Accelerated Kernel CCA plus SVDD: A Three-stage Process for Improving Face Recognition

Ming Li
School of Computer and Communication, Lanzhou University of Technology, Lanzhou, China
Email:lim3076@163.com

Yuanhong Hao
School of Computer and Communication, Lanzhou University of Technology, Lanzhou, China
Email: haoren313@163.com

***Abstract*—kernel canonical correlation analysis (KCCA) is a recently addressed supervised machine learning methods, which shows to be a powerful approach of extracting nonlinear features for face classification and other applications. However, the standard KCCA algorithm may suffer from computational problem as the training set increase. To overcome the drawback, we propose a three-stage method to improve the performance of KCCA. Firstly, a scheme based on geometrical consideration is proposed to enhance the extraction efficiency. The algorithm can select a subset of samples whose projections in feature space (Hilbert space) are sufficient to represent all of the data in feature space. Subsequently, an improved algorithm inspired by principal component analysis (PCA) is developed. The algorithm can select the most contributive eigenvectors for training and classification instead of considering all the ones. Finally, a multi-class classification method based on support vectors data description (SVDD) is employed to further enhance the recognition performance as it can avoid the repeated use of training data. The theoretical analysis and the experiment results demonstrate the effectiveness of improvements.**

***Index Terms*—face recognition, kernel canonical correlation analysis, feature vector selection (FVS), support vectors data description (SVDD)**

I. INTRODUCTION

Recently many kernel-based machine learning methods have attracted more and more attentions. The kernel methods first map the nonlinearly separable input data into a high dimensional linearly separable feature space (Hilbert space) H with a nonlinear mapping ϕ, and then the dot product in the feature space H is replaced by appropriate kernel functions, with which the explicit nonlinear mapping ϕ process is not required to know at all. Using appropriate kernel functions, the kernel methods show better classification performance than linear machines and standard multivariate analysis. Many kernel machines represented by support vector machines [1], kernel-based discriminant algorithms and kernel-based clustering algorithms [2, 3] have been widely successfully used for nonlinear estimation and pattern recognition [4, 5, 6, and 7]. Nevertheless, this study focuses on the latest kernel machines based on kernel canonical correlation analysis (KCCA) [8, 9] and support vectors data description (SVDD) [10, 11] and their applications in face recognition.

As is true of other kernel machines, the standard KCCA has to store and calculate the kernel matrix $\mathbf{K}$, the size of which is the square of the number of samples, in other words, the computational complexity increases with the sample number. Thus the conventional computation method involves a time complexity of $O(n^3)$, where n is the number of face image samples. Therefore, implementing KCCA on large datasets is time-consuming. In order to overcome this problem, we propose two-fold methods to accelerate KCCA-based feature extraction. Firstly, a geometry-based feature vector selection scheme is proposed to be adopted before extraction. Secondly, an efficient algorithm is developed to improve both of the training and extraction procedures. The theoretical analysis and the results of experiments show that our method is faster than previous approaches. What is more important, it is capable of maintaining close approximation to the ground truth given by the standard KCCA computation method. Moreover, we are particularly interested in the application of SVDD which is first proposed by Tax and Duin [10] and then generalized to solve multi-classification problem by Zhang et al [11] and Kang et al [12]. Adopting SVDD-based multi-class classifier, we can use only the data of one class for training each classifier without reusing training data. This is significantly meaningful for reducing the computational complexity.

The rest of this paper is organized as follows. We give a brief description of KCCA in Section II. Then the improved KCCA based on FVS algorithm is presented in Section III followed by an accelerated algorithm developed in Section IV. In Section V a KCCA combining with SVDD-based multi-classifier scheme for face recognition proposed. In Section VI we provide the experimental results. Finally, the conclusion is presented in section VII.

II. Theoretical Background

A. Nonlinear Feature Extraction via Kernel CCA

Given two centered random multivariables $\boldsymbol{x} \in \mathbb{R}^{m_x}$ and $\boldsymbol{y} \in \mathbb{R}^{m_y}$, the goal of CCA is to determine a pair of direction vectors $\boldsymbol{w_x}$ and $\boldsymbol{w}_y$ to maximize the correlation between the two projections $\boldsymbol{w}_x^T \boldsymbol{x}$ and $\boldsymbol{w}_y^T \boldsymbol{y}$ [8]. More formally, CCA maximizes the function [9]:

$$\begin{aligned}\rho &= E[\boldsymbol{xy}] \Big/ \sqrt{E[\boldsymbol{x}^2]E[\boldsymbol{y}^2]} \\ &= E[\boldsymbol{w}_x^T \boldsymbol{xy}^T \boldsymbol{w}_y] \Big/ \sqrt{E[\boldsymbol{w}_x^T \boldsymbol{xxw}_x]E[\boldsymbol{w}_y^T \boldsymbol{yy}^T \boldsymbol{w}_y]}\end{aligned} \tag{1}$$

For the basis vectors $\boldsymbol{w_x}$ and $\boldsymbol{w}_y$ of CCA, the following Theorem 1 holds.

