10

Constructing Kernel Machines in the Empirical Kernel Feature Space

Huilin Xiong[1] and Zhongli Jiang[2]
[1]*Shanghai Jiao Tong University, Shanghai*
[2]*Shanghai University of Political Science and Law, Shanghai*
China

1. Introduction

Over the last decade, kernel-based nonlinear learning machines, e.g., support vector machines (SVMs) Vapnik (1995), kernel principal component analysis (KPCA) Scholkopf (1998), and kernel Fisher discriminant analysis (KFDA) Mika (1999), attracted a lot of attentions in the fields of pattern recognition and machine learning, and have been successfully applied in many real-world applications Mika (1999); Yang (2002); Lu (2003); Yang (2004). Basically, the kernel-based learning methods work by mapping the input data space, $\mathcal{X}$, into a high dimensional space, $\mathcal{F}$, called the kernel feature space: $\Phi : \mathcal{X} \longrightarrow \mathcal{F}$, and then building linear machines in the kernel feature space to implement their nonlinear counterparts in the input space. This procedure is also known as a "kernelization", in which the so-called kernel trick is associated in such a way that the inner product of each pair of the mapped data in the kernel feature space is calculated by a kernel function, rather than explicitly using the nonlinear map, Φ.

The kernel trick provides an easy way to kernelize linear machines. However, in many cases, formulating a kernel machine via the kernel trick could be difficult and even impossible. For example, it is pretty tough to formulate the kernel version of the direct disciminant analysis algorithm (KDDA) Lu (2003) using the kernel trick. Moreover, for some recently developed linear discriminant analysis schemes, such as the uncorrelated linear discriminant analysis (ULDA) Ye (2004), and the orthogonal linear discriminant analysis (OLDA) Ye (2005), which have been shown to be efficient in many real-world applications Ye (2004), it is impossible to directly kernelize them via the kernel trick, since these schemes need first computing the singular value decomposition (SVD) of an interim matrix, namely, H_t (see Ye (2004)), which is generally of infinite column size in the case of the kernel feature space.

Theoretically, the kernel feature space is generally an infinite dimensional Hilbert space. However, given a training data set $\{x_i\}$ $(i = 1, 2, \ldots, n)$, the kernel machines we known perform actually in a subspace of the kernel feature space, $span\Phi(x_i)$ $(i = 1, 2, \ldots, n)$, which can be embedded into a finite-dimensional Euclidean space with all data's geometrical measurements, e.g., distance and angle, being preserved Xiong (2005). This finite-dimensional embedding space, called empirical kernel feature space, provides a unified framework for kernelizing all kinds of linear machines. With this framework, kernel machines can be

"seamlessly" formulated from their linear counterparts without any difficulty: performing linear machines in the finite-dimensional empirical kernel feature space, the corresponding nonlinear kernel machines are then constructed in the input data space.

In this chapter, we propose to approach the kernelization from the empirical kernel feature space, that is, we formulate nonlinear kernel machines by directly performing their linear counterparts in the empirical kernel feature space. The kernel machines constructed, called empirical kernel machines, are usually different from the conventional kernel machines based on the kernel trick, and surprisingly, the empirical kernel machines are shown to be more efficient in many real-world applications, such as face recognition, facial expression recognition, and handwritten digit recognition, than the conventional nonlinear kernel machines and their linear counterparts.

The remainder of this chapter is organized as follows: In Section 2, we introduce the concepts and related notation concerning the empirical kernel feature space. Section 3 shows the difference in formulation between the conventional kernel principal component analysis (KPCA) and the empirical kernel principal component analysis (eKPCA), which is constructed by performing the linear principal component analysis (PCA) in the empirical kernel feature space. In Section 4, we formulate three other empirical kernel machines, namely, the empirical kernel direct discriminant analysis (eKDDA), the empirical kernel ULDA, denoted as eKUDA, and the empirical kernel OLDA, denoted as eKODA, via directly performing the DLDA Yu (2001), ULDA, and OLDA schemes in the empirical kernel feature space. Experiments for evaluating the performance of the empirical kernel machines in the real-world applications, e.g., face and facial expression recognition, are presented in Section 5.1. Finally, Section 6 concludes this chapter.

2. The empirical kernel feature space

Let $\{x_i, \xi_i\}_{i=1}^n$ be a d-dimensional training data with class labels $\{\xi_i\}$, the kernel matrix $K = [k_{ij}]_{n\times n}$, where $k_{ij} = \Phi(x_i)\cdot\Phi(x_j) = k(x_i, x_j)$, and $rank(K) = r, r \leq n$. Since K is a symmetrical positive semi-definite matrix, K can be decomposed as:

$$K_{n\times n} = P_{n\times r}\Lambda_{r\times r}P_{r\times n}^T \tag{1}$$

where Λ is a diagonal matrix only containing the r positive eigenvalues of K in decreasing order, and P consists of the eigenvectors corresponding to the positive eigenvalues. The map from the input data space to an r-dimensional Euclidean space Φ^e: $\mathcal{X} \longrightarrow \mathbf{R^r}$

$$x \longrightarrow \Lambda^{-\frac{1}{2}}P^T(k(x, x_1), k(x, x_2), \ldots, k(x, x_n))^T$$

is referred to the empirical kernel map in Xiong (2005); Scholkopf (1999). We call the subspace $span\{\Phi^e(x_i)\}$ the empirical kernel feature space, and denote it by $\mathcal{F}^e$. Obviously, we have $\overline{span\{\Phi^e(x_i)\}} \subset \overline{span\{\Phi^e(\mathcal{X})\}} \subset \mathbf{R}^r$. For the completion of the subspaces, it is easy to verify: $\overline{span\{\Phi^e(x_i)\}} = \overline{span\{\Phi^e(\mathcal{X})\}} = \mathbf{R}^r$.

