validation approach, resulting in $(L_1, L_2, \alpha_{lda}, \alpha_r) = (15, 15, 10^{-2}, 10^{-4})$.

We ran Regularized 2DLDA and 1D-RR-2DLDA on 30 random splits of each i-train experiment ($i = 2, 3, 4$). Table 1 shows that Regularized 2DLDA can boost the recognition accuracy of 2DLDA.

The recognition accuracy of 1D-RR-2DLDA was compared with PCA [12], Fisherface [2], LPP [7], SR [4], Ridge Regression (RR) [1] and 2DLDA [13] in Table 1. The table shows that the accuracy rate of 1D-RR-2DLDA is the highest in all experiments across the different algorithms, boosting the recognition accuracy for 2DLDA significantly.

4.2 Experiments on the PIE

The PIE face database [11] has face images captured by 13 cameras under different poses and illumination conditions. We selected images from five near frontal poses (C05, C07, C09, C27 and C27) and resized the images to 32×32. In total, 11554 images of 68 people were

[1] http://vismod.media.mit.edu/vismod/classes/mas622-00/datasets/

Table 1. Performance (%) on Yale database.

	2-train	3-train	4-train
PCA [12]	76.94 ±3.37	79.58 ±2.99	81.21 ±2.21
Fisherface [2]	83.90 ±3.86	94.50 ±2.08	97.05 ±1.63
LPP [7]	90.47 ±3.41	95.56 ±1.92	97.52 ±1.73
SR [4]	90.96 ±3.44	95.03 ±2.02	97.56 ±1.56
RR [1]	89.75 ± 3.09	94.42 ± 1.77	97.08 ± 1.58
2DLDA [13]	86.27 ±4.43	92.64 ±2.61	94.35 ±2.53
Regularized 2DLDA	87.51 ±4.17	93.39 ±2.73	95.21 ±2.11
1D-RR-2DLDA	**92.35** ±2.97	**96.47** ±2.38	**97.87** ±1.51

Table 2. Performance (%) on PIE database.

	2-train	3-train	4-train
PCA	17.4 ±0.7	22.6±1.0	26.4±0.9
Fisherface	34.2±1.7	53.4±2.1	62.0±1.7
LPP	34.6±1.8	43.0±1.6	49.4±1.6
SR	45.3 ±1.4	58.4 ±1.9	66.7 ±1.7
RR	44.5 ±1.8	57.4 ±1.8	65.4 ±1.6
2DLDA	49.5 ±2.7	60.1 ±2.3	66.8 ±1.7
Regularized 2DLDA	50.0 ±2.5	60.6 ±1.9	66.6 ±1.7
1D-RR-2DLDA	**51.7 ±2.4**	**66.0 ±2.2**	**73.7 ±1.6**

selected. We again compared performance of 1D-RR-2DLDA versus other algorithms in the 2-train, 3-train and 4-train experiments. 1D-RR-2DLDA was run with $L_1 = 20$, $L_2 = 5$, $\alpha_{lda} = 10^{-1}$ and $\alpha_r = 10^{-2}$ which were obtained by cross-validation. Table 2 compares the recognition accuracy of 1D-RR-2DLDA with the other algorithms. One can see that, in all experiments 1D-RR-2DLDA achieves the highest accuracy rate.

4.3 Experiments on the FERET database

We used the FERET database [8, 9] to test the performance of 1D-RR-2DLDA. We selected all people in FERET having at least four frontal images. In total, 1433 images of 240 people were selected. The images were pre-processed using the CSU Face Identification Evaluation System [3] . Two experiments (2-train, 3-train) were considered. For each experiment, 30 random splits (training images, test images) of the database were created. 1D-RR-2DLDA was run with $L_1 = 25$, $L_2 = 25$, $\alpha_{lda} = 10^0$ and $\alpha_r = 10^{-2}$, which were obtained by cross-validation. Table 3 shows the average accuracy rate of 1D-RR-2DLDA compared with the other algorithms. One can observe that 1D-RR-2DLDA outperforms the other algorithms in all of the experiments.

Table 3. Performance (%) on FERET.

	2-train	3-train
PCA	70.83 ±1.57	79.23 ±1.72
Fisherface	84.41 ±1.32	91.00 ±1.41
LPP	84.55 ±1.55	91.60 ±1.47
SR	85.06 ±1.69	92.42 ±1.50
RR	90.30 ±1.17	95.15 ± 1.03
2DLDA	77.01 ±1.93	84.47 ±2.08
Regularized 2DLDA	77.36 ±1.64	84.84 ±1.81
1D-RR-2DLDA	**90.61** ±1.21	**95.24** ±1.11

5 Conclusions

We have presented two algorithms for face recognition, Regularized 2DLDA and Ridge Regression for 2DLDA (RR-2DLDA). Regularized 2DLDA is an extension of 2DLDA with the introduction of regularization term, while RR-2DLDA integrates ridge regression into Regularized 2DLDA. The experimental results on the Yale, PIE and FERET databases show that Regularized 2DLDA boost the recognition accuracy of 2DLDA and RR-2DLDA outperforms other state-of-the-art algorithms such as PCA, Fisherface, LPP, SR and 2DLDA.

References

[1] S. An, W. Liu, and S. Venkatesh. Face recognition using kernel ridge regression. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR'07)*, 2007.

[2] P. Belhumeur, J. Hespanda, and D. Kiregeman. Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):711–720, 1997.

[3] J. Beveridge, D. Bolme, B. Draper, and M. Teixeira. The CSU face identification evaluation system: Its purpose, features, and structure. *Machine Vision and Applications*, 16(2):128–138, 2005.

[4] D. Cai, X. He, and J. Han. Spectral regression for efficient regularized subspace learning. In *IEEE International Conference on Computer Vision (ICCV)*, Rio de Janeiro, Brazil, 2007.

[5] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, New York, 2nd edition, date 1990.

[6] G. Golub and C. V. Loan. *Matrix computations*. The Johns Hopkins University Press, MD, USA, 3rd edition, date 1996.

[7] Z. He and P. Niyogi. Locality preserving projections. In *Proceedings of Advances in Neural Information Processing Systems*, volume 16, Vancouver, Canada, 2003.

[8] P. Phillips, H. Moon, S. Rizvi, and P. Rauss. The FERET evaluation methodology for face recognition algorithms. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22:1090–1104, 2000.

[9] P. Phillips, H. Wechsler, J. Huang, and P. Rauss. The FERET database and evaluation procedure for face recognition algorithms. *Image and Vision Computing J*, 16(5):295–306, 1998.

[10] S. Raudys and A. Jain. Small sample size effects in statistical pattern recognition: Recommendations for practitioners. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(3):252–264, 1991.

[11] T. Sim, S. Baker, and M. Bsat. The CMU pose, illumination, and expression database. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(12):1615–1618, 2003.

[12] M. Turk and A. Pentland. Eigenfaces for recognition. *Cognitive Neuroscience*, 3(1):71–86, 1991.

[13] J. Ye, R. Janardan, and W. Li. Two-dimensional linear discriminant analysis. In *Proceedings of Advances in Neural Information Processing Systems*, volume 17, 2004.