Theorem 1 The basis vectors $\boldsymbol{w_x}$ and $\boldsymbol{w}_y$ of CCA can be expressed as the linear combination of the training samples $\{\boldsymbol{X}_i\}_{i=1}^n$ and $\{\boldsymbol{Y}_i\}_{i=1}^n$, respectively. In other words, $\boldsymbol{w_x} = \sum_{i=1}^n \boldsymbol{\alpha}_i \boldsymbol{X}_i = \boldsymbol{X\alpha}$ and $\boldsymbol{w}_y = \sum_{i=1}^n \boldsymbol{b}_i \boldsymbol{Y}_i = \boldsymbol{Yb}$ hold simultaneously, where $\boldsymbol{\alpha}, \boldsymbol{b} \in \mathbb{R}^n$ denote corresponding combination coefficient vectors (The proof is similar to that of [9] thus omitted here.)

As a nonlinear method, KCCA is one approach of generalizing linear CCA into nonlinear case using the kernel method. We assume that $\boldsymbol{X} = (\boldsymbol{x}_{ij})^T$ denotes a set of face image sample, where $\mathbf{x}_{ij} \in R^d$ means that it belongs to i th class and j th sample means that it belongs to i th class and j th sample. Then we can construct the following class-membership matrix according to the defined set of face image sample [13]:

$$\mathbf{Y} = \begin{bmatrix} 1_{n_1} & 0_{n_1} & \dots & 0_{n_1} \\ 0_{n_2} & 1_{n_2} & \dots & 0_{n_2} \\ \vdots & \cdots & \ddots & 1_{n_{C-1}} \\ 0_{n_C} & 0_{n_C} & \cdots & 1_{n_C} \end{bmatrix}_{N \times (c-1)} \tag{2}$$

where 1_{n_i} is a $n_i \times 1$ column vector containing all ones which means that n_i samples belong to the i th class. In the same way, 0_{n_i} is a $n_i \times 1$ column vector containing all zeros. If one sample is belonging to certain class, the corresponding element in the class-membership matrix will be set one, otherwise zero.

Here we define the Gram matrices

$$\boldsymbol{K} = (k(x_i, x_j))_{i=1,\dots,n; j=1,\dots,n} = (\phi_x(x_i))^T \phi_x(x_j) \tag{3}$$

Thus, KCCA can be represented as the following optimization problem:

$$\arg\max_{a,b} \boldsymbol{a}_i^T \boldsymbol{X}_\phi^T \boldsymbol{Yb}_i \tag{4}$$

$$\text{Subject to } \boldsymbol{a}_i^T \boldsymbol{X}_\phi^T \boldsymbol{X}_\phi \boldsymbol{a}_i = \boldsymbol{b}_i^T \boldsymbol{Y}^T \boldsymbol{Yb}_i = 1 \tag{5}$$

According to Theorem 1, there exists column vector $\boldsymbol{a}_i = (\alpha_1, \dots, \alpha_n)^T$, such that

$$\boldsymbol{a}_\phi = \boldsymbol{\alpha}^T \boldsymbol{X}_\phi^T = \sum_{i=1}^n \boldsymbol{X}_\phi^T a_i \tag{6}$$

From Eq. (5) and Eq. (6), we obtain

$$\max_{a,b_i}(r(\boldsymbol{a}_i, \boldsymbol{b}_i)) = \boldsymbol{a}_i^T \boldsymbol{X}_\phi^T \boldsymbol{X}_\phi \boldsymbol{Yb}_i = \boldsymbol{a}_i^T \boldsymbol{KYb}_i \tag{7}$$

$$\text{Subject to } \boldsymbol{a}_\phi^T \boldsymbol{X}_\phi^T \boldsymbol{X}_\phi \boldsymbol{a}_\phi = \boldsymbol{b}_i^T \boldsymbol{Y}^T \boldsymbol{Yb}_i = 1 \tag{8}$$