It is well-known that various kernel machines, such as KPCA and SVM, perform only in a subspace of the kernel feature space: $span\{\Phi(x_i)\}$, which is actually isometric isomorphic with the empirical kernel feature space $span\{\Phi^e(x_i)\}$. In fact, let Y denote the data matrix

with size $r \times n$ in the empirical kernel feature space, that is,

$$Y = (\Phi^e(x_1), \Phi^e(x_2), \ldots, \Phi^e(x_n)) = \Lambda^{-\frac{1}{2}} P^T K. \tag{2}$$

The dot product matrix of $\{\Phi^e(x_i)\}$ in the empirical kernel feature space can be calculated as

$$Y^T Y = KP\Lambda^{-\frac{1}{2}}\Lambda^{-\frac{1}{2}} P^T K = K. \tag{3}$$

This is exactly the dot product matrix of $\{\Phi(x_i)\}$ in the feature space. Since the distances of the n vectors $\{\Phi(x_i)\}_1^n$ in the kernel feature space are uniquely determined by the dot product matrix, we can see the training data have the same distance matrix in both the empirical kernel feature space, $\mathcal{F}^e$, and the kernel feature space, $\mathcal{F}$, that is, as pointed out in Xiong (2005), $span\{\Phi(x_i)\}$ can be embedded into an r-dimensional Euclidean space with the distances between each pair of the training data being preserved. Note that the dimension of the samples in the empirical kernel feature space is always smaller than the sample size, $r \leq n$, which may help to some extent to alleviate the so-called "Small Sample Size" (SSS) problems Chen (2000); Yu (2001) in discriminant analysis.

3. Principal component analysis in the empirical kernel feature space

Principal component analysis (PCA) is a widely used subspace method in pattern recognition and dimension reduction. It gives the optimal representation of the pattern data with the minimum mean square error. The PCA transform (projection) matrix can be calculated from the eigendecomposition of the sample covariance matrix, or alternatively, from the eigendecomposition of the inner product matrix of samples in the case of high data dimensionality. Kernel principal component analysis (KPCA) is carried out by applying PCA in the kernel feature space. Using the kernel trick, the KPCA transform matrix can be computed from the eigendecomposition of the kernel matrix.

Let us perform the linear PCA in the empirical kernel feature space. The scheme obtained is called empirical kernel principal component analysis, denoted as eKPCA for short. Let K_c represent the centered kernel matrix, that is,

$$K_c = (I_{n\times n} - \frac{1}{n}1_{n\times n})K(I_{n\times n} - \frac{1}{n}1_{n\times n}),$$

where $I_{n\times n}$ is the $n \times n$ identity matrix, and $1_{n\times n}$ represents the $n \times n$ matrix with all entries being equal to unity. The centered kernel matrix can be decomposed as $K_c = Q\Sigma Q^T$, where Σ is a diagonal matrix containing the positive eigenvalues of K_c, and Q consists of the eigenvectors corresponding to the positive eigenvalues. Given a sample x, the conventional KPCA maps x to $\Sigma^{-\frac{1}{2}} Q^T (k(x, x_1), \ldots, k(x, x_n))^T$. However, when we perform the linear PCA in the empirical feature space, the x will be transformed to

$$\begin{aligned} & \Sigma^{-\frac{1}{2}} Q^T Y^T \Phi^e(x) \\ = & \Sigma^{-\frac{1}{2}} Q^T Y^T \Lambda^{-\frac{1}{2}} P^T (k(x, x_1), \ldots, k(x, x_n))^T \\ = & \Sigma^{-\frac{1}{2}} Q^T P P^T (k(x, x_1), \ldots, k(x, x_n))^T. \end{aligned}$$

This is our eKPCA formula. Note that $P^T P$ is the identity matrix of size $r \times r$, however, PP^T generally is not the identity matrix of size $n \times n$. If $Q^T PP^T = Q^T$, or equivalently, $PP^T Q = Q$

holds, our eKPCA scheme turns to be $\Sigma^{-\frac{1}{2}}Q^T(k(x,x_1),\ldots,k(x,x_n))^T$, which is actually the conventional KPCA. Many experiments (see the experiment section below) show that eKPCA and KPCA usually lead to the same results, which may suggest that the equation $PP^TQ = Q$ holds frequently in practices.

4. Discriminant analysis in the empirical kernel feature space

Currently, linear discriminant analysis (LDA) has become a classical statistical approach for pattern classification, feature extraction, and dimension reduction. It has been successfully applied in many real-world applications, e.g., face recognition Belhumeour (1997), information retrieval Berry (1995), and microarray gene expression data analysis Dudoit (2002). while PCA calculates the optimal projection for pattern representation, LDA projects data aiming to discriminate the labeled pattern data. LDA calculates the optimal projection directions by maximizing the ratio of the between-class scatter measure to the within-class scatter measure, and thus, achieves the maximum class discrimination. A big challenge facing the conventional LDA is that it requires the within-class scatter matrix (or the total scatter matrix) be nonsingular, which usually cannot be met in practices, specifically for the "SSS" problems Chen (2000); Yu (2001).

In recent years, we have witnessed a great development of the linear discriminant analysis (LDA) research in handling the problem caused by the singularity of the scatter matrices. A variety of linear schemes have been proposed, from the pseudo-inverse LDA Raudys (1998), the null space LDA Chen (2000), and the direct linear discriminant analysis (DLDA) Yu (2001), to the recently developed sophisticated schemes, the uncorrelated LDA (ULDA) Ye (2004) and orthogonal LDA (OLDA) Ye (2005).

In this section, we perform various linear discriminant analysis schemes in the r-dimensional empirical kernel feature space to formulate our kernel nonlinear discriminant analysis schemes. It needs to emphasis that, in the empirical kernel feature space, the data dimension and the scatter matrix size are always smaller than the sample size ($r \leq n$). However, even so, we still face the singularity problem of the scatter matrices. We choose to kernelize three LDA schemes, namely, the DLDA, ULDA, and OLDA schemes, which are three typical extensions of the classical LDA scheme in overcoming the singularity problem. With these examples, we want to highlight our point that performing linear LDA schemes in the empirical kernel feature space can seamlessly formulate the kernel versions of vaious linear discriminant analysis schemes.