Adopting the methods in [14] to solve the KCCA problem, the following generalized eigenvalue equation is obtained:

$$\widetilde{\boldsymbol{K}}^* \boldsymbol{Y}(\boldsymbol{Y}^T \boldsymbol{Y})^{-1} \boldsymbol{Y}^T \boldsymbol{Ka}_i = \lambda^2 \boldsymbol{a}_i \tag{9}$$

where $\widetilde{\boldsymbol{K}}^*$ is the generalized inverse matrix of $\boldsymbol{K}$. The rank of $\widetilde{\boldsymbol{K}}^*$ is n and the rank of $\boldsymbol{Y}^T\boldsymbol{Y}$ is $c-1$. Therefore, the eigenvectors of $a_1, a_2, \dots a_{c-1}$ will be obtained corresponding to the $c-1$ eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_{c-1}$. Given a new sample $\boldsymbol{Z} = (\boldsymbol{z}_1, \boldsymbol{z}_2, \dots, \boldsymbol{z}_n)^T$, the following extraction equation can be obtained:

$$\begin{aligned} feature(\boldsymbol{z}) &= \boldsymbol{a}_\phi \boldsymbol{Z}_\phi = (a_1, a_2, \dots, a_{c-1})^T \boldsymbol{X}_\phi^T \boldsymbol{Z}_\phi \\ &= (a_1, a_2, \dots, a_{c-1})^T \boldsymbol{K}_z \end{aligned} \tag{10}$$

where $\boldsymbol{K}_z = (k(\boldsymbol{x}_1, \boldsymbol{z}), k(\boldsymbol{x}_2, \boldsymbol{z}), \dots, k(\boldsymbol{x}_i, \boldsymbol{z}))^T (i = 1,2,\dots,n)$ and k is the kernel function.

B. Computational Load Analysis of the Kernel CCA

This section concerns the computation complexity of kernel CCA. In the training stage of kernel CCA, it requires to store and manipulate the kernel matrix $\boldsymbol{K}$, the size of which is $n \times n$, where n denotes the size of training set. Moreover, the complexity of diagonalizing the kernel matrix $\boldsymbol{K}$ is $O(n^3)$. When the sample number becomes large, the kernel CCA suffers from computational problem. In next two sections, we will propose two efficient algorithms to overcome the drawback of KCCA as well as to reduce the computation complexity.

III. Improved Kernel CCA

A. Feature Vector Selection

According to Eq. (6), all the training samples in $\boldsymbol{F}$ are used to represent eigenvectors $\boldsymbol{a}_\phi$. Actually, the dimensionality of the subspace spanned by $\boldsymbol{\Phi}(\boldsymbol{X})$ is just equal to the rank of kernel matrix $\boldsymbol{K}$. Usually, the rank of $\boldsymbol{K}$ is not more than n, that is to say, rank ($\boldsymbol{K}$) <n.

If a basis of the feature vectors $\{\phi(\boldsymbol{x}_{a_i})\}(a_j = 1, 2, \dots, rank(\boldsymbol{K}))$ is known, we can rewrite Eq. (6) as follows:

$$\boldsymbol{a}_\phi = \sum_{j=1}^{rank(K)} a_j \phi(\boldsymbol{x}_{a_j})^T \tag{11}$$

It will greatly improve the computational efficiency. This is because $rank(\boldsymbol{K}) << n$.

In this paper, we employ an algorithm based on geometrical approach [15, 16] to select such two bases of the feature vectors in feature space $\boldsymbol{F}$. The goal is to find a subset of the samples whose projections in the transformed space $\boldsymbol{F}$ are sufficient to express all of the data in $\boldsymbol{F}$ as a linear combination of them.

For a given set of selected vectors $\boldsymbol{S} = \{\boldsymbol{x}_{S_1}, \boldsymbol{x}_{S_2} \dots, \boldsymbol{x}_{S_L}\}$, where L denotes the number of Feature Vectors, the estimation of the mapping of any sample $\boldsymbol{x}_i$ is regarded as

---

✧ **Input:**

1) Training sample set $\{x_i\} \in \mathbb{R}^{m\times n}(i=1,...,n)$;

2) Kernel function and corresponding parameters.

✧ **Initialization:** $\boldsymbol{S}=\varnothing$, L=0

First iteration:

Select one sample in $\boldsymbol{x}_i$ that maximizes $\boldsymbol{J}_S$, and add it into $\boldsymbol{S}$, set L=1.

✧ **Do iteration procedure**

Combining with previous L samples, select one sample remaining samples in $\boldsymbol{x}_i$.

If $\boldsymbol{K}_{SS}$ is invertible

Append the sample which can maximizes $\boldsymbol{J}_S$ into $\boldsymbol{S}$, and set L=L+1.

Else

Stop iteration.

End if.

End iteration;

Output $\boldsymbol{S}$ as the selected sample set.

Figure 1. Feature vector selection algorithm

linear combinations of $\boldsymbol{S}$, which can be written as follows:

$$\hat{\phi}_i = \phi_S \cdot \boldsymbol{a}_i\ , \tag{12}$$

where $\phi_S=(\phi_{S_1},\phi_{S_2},...,\phi_{S_L})$ is the mapping matrix of the selected samples in $\boldsymbol{F}$ and $\boldsymbol{a}_i=(a_{i1},...,a_{iL})^T$ is the coefficient vectors. Thus the goal becomes how to find the coefficient vectors $\boldsymbol{a}_i$, such that the estimated mapping $\hat{\phi}_i$ is as close to the corresponding real mappings ϕ as possible. This can be achieved by minimizing the normalized Euclidean distance as the following equations:

$$\delta_i = \|\phi_i - \hat{\phi}_i\|^2 / \|\phi_i\|^2 \tag{13}$$

Let $\partial\delta_i/\partial\boldsymbol{a}_i=0$ then rewrite it in matrix form, we obtain

$$\min(\delta_i) = 1-\boldsymbol{K}_{Si}^T\boldsymbol{K}_{SS}^{-1}\boldsymbol{K}_{Si}/k_{ii}\ , \tag{14}$$

where $k_{ii}=k(\boldsymbol{x}_i,\boldsymbol{x}_i)$, $\boldsymbol{K}_{S_xS_x}=(k(\boldsymbol{x}_{Sp},\boldsymbol{x}_{Sq}))_{1\le p\le L,1\le q\le L}$ is a square matrix of inner products of the selected samples, and $\boldsymbol{K}_{S_xi}=(k(\boldsymbol{x}_{Sp},\boldsymbol{x}_i))_{1\le p\le L}$ is matrix of dot product between $\boldsymbol{x}_i$ and the selected set $\boldsymbol{S}$.

Let Eq. (14) satisfy all the samples, thus we can obtain:

$$\max \boldsymbol{J}_S = \frac{1}{n}\sum (\boldsymbol{K}_{Si}^T\boldsymbol{K}_{SS}^{-1}\boldsymbol{K}_{Si}/k_{ii}) \tag{15}$$

The problem can be solved via an iterative process, and the process stops when $\boldsymbol{K}_{SS}$ is no longer invertible numerically, which means that $\boldsymbol{S}$ is good approximation of basis for the data in $\boldsymbol{F}$ [16]. Moreover, the feature vectors selection process [17] is outlined in Fig.1.

B. Kernel CCA plus Feature Vector Selection

Given training samples with a large number n, FVS algorithm is employed to reduce the sample size to L ($L<n$), and then the computational load of training procedure as well as the feature extraction procedure.

After performing FVS, kernel CCA is adopted for face recognition. Then Eq. (6) and Eq. (10) can be respectively rewritten as follows:

$$\boldsymbol{a}_\phi = \sum_{i=1}^{L} i\cdot\phi(\boldsymbol{x}_{S_i})^T\ (i=1,2,...,L) \tag{16}$$

$$feature(\boldsymbol{z}) = (a_1,a_2,...,a_{c-1})^T\left[k(\boldsymbol{x}_{S_1},\boldsymbol{z}),k(\boldsymbol{x}_{S_2},\boldsymbol{z})...,k(\boldsymbol{x}_{S_L},\boldsymbol{z})\right]^T \tag{17}$$

where L is the number of selected samples.

When a new face image $\boldsymbol{z}_{new}$ is obtained, the features can be obtained via Section II and Eq. (17).

C. Computational Load Analysis of the Kernel CCA plus FVS algorithm

As Section II showed, the original kernel CCA diagonalizes the kernel matrix $\boldsymbol{K}$ with the complexity of $O(n^3)$. However, after FVS algorithm is performed, only L ($L<n$) selected samples are used to train kernel CCA. Since $L<n$, the complexity will be reduced to $O(L^3)$, which is an improvement over the $O(n^3)$ operations required by kernel CCA.

IV THE PROPOSED ACCELERATED ALGORITHM

A. Algorithm for Accelerating Feature Extraction Based on KCCA

Section II has shown that, in order to implement extraction process via standard KCCA, we have to calculate all the kernel functions between this sample and the total training samples, which means that the feature extraction process associated with a training sample set of a large size is quite inefficient. However, just like the principle of PCA, different extractive eigenvectors have dissimilar representative effects, that is to say, small amount of eigenvectors extracted contribute much to representing a face while others contribute less. We call these vectors “representative eigenvectors”. Since the “representative eigenvectors” are fewer than the total eigenvectors and it can involve almost all of the useful information of initial data, we can achieve more efficient feature extraction process. According to the physical meaning of CCA, feature extraction of KCCA must be performed on the eigenvectors of the corresponding eigenvalue equation.