Suppose the labeled training data $\{x_i, \xi_i\}_{i=1}^n$ are grouped into m class, and each class contains n_i samples, where $\sum_{i=1}^m n_i = n$. The data matrix of the training data in the empirical kernel feature space is Y, that is, $Y = \Lambda^{-\frac{1}{2}}P^TK$. Let us define three matrices H_b, H_w, and H_t as follows:

$$H_b = \frac{1}{\sqrt{n}}[\sqrt{n_1}(\overline{y}_1 - \overline{y}), \cdots, \sqrt{n_m}(\overline{y}_m - \overline{y})]$$

$$H_w = [\frac{1}{\sqrt{n}}(Y_1 - \overline{y}_1 1_{n_1}^T), \cdots, (Y_m - \overline{y}_m 1_{n_m}^T)]$$

$$H_t = \frac{1}{\sqrt{n}}(Y - \overline{y}1_n^T)$$

where Y_i and $\overline{y}_i$ respectively denote the data matrix and centroid of the i-th class in the empirical kernel feature space, $\overline{y}$ is the global centroid of the data in the empirical kernel feature space, and 1_{n_i} represents the n_i-dimensional vector with entries being unity. Then, the *between-class scatter matrix* S_b, the with-in class scatter matrix S_w, and the *total scatter matrix* S_t defined in Fukunaga (1990) can be represented as: $S_b = H_b H_b^T$, $S_w = H_w H_w^T$, and $S_t = H_t H_t^T$. It is easy to verify:

$$H_b = YE_b, \ H_w = YE_w, \text{ and } H_t = YE_t \tag{4}$$

and therefore, we have

$$S_b = YE_bY^T, \ S_w = YE_wY^T, \text{ and } S_t = YE_tY^T \tag{5}$$

where the three constant matrices, E_b, E_w, and E_t, are:

$$\begin{aligned} E_b &= D - \frac{1}{n}1_{n\times n} \\ E_w &= I_{n\times n} - D \\ E_t &= I_{n\times n} - \frac{1}{n}1_{n\times n} \end{aligned}$$

in which matrix D is:

$$\begin{pmatrix} \frac{1}{n_1}1_{n_1\times n_1} & & \\ & \ddots & \\ & & \frac{1}{n_m}1_{n_m\times n_m} \end{pmatrix},$$

$I_{n\times n}$ is the $n \times n$ identity matrix, and $1_{n_i\times n_i}$ represents the $n_i \times n_i$ matrix with all the entries being equal to unity.

4.1 Empirical kernel direct discriminant analysis

In discriminant analysis, it has been recognized that the null space of the within-class scatter matrix may contain significant discriminant information. The so-called "direct LDA", or DLDA in the literature, involves two schemes Chen (2000); Yu (2001) in extracting the discriminant information from the null space, and meanwhile addressing the singularity problem of the scatter matrix. Different from Chen *et.al.*'s scheme Chen (2000), Yu *et.al.*'s scheme Yu (2001) first projects the data into the range space of the between-class matrix, and then calculates the projection in the null space of the within-class scatter matrix. Yu *et.al.*'s scheme is more efficient in computation than Chen *et.al.*'s, and this scheme has been kernelized by Lu *et.al.* in Lu (2003). In this section, we formulate our kernel direct discriminant analysis by performing the Yu's DLDA scheme in the empirical kernel feature space. The obtained kernel direct discriminant analysis algorithm is called empirical kernel direct discriminant analysis, denoted as eKDDA in order to differentiate it from Lu's KDDA scheme:

- **Step 1**. Calculate the matrices Y, S_b, and S_w in Eq.(2) and Eq.(5).
- **Step 2**. Calculate the eigen decomposition of $S_b = YE_bY^T$ as $S_b = P_b\Lambda_b P_b^T$, where Λ_b is the diagonal matrix consisting of the r_b positive eigen values sorted in decreasing order, and $r_b = rank(S_b)$. Let $M_1 = P_b\Lambda_b^{-\frac{1}{2}}$.

- **Step 3**. Calculate $\widetilde{S}_w = M_1^T S_w M_1$, and decompose it as:

$$\widetilde{S}_w = \left(\widetilde{P}_w, \widetilde{N}_w\right) \begin{pmatrix} \widetilde{\Lambda}_w & \\ & 0 \end{pmatrix} \begin{pmatrix} \widetilde{P}_w^T \\ \widetilde{N}_w^T \end{pmatrix}$$

- **Step 4**. Suppose we need extracting q-dimensional feature vectors, where $q \leq m-1$. Let $M = M_1 \widetilde{N}_w(:, 1:q)$, then, for given $x \in \mathcal{X}$, eKDDA transform x to

$$G(k(x,x_1), k(x,x_2), \ldots, k(x,x_n))^T,$$

where $G = M^T \Lambda^{-\frac{1}{2}} P^T = \widetilde{N}_w^T \Lambda_b^{-\frac{1}{2}} P_b^T \Lambda^{-\frac{1}{2}} P^T$.

In the implementation of the eKDDA algorithm, to avoid possible numerical instability in step 2, we introduce an extra parameter, ε, to discard some tiny eigenvalues. The eigenvalue λ is considered to be zero if $\frac{\lambda}{\lambda_{max}} \leq \varepsilon$, where λ_{max} denotes the maximum eigenvalue. In the step 3, we only need calculate the eigen decomposition of the matrix $\widetilde{S}_w$, and sort the eigenvalues (or the absolute values of the eigenvalues) in ascend order. The $\widetilde{N}_w(:, 1:q)$ is then composed of the q eigenvectors corresponding to the first p small eigenvalues.

4.2 Empirical kernel uncorrelated and orthogonal discriminant analysis

Uncorrelated linear discriminant analysis (ULDA) Ye (2004) and orthogonal linear discriminant analysis (OLDA) Ye (2005) are two recently developed LDA schemes, in which some sophisticated matrix techniques such as singular value decomposition (SVD) and QR-decomposition are used to address the singularity problem in the classical LDA scheme. In the ULDA and OLDA algorithms, we need first compute the SVD of the matrix H_t, which makes it difficult to kernelize ULDA and OLDA directly via the conventional kernel trick, since the dimension of the matrix H_t in the kernel feature space is infinite in general. In Ji (2008), an indirect kernelization scheme of ULDA and OLDA, refereed to as KUDA and KODA, respectively, is proposed. Essentially, in the scheme of Ji (2008), KUDA "is equivalent to applying ULDA to the kernel matrix, where each column is considered as an n-dimensional data point" Ji (2008). Since the geometrical structure, e.g., distance and angle, among the "column" data of the kernel matrix is different from that of the data in the kernel feature space, some discriminatory information may be changed or lost as we use the "column" data to replace the data in the kernel feature space. On the contrary, the empirical kernel feature space preserves the geometrical structure of the training data in the kernel feature space, therefore, there would be no information loss in performing LDA in the empirical kernel feature space instead of the kernel feature space. Furthermore, our experiments show (see the experiment section) that the kernel ULDA and OLDA formulated in the empirical kernel feature space perform substantially better than KUDA and KODA in most cases.

According to the schemes of ULDA and OLDA Ye (2004; 2005), we simply perform the ULDA and OLDA algorithms in the empirical kernel feature space to formulate our empirical kernel ULDA and OLDA, denoted as eKUDA and eKODA, respectively.

4.2.1 The eKUDA algorithm

- **Step 1**. Calculate the matrices Y, H_t, and H_b in Eq.(2) and (4).
- **Step 2**. Calculate the reduced SVD of H_t as $H_t = U_t \Sigma_t V_t^T$.

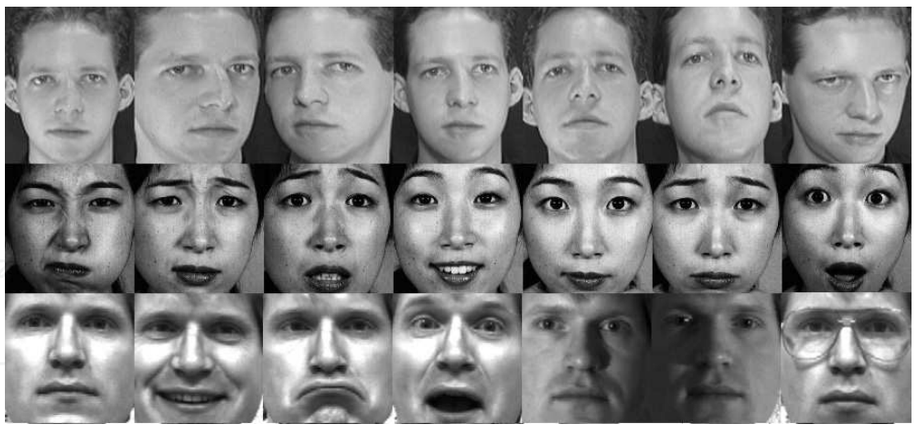


Fig. 1. Some sample images in the ORL, JAFFE, and Yale data sets.

- **Step 3**. Let $B = \Sigma_t^{-1} U_t^T H_b$, and $q = rank(B)$. Calculate the reduced SVD of B as $B = U_B \Sigma_B V_B^T$.
- **Step 4**. Let $X = U_t \Sigma_t^{-1} U_B$, $M = X(:, 1:q)$, then, for given $x \in \mathcal{X}$, eKUDA transform x to

$$G(k(x, x_1), k(x, x_2), \ldots, k(x, x_n))^T,$$

where $G = M^T \Lambda^{-\frac{1}{2}} P^T$.

4.2.2 The eKODA algorithm

- **Step 1**. Calculate the matrices Y, H_t, and H_b in Eq.(2) and (4).
- **Step 2**. Calculate the reduced SVD of H_t as $H_t = U_t \Sigma_t V_t^T$.
- **Step 3**. Let $B = \Sigma_t^{-1} U_t^T H_b$, and $q = rank(B)$. Calculate the reduced SVD of B as $B = U_B \Sigma_B V_B^T$.
- **Step4**. Let $X = U_t \Sigma_t^{-1} U_B$. Calculate the QR-decomposition of $X_q = X(:, 1:q)$ as $X_q = QR$, then, for a given sample $x \in \mathcal{X}$, eKODA transform x to

$$G(k(x, x_1), k(x, x_2), \ldots, k(x, x_n))^T,$$

where $G = Q^T \Lambda^{-\frac{1}{2}} P^T$.

5. Experiments

We conduct three types of experiments to investigate the efficiency of our empirical kernel machines in a wide range of real-world applications. We compare the performances of our empirical kernel machines, specifically, eKPCA, eKDDA, eKULDA ,and eKOLDA, with those of the kernel-trick-based machines, namely, KPCA, KDDA, KUDA, and KODA, and the linear machines, namely, PCA, ULDA, and OLDA, in the applications of face recognition, facial expression recognition, and handwritten digit recognition.

Four standard databases, including three face image data sets and one handwritten digit image data set, are used to evaluate the pattern classification algorithms

p	0.2	0.3	0.4	0.5	0.6
PCA	81.40±1.99	88.03±2.36	92.14±1.65	94.62±1.66	95.52±1.49
KPCA	81.44±1.97	88.16±2.36	92.26±1.62	94.90±1.65	95.94±1.35
eKPCA	81.44±1.97	88.15±2.36	92.26±1.62	94.90±1.65	95.94±1.35
KDDA	78.80±5.27	86.29±2.54	93.09±1.63	95.90±1.10	97.70±1.39
eKDDA	83.38±2.01	89.93±2.07	93.51±1.68	94.64±1.44	96.34±1.35
ULDA	80.84±2.57	86.46±2.01	90.18±1.91	92.05±2.26	93.33±1.49
KUDA	**85.96±2.06**	**91.78±1.88**	95.06±1.55	96.51±1.08	97.77±1.10
eKUDA	85.52±2.14	91.42±1.89	94.82±1.53	96.91±1.21	97.67±1.10
OLDA	84.96±2.18	90.86±2.09	94.18±1.47	96.01±1.25	97.25±1.35
KODA	85.07±2.44	91.41±1.95	95.08±1.75	96.57±1.16	97.84±1.33
eKODA	85.30±2.13	91.58±1.90	**95.37±1.35**	**96.95±1.10**	**98.09±1.11**

Table 1. Experimental results in terms of the average values and the standard deviations of the best recognition accuracy (%) on test data for the ORL data set

mentioned above. The three face image databases are ORL face images (available at http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html), Yale face images (available at http://cvc.yale.edu/projects/yalefaces/yalefaces.html), and JAFFE facial expression images Lyons (1998). The handwritten digit images, in size 16×16, are collected from the USPS database Hull (1994). Some samples of these image databases are shown in Fig.(1). Except the JAFFE images and Yale images, where the face part of each image is cropped, in size of 128×128 and 112×112, respectively, from the original images, no any other preprocessing is applied to the images. The ORL and Yale data are used to evaluate the algorithms for the task of face recognition, and the JAFFE face images are used for facial expression recognition.