Based on above analysis, we propose to determine “representative eigenvectors” using the following algorithm to train the classifier and speed up the extraction process.

Step1. From all samples and Eq. (9), we can obtain the maximum eigenvalue which is denoted by λ_1. Corresponding eigenvector is denoted by the first “representative eigenvectors”;

Step2. Since λ_1 has been selected, λ_2 can be obtained by maximize $u=\lambda_1+\lambda_2$ based on Eq. (9) and all samples, and corresponding eigenvector is denoted by the second “representative eigenvectors”;

Step3. Suppose $u = \lambda_1 + \lambda_2 + \lambda_3$, since λ_1 and λ_2 has been selected, λ_3 can be obtained by maximize $u = \lambda_1 + \lambda_2 + \lambda_3$ based on Eq. (9) and all samples, and corresponding eigenvector is denoted by the third "representative eigenvectors";

(Following works based on the similar operation above until the l th "representative eigenvectors" is obtained.)

Step l : Suppose that l "representative eigenvectors" are required. And $l-1$ eigenvectors have been obtained. We define a variable u as follows:

If $i \le l$, then $u = \lambda_1 + \lambda_2 + ... + \lambda_i$, otherwise $u = \lambda_1 + \lambda_2 + ... + \lambda_l$,

where $l = n\xi, 0 < \xi \le 1$ and n is the number of face image samples. In order to fix on the optimal ξ , we define the following formula:

$$\xi = \sum_{j=1}^{l} \lambda_j \Big/ \sum_{i=1}^{n} \lambda_i \tag{18}$$

The selection of "representative eigenvectors" will not be finished until $l \ge n\xi$ is satisfied.

By using only the first several "representative eigenvectors" sorted in descending order of the corresponding eigenvalue, the number of eigenvectors can be reduced distinctly.

After performing the above selection algorithm, then Eq. (16) and Eq. (17) can be respectively rewritten as follows:

$$\boldsymbol{a}_\phi = \sum\nolimits_{i=1}^{l} i \cdot \phi(\boldsymbol{x}_{S_i})^T \, (i = 1, 2, ..., l) \tag{19}$$

$$feature(\boldsymbol{z}) = (a_1, a_2, ..., a_l)^T \left[k(\boldsymbol{x}_2, \boldsymbol{z}), k(\boldsymbol{x}_2, \boldsymbol{z}) ..., k(\boldsymbol{x}_l, \boldsymbol{z}) \right]^T \tag{20}$$

where l is the number of selected samples.

When a new face image $\boldsymbol{z}_{new}$ is obtained, the features can be obtained via Section II and Eq. (17).

B. Computational Load Analysis of the New Accelerated Algorithm

In this way, the improved KCCA extracts l eigenvectors, where $l \le L \ll n$. As $O(iL^2)$ extraction operations are required to extract the i -th "representative eigenvectors", the total computational load for l "representative eigenvectors" is $O(lL^2)$, which is an improvement over the $O(n^3)$ operations required by standard KCCA. Thus, after two-stage improved algorithm, the computation complexity will be $\min\{O(lL^2), O(L^3)\}$

V. CLASSIFIERS

A. support vectors data description

The aim of this section is to show the fundamentals of the SVDD-based classifiers, which is used in the research reported below. Meanwhile, in order to reducing training set size requirements for the classification of high-performance, this paper aims to utilize this type of single-class classifier. In terms of training the classifier and realizing class allocations, conventional classification methods are not focused on the training samples of interest. This may result in time and resource being wasted on the training data that is not worthy. SVM-based classifier is everyone's concern in pattern recognition, especially in image classification applications. However, in using the basic SVM approach for pattern classification, it requires training data on all classes (both interest and disinterested), which brings about large computation. This can be replaced by one-class approach, requiring only training data on single class. In addition, in doing so it can reduce the training set size requirements for the classification derived through the use of conventional heuristics without significant loss of accuracy from the expectation [18].