We only consider the Gaussian kernel, $k(x, y) = exp(-\gamma\|x - y\|^2)$, in this chapter. There is no parameter need to be set in advance for the ULDA and OLDA schemes, and only one parameter, γ, need to set for the KUDA, eKUDA, KODA, and eKODA schemes. However, for the KPCA, eKPCA, KDDA, and eKDDA schemes, an extra parameter, ε, is introduced to avoid the numerical instability caused by the tiny eigenvalues. The tiny eigenvalue λ is considered to be zero, if $\frac{\lambda}{\lambda_{max}} \leq \varepsilon$, where λ_{max} denotes the maximum eigenvalue. We select the parameter γ from set {10^{-5}, 10^{-6}, 10^{-7}, 10^{-8}, 10^{-9}, 10^{-10}}, and the parameter ε from set {10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}, 0}. For the KDDA and eKDDA schemes, the final projection dimension q, where $q \leq m - 1$, still needs to be pre-specified. However, to avoid setting too many parameters, especially, for the KDDA scheme, we usually fix q at $m - 2$. In the experiments, we implement the KDDA scheme using the Matlab code written by Lu, which is available for downloading at http://www.dsp.utoronto.ca/juwei/juwei_pubs.html). However, for the sake of fairness in the comparisons, the regularization constant, "Eta_sw", in Lu's KDDA code is set to zero, since no other scheme employs the regularization technique to further improve performance.

After data are mapped to the different projection spaces, the nearest neighbor (NN) classifier is employed to classify the sample images, and the classification accuracy on test samples are used to evaluated the performances of various learning machines.

p	0.2	0.3	0.4	0.5	0.6
PCA	57.59±4.91	64.62±3.01	67.26±4.32	68.92±4.11	72.12±4.85
KPCA	58.17±4.71	64.92±2.98	67.69±4.17	69.22±4.13	72.37±4.73
eKPCA	58.13±4.69	64.90±2.95	67.67±4.15	69.19±4.12	72.42±4.67
KDDA	49.46±6.14	69.15±3.67	74.21±4.35	76.69±4.04	81.00±4.43
eKDDA	63.33±3.87	74.94±3.79	77.45±3.59	82.53±3.68	86.46±3.60
ULDA	70.63±3.73	79.60±3.10	81.38±5.33	83.94±4.81	86.54±5.34
KUDA	68.74±6.57	79.69±2.93	76.29±15.86	74.08±21.16	72.88±24.09
eKUDA	**71.63±3.26**	**80.54±2.99**	**83.05±4.34**	85.44±4.36	88.58±4.46
OLDA	66.67±3.74	77.65±3.55	81.90±4.00	84.92±2.81	87.00±4.79
KODA	63.20±4.01	75.35±3.46	79.00±3.92	82.08±3.08	87.33±4.23
eKODA	67.19±3.75	78.21±3.44	82.55±3.95	**85.78±2.65**	**88.96±3.92**

Table 2. Experimental results in terms of the average values and the standard deviations of the best recognition accuracy (%) on test data for the Yale data set

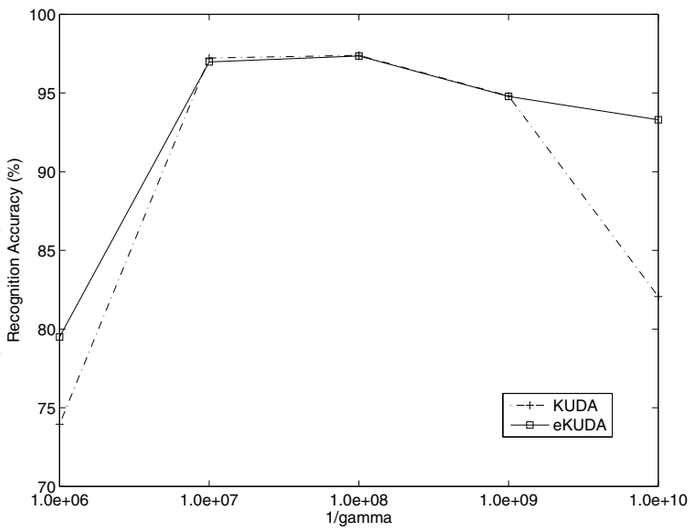
p	0.5	0.6	0.7	0.8	0.9
PCA	58.48±4.48	64.52±4.15	69.25±5.69	73.04±5.69	78.33±8.62
KPCA	59.05±4.40	65.06±4.03	70.08±5.54	73.69±6.31	79.52±8.50
eKPCA	58.93±4.38	65.06±4.04	70.04±5.54	73.69±6.31	79.40±8.45
KDDA	61.57±5.32	65.51±4.36	68.33±5.20	70.77±7.15	73.33±8.68
eKDDA	69.48±5.00	73.87±4.51	77.06±4.97	79.46±4.96	86.55±7.54
ULDA	70.62±4.71	74.37±4.41	77.34±5.05	79.70±5.39	85.71±7.78
KUDA	71.69±4.50	75.71±4.12	79.25±5.00	82.74±5.08	88.07±6.82
eKUDA	71.83±4.48	75.95±4.23	79.44±5.01	83.04±4.80	88.10±7.23
OLDA	72.14±5.31	76.82±5.09	78.97±5.36	82.38±5.80	87.74±7.46
KODA	73.50±5.03	**78.42±5.09**	80.55±5.74	85.24±5.32	89.52±6.92
eKODA	**73.62±4.72**	78.07±5.01	**81.03±5.26**	**85.36±4.84**	**90.12±6.23**

Table 3. Experimental results in terms of the average values and the standard deviations of the best recognition accuracy (%) on test data for the JAFFE data set

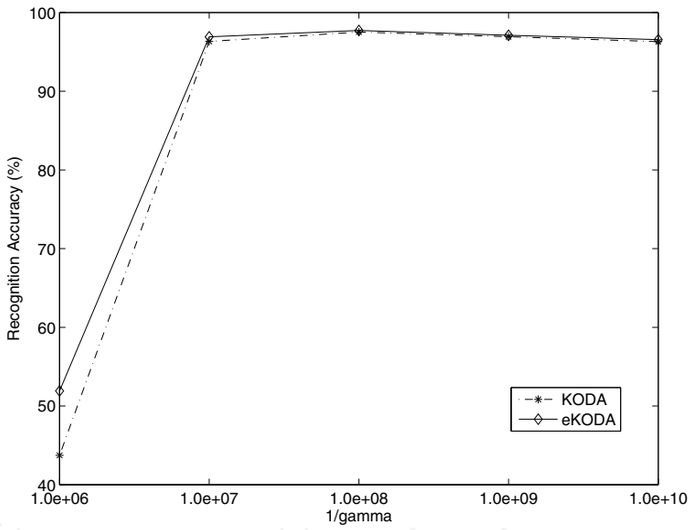
5.1 Experiment on face recognition

In this experiment, we compare the empirical kernel machines with the kernel-trick-base kernel machines and the linear machines in the application of face recognition. The experiment is carried out on two face image database, the ORL and Yale database. The ORL data contain 40 persons, each having 10 different images of size 92×112 with the variation to a certain extent in pose and scaling, and the Yale data we used includes 15 individuals, each having 10 pictures (cropped to size 112×112) with different facial expressions and illuminations, wearing or without wearing glasses. The samples of each subject are randomly divided to two disjoint subsets, one is used as the training data, and the other the test data. The ratio of the training data number to the total sample number per class (individual), called training rate, is denoted by p.