One-class classifiers, which are specialized in identifying a given class and rejecting the others, have been under taken for a variety of applications and have great potential in pattern recognition. From the range of methods available for one-class classifiers, the approach of SVDD based on the principles of the SVM has aroused the greatest interest among the pattern recognition society. Unlike other single-class classifiers such as Gaussian, mixture of Gaussians and k-NN domain description [19], with the SVDD, instead of seeking a hyperplane, the analysis searches for a closed boundary around the data, a hypersphere. The hypersphere is characterized by a center a and radius R . Only training data for the class of interest is needed to be used in the analysis and all inner points are assumed to lie within the radius R of the hypersphere. The aim is to seek the hypersphere with minimum radius R and for that an error function F is formulated as follows,

$$F(R, a) = R^2 \tag{21}$$

subject to $\| x_i - a \|^2 \le R^2, \quad \forall_i$

Since the distances that are greater than R^2 should also be penalized, slack variables $\delta_i \ge 0$ are usually introduced and the minimization problem become

$$F(r, a, \delta) = r^2 + c\sum_i \delta_i \tag{22}$$

with constraints that almost all objects are within the sphere:

$$\| x_i - a \|^2 \le r^2 + \delta_i, \quad \delta_i \ge 0, \forall_i \tag{23}$$

where c is a parameter that gives a balance between the volume of the description and the classification errors.

The constraints (Eq. (20)) can be incorporated into Eq. (19) by using Lagrange multipliers, which ultimately yields

$$L = \sum_i \alpha_i K_2(x_i, x_j) - \sum_{i,j} \alpha_i \alpha_j K_2(x_i, x_j) \tag{24}$$

with the constraint that, $0 \le \alpha_i \le c, \quad \sum_{i=1}^{n} \alpha_i = 1$

where $K_2(x_i, x_j) = (\phi(x_i), \phi(x_j))$.

Figure 2. Exemplary face image in ORL Database are shown. The face images are taken in different illumination conditions.

Now Eq. (24) is in a standard quadratic optimization problem. After optimization, the parameters α_i can be zero or larger than zero.

A few objects z is classified as target object if

$$f_{SVDD}(z) = \| z - \mathbf{a} \|^2 = K_2(z,z) - 2\sum_i \alpha_i K_2(z,x_i) + \sum_i \alpha_i \alpha_j K_2(x_i,x_j) \le r^2 \tag{25}$$

Detailed analysis of SVDD can be available from [10] and [11].

B. Combination with the Improved KCCA and SVDD

An approach to speed up the efficiency of feature extractor based on KCCA has been shown in Section3. Moreover, since the time for the training of schemes using binary classification increase rapidly with the increment of data and class due to the property of SVM, it is not suitable to utilize SVM as classifier in real application. The basic SVM-based classifier requires training data on all classes, while multi-SVDD based classifier only requires data trained on the class of interest. This property enables SVDD approach to be of great potential in pattern recognition and other classified applications. Moreover, the consideration of binary pattern recognition can be done in the solution of separating the class of interest from all others. For instance, there are the one-against-one and one-against-all strategies which both reduce the computational complexity of multi-class classification problem for applying SVMs to multi-class classification [12]. That may be the case for SVDD, too.

Hence, a multi-class classifier based on SVDD is employed. The more detail of SVDD based multi-class classifier can be seen in the reference [11, 12].

Denote the kernel of KCCA by $\boldsymbol{K}_1$ and SVDD by $\boldsymbol{K}_2$. Subsequently, we will discuss how to achieve combining the advantage of improved KCCA and SVDD classifier by following approach.

Step1: Determine kernel K_2 and confirm the size of training samples required

Step2: Utilize FVS algorithm to select the subset of all training samples.

Step3: Then, corresponding direction-vector $\boldsymbol{\alpha}_\phi$ can be obtained from Section II and Eq. (16) and then, the discriminate features obtained via Eq. (17) can be as input data to train the SVDD-based classifier.

Step4: For a new sample $\boldsymbol{z}$, we can determine its feature vector based on the method proposed above. Then, the class of the new sample can be determined by the trained classifier.

VI EXPERIMENTAL RESULTS AND ANALYSIS

We tested the extraction speed and classification accuracy of the proposed method based on ORL face database [20] and compared the proposed method with other methods such as original KCCA with SVDD classifier and original KCCA combining multi-class SVM. ORL face database is composed of 400 gray-scale images of 40 individuals with 10 images of each human. For some human subjects, the images were taken with varying lighting, expressions, with tolerance for some tilting and rotation of up to about 20°. The original face images on ORL database are all size 92×112 with a 256-level gray scale.

We investigated the effection of variations over the numbers of training samples in each class. Firstly, we selected three images per class (individual) for training and the rest images for testing. Thus, the total number of

TABLE I.
COMPARISON OF THE TRAINNING TIME AND MEMORY NEEDED WITH DIFFERENT SAMPLES AND POLYNOMIAL KERNEL (*d*=2) ON ORL DATABASE

Method	Number of Training Samples	Time (s)	Memory Occupancy (MB)	Recognition Accuracy (%)
KCCA plus SVM	120	4.01	62	84.2
	240	5.84	103	90.3
	360	7.13	179	93.7
Improved KCCA plus SVM	120	0.083	19	91.5
	240	0.169	31	93.3
	360	0.232	45	94.5
KCCA plus SVDD	120	0.053	55	92.2
	240	0.097	97	94.6
	360	0.192	165	95.7
Improved KCCA plus SVDD	120	0.019	17	93.3
	240	0.025	29	94.3
	360	0.037	42	95.7

TABLE II.
COMPARISON OF THE TRAINNING TIME AND MEMORY NEEDED AND GAUSSIAN KERNEL ($2\sigma^2=4\times10^4$) ON ORL DATABASE

Method	Number of Training Samples	Time (s)	Memory Occupancy (MB)	Recognition Accuracy (%)
KCCA plus SVM	120	5.34	99	87.1
	240	7.72	145	93.5
	360	8.56	278	96.7
Improved KCCA plus SVM	120	0.53	46	89.1
	240	0.84	61	94.3
	360	1.33	85	95.8
KCCA plus SVDD	120	0.45	107	91.1
	240	0.68	177	94.3
	360	1.33	241	**97.5**
Improved KCCA plus SVDD	120	0.23	42	91.5
	240	0.36	55	93.7
	360	0.73	81	**97.3**

training samples is 40×3=120 and the total number of testing samples is 40×7=280. Secondly, we selected six images per class for training and the rest images for testing, hence we got 40×6=240 training samples. Finally, we totally selected 40×9=360 training samples.

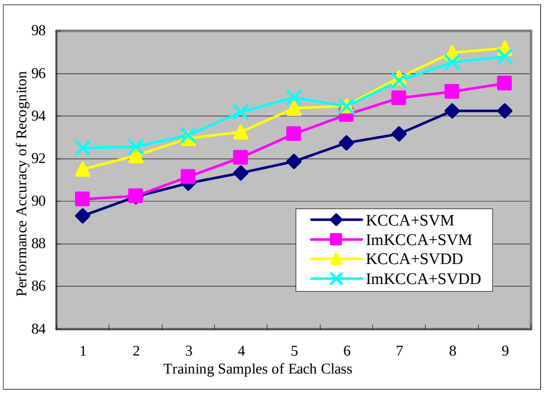

Figure 3. Performance of KCCA and Improved KCCA plus multi-SVM and multi-SVDD classifier on ORL databse.

The comparison of the training time and memory needed of original KCCA and improved KCCA with different training samples and kernel functions on ORL DB is displayed in Table I and Table II. We use polynomial kernel ($k(x,y)=(x\cdot y)^d$) and Gaussian kernel ($k(x,y)=\exp(-|x-y|^2/2\sigma^2)$) for KCCA and polynomial kernel for SVDD (d=2). From the two tables, we can find that the improved KCCA achieves almost the same best results (97.3%) as the original KCCA (97.5%). And the corresponding results of the two KCCA approach are highly alike. Thus the improved KCCA is the approximate version of the standard KCCA. Furthermore, we find that the improved KCCA can largely speed up the training time and save the memory.

Moreover, according to the variations over the numbers of training samples in each class, the average performance accuracy of KCCA and improved KCCA respectively based on multi-SVDD classifier and one-against-one SVM classifier is shown in Fig.3

VII CONCLUSION

This paper describes, analyzes, and demonstrates improved KCCA, a novel two-stage accelerated feature extraction algorithm derived from the feature selection vector scheme and PCA respectively. The time complexity of improved KCCA is $\min\{O(lL^2),O(L^3)\}$ which indicates a significant improvement over the complexity $O(n^3)$ of standard KCCA. Furthermore, the face recognition experiment shows that improved KCCA has the similar classification performance, which demonstrates that the features extracted by improved KCCA are practically useful. Meanwhile, in order to improve the performance of KCCA under small sample size problem, SVDD-based multi-class classifier is proposed for classification after KCCA-based feature extraction. Simulation results based on ORL database are given to show the effectiveness of improved KCCA and SVDD in face recognition.

ACKNOWLEDGMENT

The authors would like to thank the anonymous referees for their careful review. This research was supported by Natural Science Foundation of GanSu Province (No. 2007GS04782). This support is gratefully acknowledged.