We investigate the performances of different machines with different values of p. The best value of the recognition accuracy on the test data over different parameter settings is used to evaluate the performances of different algorithms. The experiment is repeated 40 times, and the experimental results in terms of the average values and the standard deviations of the recognition accuracy on test data are shown in Table 1, for the ORL data, and Table 2, for



(a)



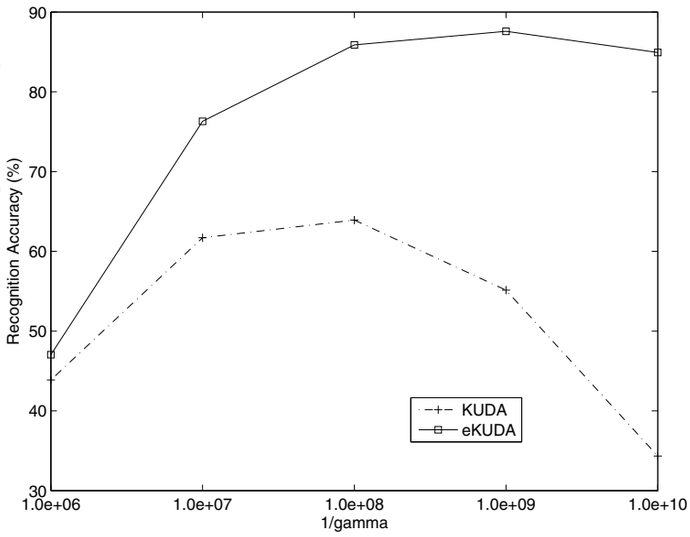
(b)

Fig. 2. Performance comparisons of (a) KUDA vs. eKUDA, and (b) KODA vs. eKODA on ORL data, with different parameter γ settings.

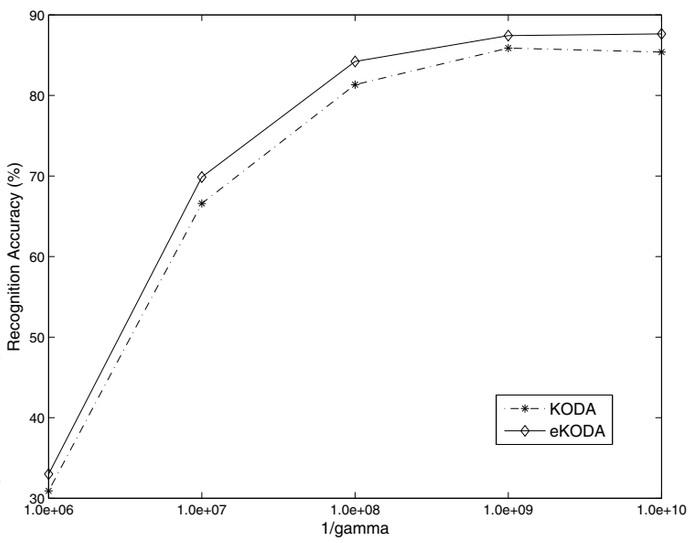
the Yale data. The best results under different training rates (p) are shown in boldface in the tables.

We also compare the performances of two pairs of kernel machines, namely, KUDA vs. eKUDA, and KODA vs. eKODA, when their unique parameter γ is set to different values. Fig.(2) (a) (b) illustrate the average test recognition accuracy (%) as a function of $1/\gamma$ on the ORL data set, where the training rate is set at $p = 0.6$. The corresponding result on the Yale data set is presented in Fig.(3)

The experimental results in Tables 1 and 2 and Figs.(2)(3) lead to following points:



(a)



(b)

Fig. 3. Performance comparisons of (a) KUDA vs. eKUDA, and (b) KODA vs. eKODA on Yale data, with different parameter γ settings.

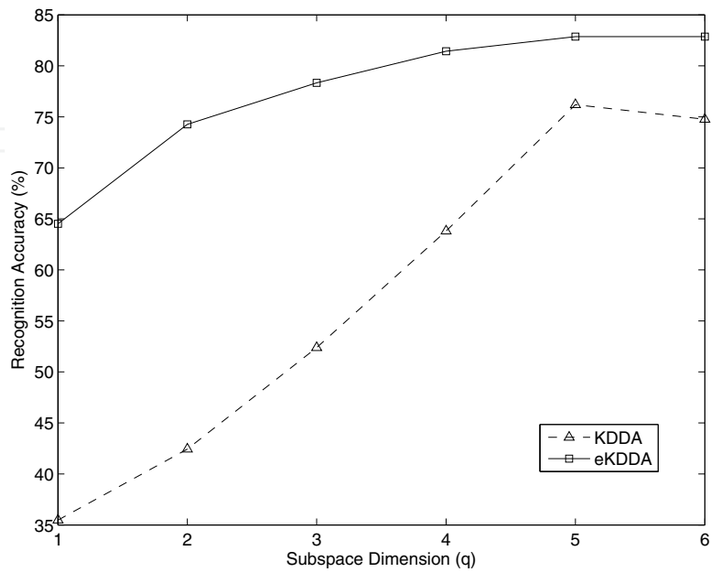
p	0.2	0.3	0.4	0.5	0.6
PCA	82.54±2.31	83.25±2.37	84.30±1.73	87.30±2.13	88.53±2.39
KPCA	83.33±2.21	84.25±1.99	85.24±1.67	87.97±1.86	89.41±1.89
eKPCA	83.33±2.21	84.25±1.99	85.22±1.66	88.00±1.85	89.41±1.89
KDDA	82.92±2.46	84.45±2.04	83.62±1.87	87.10±1.80	87.45±2.09
eKDDA	83.84±2.59	85.82±1.71	86.08±2.07	88.11±2.13	90.23±2.20
ULDA	60.80±4.00	52.75±4.95	46.60±3.60	40.99±4.11	29.18±3.04
KUDA	83.91±2.99	86.85±2.54	88.16±1.76	90.20±2.11	92.19±1.87
eKUDA	86.16±2.00	88.44±1.98	89.45±1.41	90.65±1.98	92.78±1.73
OLDA	67.32±3.50	59.85±3.52	57.60±3.06	50.84±3.81	37.05±3.63
KODA	83.22±2.62	85.93±2.22	86.98±1.66	88.45±2.13	89.37±2.24
eKODA	**86.74±2.20**	**89.12±1.97**	**89.75±1.33**	**91.09±1.66**	**92.91±1.90**