REFERENCES

[1] Y. C. Wu, Y. S. Lee, and J. C. Yang, “Robust and efficient multiclass SVM models for phrase pattern recognition” *Pattern Recognition*, vol. 41, pp. 2874-2889, 2008.
[2] J. Wang, K. N. Plataniotis, et al. “Kernel quadratic discriminant analysis for small sample size problem” *Pattern Recognition*, vol. 41, pp. 1528-1538, 2008.
[3] H. A. Boubacar and S. Lecoeuche, S. Maouche. “Self-Adaptive kernel Machine: A new kernel-based algorithm for online clustering” *Neural Networks*, pp. 271–350.
[4] X. Y. Jing, Y. F. Yao, J. Y. Yang. “A novel face recognition approach based on kernel discriminative common vectors (KDCV) feature extraction and RBF neural network” *Neurocomputing*, pp. 1-5, 2007.
[5] K. Saadi, N. L. C. Talbot, G. C. Cawley. “Optimally regularized kernel Fisher discriminant classification”, *Neural Networks*, vol.20, pp. 832-843, 2007.
[6] J. N. Wu, J. Wang, L. Liu. “Feature extraction via KPCA for classification of gait patterns” *Human Movement Science*, vol. 26, pp. 393–411, 2007.
[7] B. Liu, “Adaptive training of a kernel-based nonlinear discriminator” *Pattern Recognition*, vol. 38, pp.2419-2425, 2005.
[8] W. M. Zheng, X. Y. Zhou, et al. “Facial Expression Recognition Using Kernel Canonical Correlation Analysis (KCCA).” *IEEE Trans. Neural Networks*, vol. 17, pp.233-238, 2006.
[9] T. Melzer, M. Reiter, et al. “Appearance models based on kernel canonical correlation analysis”. *Pattern Recognition*. vol.36, pp.1961-1971, 2003.
[10] David M. J. Tax, Robert P. W. Duin. “Support Vector Data Description”. *Machine Learning*. vol. 54, pp.45-66, 2004.
[11] Y. Zhang, Z. X. Chi. “A Fuzzy support vector classifier based on Bayesian optimization” *Fuzzy Optimization Decision Making*, vol. 7, pp. 75-86, 2008.
[12] W. S. Kang, J. H. Na, et al. “Face Recognition System by Fast and Incremental Learning Method” *SICE-ICASE International Joint Conference 2006*, pp.3751-3756, 2006.
[13] C. R. Zou, N. Sun, et al. “2DCCA: A Novel Method for Small Sample Size Face Recognition” *IEEE Workshop on Application of Computer Vision*, pp.43, 2007.
[14] Y. H. He, L. Zhao, C. R. Zou. “Kernel Canonical Correlation Analysis and Application for Face discrimination (in Chinese)” *Chinese Journal of Applied Sciences*. vol. 24, pp.140-144, 2006.
[15] G. Baudat, F. Anouar. “Kernel-based Methods and Function Approximation” *International Joint Conference on Neural Networks*, vol. 2, pp.1244-1249, 2001.
[16] G. Baudat, F. Anouar. “Feature vector selection and projection using kernels” *Neurocomputing*. vol.55, pp.21-38, 2003.
[17] Q. S. Liu, H. Q. Lu, et al. “Improving Kernel Fisher Discriminant Analysis for Face Recognition” *IEEE Trans. Circuits and Systems for Video Technology* vol.14, pp. 42-49, 2004.

[18] G. M. Foody, A. Mathur, et al. “Training set size requirements for the classification of a specific class”, *International Journal of Remote Sensing of Environment*, vol. 104, pp.1-14, 2006.
[19] J. M. Mari. L. Bruzzone, “A Support Vector Domain Description Approach to Supervised Classification of Remote Sensing Images”. *IEEE Trans on Geosciences and Remote Sensing*, vol. 45, No. 8, pp.2683-2692, 2007.
[20] ORL database available from the website: <http://www.uk.research.att.com:pub/data/att_faces.tar.Z>

Ming Li was born in Lanzhou, China, in 1959. He received his B.S. degree in 1982. Now he is a professor in the Department of Computer and Communication at LanZhou University of Technology, Lanzhou, China. His current research interests include image processing, pattern recognition, data mining.

Yuanhong Hao was born in Shanxi, China, in 1982. He received his B.S. degree in 2005. He is currently doing his M.S. in the Department of Computer and Communication at the LanZhou University of Technology, Lanzhou, China. His current research interests include image processing, pattern recognition.