Table 4. Experimental results in terms of the average values and the standard deviations of the best recognition accuracy (%) on test data for the USPS data set

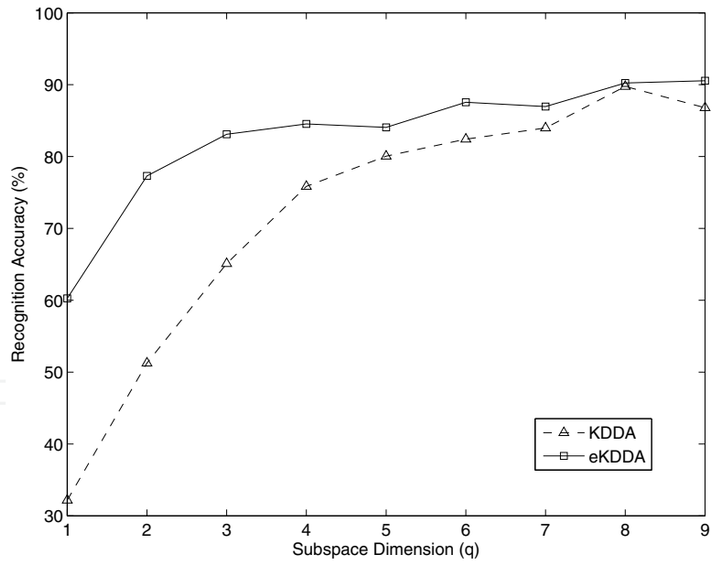
1. Empirical kernel machines achieve the best results in most cases.
2. Empirical kernel PCA performs almost the same as the conventional KPCA, which may suggests that the Eq.(2) holds or approximately holds in practices.
3. Lu's KDDA scheme works better than eKDDA in two cases on the ORL data. However, on the Yale data set, where the within-class scatter measure is much larger than that of the ORL data due to the variations of illumination, the eKDDA scheme performs much better than the KDDA scheme.
4. For the SVD-based discriminant analysis schemes, either ULDA, OLDA, or their kernel counterparts, they usually outperform the PCA schemes and the direct-LDA schemes. Moreover, while the KUDA and KODA schemes work better than their linear counterparts on the ORL data, their performances degenerate remarkably on the Yale data, especially for KUDA. However, in either case, our eKUDA and eKODA work well, and lead to most best results.

5.2 Experiment on facial expression recognition

We investigate the efficiency of our empirical kernel machines in the application of facial expression recognition, and compare their performances with those of the other pattern classification methods. Compared with face recognition, the facial expression recognition is a more challenging classification task, since the between-class discrimination among different facial expression patterns is much smaller than the within-class discrimination of the expression patterns. In this experiment, we use the JAFFE facial expression database to test and evaluate various algorithms. The JAFFE data set is a widely-used database for facial expression recognition. It contains ten Japanese women's face images with 7 typical facial expressions (angry, disgust, fear, happy, sad, surprise, and neutral), each expression having three different pictures, which are cropped to size 128×128. Since facial expression recognition is a difficult classification task, the training rate p is set to a relatively large value. The experimental results are shown in Table 3 in terms of the average best recognition accuracy on test data over 40 trails , corresponding to the training rate $p = 0.9, 0.8, 0.7, 0.6,$ and 0.5, respectively. Furthermore, we also compare the performances of KDDA and eKDDA



(a)



(b)

Fig. 4. Comparison of the performances of the KDDA and eKDDA schemes on, (a) the JAFFE data, and (b) the USPS data, under different projection dimension q

with different projection dimension q (in the previous experiments, we fix q at $m-2=5$), as the training rate p is at level 0.9. Fig.(4) (a) shows the results.
It can be seen that, for the facial expression recognition on the JAFFE data set, 1)the eKDDA scheme remarkably outperforms the KDDA scheme; 2)the orthogonal discriminant analysis schemes perform better than the uncorrelated disciminant analysis schemes, either in OLDA vs. ULDA, KODA vs. KUDA, or eKODA vs. eKUDA, and furthermore, the eKODA scheme achieves the best results in all cases except $p = 0.6$.

5.3 Experiment on handwritten digit recognition

To test our algorithms in a wide-range of applications, we conduct experiment for handwritten digit recognition using the USPS data. The USPS handwritten digit data set, available for downloading at http://www.csie.ntu.edu.tw/ ~cjlin/libsvmtools/datasets/, is widely used as a benchmark for evaluating various learning methods. It contains more then 7 thousands training samples and two thousands test samples of handwritten digits from 0 to 9. Each sample is represented by an 16×16 image.
Since our goal in this experiment is focused at comparing different classification algorithms, to reduce the computational burden, we randomly select 800 samples, 80 samples per class, from the training set of the USPS data to form our experiment data set. Considering the data dimension in this experiment is much smaller than that of the data used in other experiments, we choose the value of the parameter γ from $\{10^0, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\}$, and the parameter ε from $\{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 0\}$. Table 4 gives the experimental results in terms of the average values of the best recognition accuracy on test data over 40 trails , corresponding to the training rate $p = 0.2, 0.3, 0.4, 0.5$, and 0.6, respectively. Furthermore, to compare the performances of the KDDA and eKDDA schemes under different projection dimension q (in the previous experiments, we always set $q = m-2$), we illustrate the average test recognition accuracy (%) as a function of q in Fig.(4) (b), where the training rate is set at $p = 0.6$.
From Table 4 and Fig.(4), it is easy to see that the eKODA scheme achieves the best recognition results in all cases, and eKDDA performs substantially better than KDDA. Moreover, a big difference between Table 4 and other tables is that the linear versions of the SVD-based discriminant analysis, i.e., ULDA and OLDA, perform surprisingly worse than other methods this time. However, their kernel nonlinear versions still work well, especially, the empirical kernel versions.

6. Conclusion

We have presented a new way to "seamlessly" kernelize linear machines. The empirical kernel feature space, a finite-dimensional embedding space, in which the distances of the data in the kernel feature space are preserved, provides a unified framework for the kernelization. This method is different from the conventional kernel-trick based kernelization, and more importantly, the final empirical kernel machines performs more efficiently in many real-world applications, such as face recognition, facial expression recognition, and handwritten digit identification, than the kernel-trick based kernel machines.

7. Acknowledgements

We would thank Dr. Lu and Dr. Ye for providing Matlab codes for the implementation of the KDDA, ULDA, and OLDA algorithms. This work was supported by the National Natural Science Foundation of China (grant no. 60775008) and the National High Technology Research and Development Program (863 Program) of China (grant no. 2007AA01Z196).

8. References

V. Vapnik. *The Nature of Statistical Learning Theory*, Mew York:Spring, 1995.

B. Scholkopf, A.J. Smola, and K.-R. Muller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, vol.10, pp.1299–1319, 1998.

S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K.-R. Müller. Fisher discriminant analysis with kernel. *Proc. IEEE Int'l Workshop Neural Networks for Signal Processing IX*, pp.41-48, Aug. 1999.

S. Mika, G. Rätsch, B. Schölkopf, A. Smola, J. Weston, and K.-R. Müller. Invariant feature extracion and classification in kernel space. *Advances in Neural Information Processing Systems*, 12, Cambridge, Mass.:MIT Press, 1999.

M.H. Yang. Kernel eigenfaces vs. kernel Fisherfaces: Face recognition using kernel methods. *Proc. Fifth IEEE Int'l Conf. Automatic Face and Gesture Recognition*, pp.215-220, May 2002.

J. Lu, K.N. Plataniotis, and A.N. Venetsanopoulos. Face recognition using kernel direct discriminant analysis algorithms. *IEEE Trans Neural Networks*, vol.14, no.1, 2003, pp.117-126.

J. Yang, A.F. Frangi, and J.-Y. Yang. A new kernel Fisher disciminant algorithm with application to face recognition. *Neurocomputing*, vol.56, pp.415-421, 2004.

J. Ye, T. Li, T. Xiong, and R. Janardan. Using uncorrelated discriminant analysis for tissue classification with gene expression data. *IEEE/ACM Trans. Computational Biology and Bioinformatics*, vol.1, no.4, 2004, pp.181-190.

J. Ye. Characterization of a family of algorithms for generalized discriminant analysis on undersampled problems. *Journal of Machine Learning Research*, vol.6, 2005, pp.483-503.

H. Xiong, M.N.S. Swamy, and M.O. Ahmad. Optimizing the kernel in the empirical feature space. *IEEE Trans. Neural Networks*, vol.16, no.2, 2005, pp.460-474.

B. Schölkopf, B. Mika, C.J.C. Burges, P. Knirsch, K.-R. Müler, G. Rätsch, and A.J. Smola. Input space versus feature space in kernel-based methods. *IEEE Trans. Neural Networks*, vol.10, no.5, 1999, pp.1000-1017.

L.F. Chen, H.Y.M. Liao, M.T. Ko, J.C. Lin, and G.J. Yu. A new LDA-based face recognition system which can solve the small sample size problem. *Pattern Recognition*, vol.33, 2000, pp.1713-1726.

H. Yu and J. Yang. A Direct LDA Algorithm for High-Dimensional Data–with Application to Face Recognition. *Pattern Recognition*, vol.34, no.10, pp.2067-2070, 2001.

P.N. brlhumeour, J.P. Hespanha, and D.J. Kriegman. Eigenfaces vs. Fisherfaces: Recognition using class specific projection. *IEEE Trans. on Pattern Analysis and machine Intelligence*, vol.19, no.7, pp.711-720, 1997.

M.W. Berry, S.T. Dumaism, and G.W. O'Brie. Using linear algebra for intelligent information retrieval. *SIAM Review*, vol.37, pp.573-595, 1995.

S. Dudoit, J. Fridlyand, and T.P. Speed. Comparison of discriminant methods for the classification of tumors using gene expression data. *Journal of the American Statistical Association*, vol.97, pp.77-87, 2002.

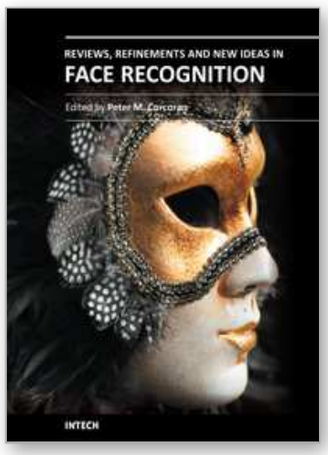
S. Raudys and R.P.W. Duin. On expected classification error of the Fisher linear classifier with pseudo-inverse convariance matrix. *Pattern Recognition Letters*, vol.19, no.5-6, 1998, pp.385-392.

K. Fukunaga. *Introduction to Statistical Pattern Recognition*, second edition. Academic Press, 1990.

S. Ji and J. Ye. Kernel uncorrelated and regularized discriminant analysis: A theoretical and computational study. *IEEE Trans. on Knowledge and Data Engineering*, vol.20, no.10, pp.1311-1321, 2008.

J. J. Hull. A database for handwritten text recognition research. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.16, no.5, pp.550-554, 1994.

M.J. Lyons, S. Akamatsu, M. Kamachi, J. Gyoba. Coding Facial Expressions with Gabor Wavelets. *Proceedings of Third IEEE International Conference on Automatic Face and Gesture Recognition*, Nara, Japan, pp. 200-205, April 14-16 1998.



Reviews, Refinements and New Ideas in Face Recognition

Edited by Dr. Peter Corcoran

ISBN 978-953-307-368-2
Hard cover, 328 pages
Publisher InTech
Published online 27, July, 2011
Published in print edition July, 2011

As a baby one of our earliest stimuli is that of human faces. We rapidly learn to identify, characterize and eventually distinguish those who are near and dear to us. We accept face recognition later as an everyday ability. We realize the complexity of the underlying problem only when we attempt to duplicate this skill in a computer vision system. This book is arranged around a number of clustered themes covering different aspects of face recognition. The first section on Statistical Face Models and Classifiers presents reviews and refinements of some well-known statistical models. The next section presents two articles exploring the use of Infrared imaging techniques and is followed by few articles devoted to refinements of classical methods. New approaches to improve the robustness of face analysis techniques are followed by two articles dealing with real-time challenges in video sequences. A final article explores human perceptual issues of face recognition.

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen