The Kernel Common Vector Method: A Novel Nonlinear Subspace Classifier for Pattern Recognition

Hakan Cevikalp, *Member, IEEE*, Marian Neamtu, and Atalay Barkana

***Abstract*—The common vector (CV) method is a linear subspace classifier method which allows one to discriminate between classes of data sets, such as those arising in image and word recognition. This method utilizes subspaces that represent classes during classification. Each subspace is modeled such that common features of all samples in the corresponding class are extracted. To accomplish this goal, the method eliminates features that are in the direction of the eigenvectors corresponding to the nonzero eigenvalues of the covariance matrix of each class. In this paper, we introduce a variation of the CV method, which will be referred to as the modified CV (MCV) method. Then, a novel approach is proposed to apply the MCV method in a nonlinearly mapped higher dimensional feature space. In this approach, all samples are mapped into a higher dimensional feature space using a kernel mapping function, and then, the MCV method is applied in the mapped space. Under certain conditions, each class gives rise to a unique CV, and the method guarantees a 100% recognition rate with respect to the training set data. Moreover, experiments with several test cases also show that the generalization performance of the proposed kernel method is comparable to the generalization performances of other linear subspace classifier methods as well as the kernel-based nonlinear subspace method. While both the MCV method and its kernel counterpart did not outperform the support vector machine (SVM) classifier in most of the reported experiments, the application of our proposed methods is simpler than that of the multiclass SVM classifier. In addition, it is not necessary to adjust any parameters in our approach.**

***Index Terms*—Common vector (CV), kernel-based subspace method, pattern recognition, subspace classifier.**

I. Introduction

THE LINEAR subspace classifiers are pattern recognition methods, which use a linear subspace for each class [1]. The motivation behind the subspace classifiers is the optimal reconstruction of multidimensional data with linear principal components that carry the most significant representative features. Therefore, the most conspicuous features are extracted from each class by using the corresponding training samples in the hope that those features also carry the most important discriminatory information. Although this assumption is seldom valid, good recognition rates can be achieved when the dimensionality of the sample space is large enough [2]. In subspace methods, it is assumed that the vector distribution of each class corresponds to a lower dimensional subspace of the original sample space. The subspaces representing classes are defined in terms of basis vectors that are linear combinations of the sample vectors of each class. Therefore, basis vectors spanning those subspaces must first be computed. Also, determining the dimension of each subspace is a major issue since subspace dimensions have a strong influence on the performance of the subspace classifier. In particular, large subspace dimensions lead to a low recognition performance due to the overlapping regions among classes, whereas small subspace dimensions increase the error rates because of a poor resulting approximation [2], [3]. After construction of subspaces, a test sample vector from an unknown class is classified based on the lengths of the projections of that sample onto each of the subspaces or, alternatively, on the distances of the test vector from these subspaces.

Watanabe *et al.* proposed the first subspace method, i.e., the class-featuring information compression (CLAFIC), for pattern classification [4]. This method employs the principal components analysis (PCA) to compute the basis vectors spanning the subspace of each class. However, the subspaces found by the CLAFIC method may sometimes have a large overlapping region in common, which causes poor classification of data samples. Therefore, the method of orthogonal subspaces was proposed, which attempts to remove the common regions of the classes and makes the subspaces mutually orthogonal [5]. Fukunaga and Koontz proposed a new method, which enabled to select the basis vectors in such a way that the projections onto the so-called rival subspaces are minimized [6]. Finally, learning subspace method (LSM), in which the subspaces are iteratively modified to diminish the number of misclassifications, has been proposed in [7]. However, it turned out that the final computed basis vectors that are obtained using the LSM are sensitive to the presentation order of the training set samples. This problem is resolved in [8] by the introduction of the averaged learning subspace (ALS) method, in which the correction of the subspaces is carried out in a batch fashion. This process increased the statistical stability since the computed basis vectors representing classes are independent of the presentation order of the training set samples [9].

In some classification tasks, the dimensionality of the sample space can be larger than the number of training samples in each class. It is reasonable to expect that these high-dimensional

Manuscript received December 26, 2005; revised November 21, 2006.

H. Cevikalp is with the Department of Electrical and Electronics Engineering, Eskisehir Osmangazi University, 26480 Eskisehir, Turkey (e-mail: hakan.cevikalp@gmail.com).

M. Neamtu is with Center for Constructive Approximation, Department of Mathematics, Vanderbilt University, Nashville, TN 37235 USA.

A. Barkana is with the Department of Electrical and Electronics Engineering, Anadolu University, 26470 Eskisehir, Turkey.

Digital Object Identifier 10.1109/TSMCB.2007.896011

spaces contain more information that can be used to detect classes with more accuracy. However, because of the curse of dimensionality phenomenon [10], most of the classification techniques that carry out computations at full dimensionality may not deliver the advantages of high-dimensional sample spaces when the number of training samples is insufficient. Therefore, the dimensionality of the sample space is usually reduced before applying a classifier to data samples in the original sample space. The PCA [11] and the Fisher's linear discriminant analysis [12] are two popular feature extraction methods that are used for dimension reduction. However, dimensionality reduction through feature extraction may cause loss of important discriminatory information. Unlike some other classifiers, the subspace classifiers have been shown to work well in classification tasks in high-dimensional sample spaces. Therefore, Gulmezoglu *et al.* proposed the common vector (CV) subspace classifier method that models subspaces in such a way that their basis vectors span the null space of the covariance matrix of each class, assuming that the number of samples in each class is smaller than or equal to the dimensionality of the sample space [13], [14]. This method has been successfully applied in the isolated word recognition and face recognition problems [14], [15]. It was demonstrated that if the training set samples are linearly independent, then the extracted features are optimal from the classification point of view, and all training set samples can be classified correctly [13], [14].

The class decision boundaries that are obtained by the linear subspaces are quadratic. However, the linear subspaces may not extract nonlinear features of classes since each class is associated with a linear subspace [16], [17]. The kernel-based nonlinear subspace method, which is called the kernel CLAFIC, was developed to overcome this limitation [16], [17]. In this approach, it is assumed that samples from each class lie in some nonlinear subspace. Therefore, all data samples in each class are mapped to a higher dimensional feature space through a nonlinear kernel mapping function, and the kernel PCA [18] has been employed to compute the principal components of the correlation matrices of classes in the mapped space. This process spreads the data over a greater volume, which in turn reduces overlapping regions among the classes and enhances the potential for discrimination. The kernel CLAFIC method is formulated in terms of dot products of the mapped samples, and kernel functions are used to compute these dot products. Therefore, the mapping function or the mapped samples are not used explicitly, which makes the method feasible. In addition, Tsuda showed that under some conditions, the subspaces in the mapped space do not have overlapping regions [17]. It has been reported that the performance of the kernel CLAFIC method is superior to the linear subspace classifiers [16], [17].

In this paper, we introduce a variation of the CV method, which is called the modified CV (MCV) method, which is then extended to the nonlinear case. The new nonlinear method, which will be referred to as the kernel CV method, consists in applying the MCV method in the setting of a nonlinearly mapped higher dimensional feature space. The remainder of this paper is organized as follows: In Section II, we first review the CV method and then introduce its variation, i.e., the MCV method. Section III describes the kernel CV method. We discuss our experimental results in Section IV, and our conclusions are given in Section V.

II. Variation of the CV Method

The CV method is a subspace classifier that extracts features that are common to all samples in each class. To accomplish this, the method eliminates certain features that are in the direction of the eigenvectors corresponding to the nonzero eigenvalues of covariance (or scatter) matrices of classes. In this way, each class is represented by the null space of its own scatter matrix. Therefore, to apply the method, the number of samples in each class must be smaller than or equal to the dimensionality of the sample space. One drawback of the CV method is that it cannot be extended to the nonlinear case. To circumvent this limitation, we propose a variant of the CV, which is called the MCV method. Since the MCV method is built on the CV method, we first recall the main idea of the CV method.

A. CV Method

Let the training set be composed of C classes, where the ith class contains N_i samples, and let x_m^i be a d-dimensional column vector, which denotes the mth sample from the ith class. There is a total of $M = \sum_{i=1}^{C} N_i$ samples in the training set. Suppose that $d \geq N_i$, for $i = 1, \ldots, C$. The scatter matrix of each class is defined as

$$S_i = \sum_{m=1}^{N_i} \left(x_m^i - \mu_i\right)\left(x_m^i - \mu_i\right)^T, \qquad i = 1, \ldots, C \quad (1)$$

where μ_i is the mean of the samples in the ith class. In the CV method, each sample in the training set is represented as

$$x_m^i = x_{m,\mathrm{dif}}^i + x_{\mathrm{com}}^i + \varepsilon_m^i, \qquad i = 1, \ldots, C;\ m = 1, \ldots, N_i \quad (2)$$

where x_{com}^i is a unique vector representing the ith class, which is called a CV, and ε_m^i is the error vector associated with the sample x_m^i. Here, the vector $x_{m,\mathrm{dif}}^i$ represents the projection of x_m^i onto the so-called difference subspace of the ith class that is spanned by the vectors $\{x_2^i - x_1^i, x_3^i - x_1^i, \ldots, x_{N_i}^i - x_1^i\}$ [14]. It was also shown that the difference subspace of any class is equal to the range of the scatter matrix of samples in that class. The CV method attempts to minimize the following criterion function for each class:

$$\begin{aligned} F_i &= \min\left(\sum_{m=1}^{N_i} \left\|\varepsilon_m^i\right\|^2\right) \\ &= \min\left(\sum_{m=1}^{N_i} \left\|x_m^i - x_{m,\mathrm{dif}}^i - x_{\mathrm{com}}^i\right\|^2\right), \qquad i = 1, \ldots, C \end{aligned} \quad (3)$$

where $\|\cdot\|$ denotes the Euclidean norm. The criterion functions attain their minimum, $F_i = 0$, $i = 1, \ldots, C$, if the CVs of classes are as chosen as

$$\begin{aligned} x^i_{\text{com}} &= x^i_m - P^{(i)}_{RS} x^i_m \\ &= P^{(i)}_{NS} x^i_m, \qquad i = 1, \ldots, C;\ m = 1, \ldots, N_i \end{aligned} \tag{4}$$

where $P^{(i)}_{RS} \in \Re^{d \times d}$ and $P^{(i)}_{NS} \in \Re^{d \times d}$ denote the orthogonal projection matrices (also called orthogonal projection operators) of the range and the null spaces of S_i, respectively. Note that the CV of each class is still d dimensional, and the CVs are independent of the choice of the sample index m. In fact, all affine combinations of samples in the ith class, $\sum_{m=1}^{N_i} \lambda_m x^i_m$, where $\sum_{m=1}^{N_i} \lambda_m = 1$ and $\lambda_m \in \Re$, give rise to the same CV. The projection matrices are unique, and they are obtained using the eigenvectors of S_i. In particular, the projection matrices can be written as $P^{(i)}_{RS} = U_i U_i^T$ and $P^{(i)}_{NS} = \overline{U}_i \overline{U}_i^T$, where U_i is the matrix whose columns are normalized eigenvectors corresponding to the nonzero eigenvalues of S_i, and $\overline{U}_i$ is the matrix whose columns are normalized eigenvectors corresponding to the zero eigenvalues of S_i. To classify a test sample x_{test}, the test sample is first projected onto the null space of the scatter matrix of each class as

$$\begin{aligned} x^i_{\text{test}} &= x_{\text{test}} - P^{(i)}_{RS} x_{\text{test}} \\ &= P^{(i)}_{NS} x_{\text{test}}, \qquad i = 1, \ldots, C. \end{aligned} \tag{5}$$

Then, the feature vectors of the test sample are compared to the CV of each class using the Euclidean distance, and the test sample is assigned to the class that gives the minimum distance, i.e.,

$$g(x_{\text{test}}) = \arg\min_{i=1,\ldots,C} \left(\left\| x^i_{\text{test}} - x^i_{\text{com}} \right\| \right). \tag{6}$$

Assuming that the class conditional means are used for computing the CVs, the aforementioned equation can also be written as

$$\begin{aligned} g(x_{\text{test}}) &= \arg\min_{i=1,\ldots,C} \left(\left\| P^i_{NS}(x_{\text{test}} - \mu_i) \right\| \right) \\ &= \arg\min_{i=1,\ldots,C} \left(\left\| \overline{U}_i (x_{\text{test}} - \mu_i) \right\| \right). \end{aligned} \tag{7}$$

Note that the aforementioned formula shows that the projection lengths can be efficiently computed in a lower dimensional space using the normalized eigenvector matrices $\overline{U}_i$ [2].

A similar method, which is called the discriminative CV (DCV) method, was proposed for the recognition tasks where the dimensionality of the sample space is larger than the total number of samples in the training set (small sample size case) [19]. In contrast to the CV method, the DCV method is built on the assumption that all classes have similar covariance structures. Therefore, the DCV method uses the same subspace (the null space of the within-class scatter) to model each class. All samples are pooled together, and they are projected onto that unique subspace in this method. As a consequence, the decision boundaries that are obtained by the DCV method are linear as opposed to the CV method, in which quadratic decision boundaries are obtained. It was reported that the DCV method usually outperforms the CV method in some face recognition tasks, where the classes have similar covariance structures. Keen reader can refer to [15] for a more detailed comparison of these methods. Recently, the kernel DCV method was proposed to apply the linear DCV scheme to recognition tasks without small sample size problem [20]. In this method, all data samples are first mapped to a higher dimensional feature space through a nonlinear mapping function, and then, the DCV method is applied in the mapped space.

B. MCV Method

It can be shown that the null space of the total scatter matrix does not contain any discriminative information for classification of data samples. This is because the projections of all samples onto this subspace give rise to the same vector [14], [20]. Therefore, without loss of generality, this subspace can be discarded from our consideration in the CV method. Then, the new subspace representing each class can be defined as the intersection of the null space of that class' scatter matrix and the range space of the total scatter matrix. The MCV method proposed here uses basis vectors spanning these mentioned intersections to represent classes. This method yields the same recognition accuracy as the CV method and, from this point of view, does not offer any improvement over the CV method. However, the MCV method enables us to extend the CV idea to the nonlinear case, as explained in Section III. The total scatter matrix is defined as

$$S_T = \sum_{i=1}^{C} \sum_{m=1}^{N_i} \left(x^i_m - \mu \right) \left(x^i_m - \mu \right)^T, \qquad i = 1, \ldots, C; \quad m = 1, \ldots, N_i \tag{8}$$

where μ is the mean of all samples. As shown in Theorem 1, the projection matrix of the null space $N(S_i)$ of the scatter matrix of the ith class and the projection matrix of the range space $R(S_T)$ of the total scatter matrix commute in the sense that

$$P^{(i)}_{NS} P = P P^{(i)}_{NS}, \qquad i = 1, \ldots, C \tag{9}$$

where $P^{(i)}_{NS}$ is the projection matrix of $N(S_i)$, and P is the projection matrix of $R(S_T)$. Therefore, the projection matrix $P^{(i)}_{\text{int}}$ of the intersection $N(S_i) \cap R(S_T)$ for each class can be found as

$$P^{(i)}_{\text{int}} = P^{(i)}_{NS} P = P P^{(i)}_{NS}, \qquad i = 1, \ldots, C \tag{10}$$

from [21]. Notice that, in general, the projection matrix of any intersection cannot be obtained using (10) if the projection matrices of the associated subspaces do not commute.

Theorem 1: Let P and $P^{(i)}_{NS}$ be the projection matrices of the subspaces $R(S_T)$ and $N(S_i)$, $i = 1, \ldots, C$, respectively. Then, P and $P^{(i)}_{NS}$ commute, i.e.,

$$P^{(i)}_{NS} P = P P^{(i)}_{NS}, \qquad i = 1, \ldots, C.$$

Proof: See Appendix I.

Since the MCV method uses the intersection subspaces $P_{\text{int}}^{(i)}$, $i = 1, \ldots, C$, to represent classes, the basis vectors spanning these intersections must be computed. The basis vectors spanning each mentioned intersection space $P_{\text{int}}^{(i)}$ can be found by using eigendecomposition techniques. In particular, the eigenvectors corresponding to eigenvalue of $\lambda = 1$ of $P_{\text{int}}^{(i)}$ span the intersection subspaces representing the classes of interest. However, this approach is not always feasible in practice, particularly if the number of samples in each class is much smaller than the dimensionality of the sample space. In this case, the size of the projection matrices can be very large (e.g., images of size 256×256 yield projection matrices of size $65\,536 \times 65\,536$). On the other hand, since the projection matrices commute, we can first efficiently transform the samples onto $R(S_T)$ by using the basis vectors of $R(S_T)$ and then find the null spaces of the classes in the transformed space, so as to compute basis vectors of the intersection subspaces. The basis vectors of $R(S_T)$ can be efficiently computed by using the smaller matrices as described in [11] and [19]. The algorithm that implements this idea can be summarized as follows:

Step 1) *Transformation of the training set samples onto $R(S_T)$:*
   a) Compute the nonzero eigenvalues and corresponding eigenvectors u_k of S_T. Set $U = [u_1 \quad \cdots \quad u_r]$, where r is the rank of S_T and cannot be bigger than $M - 1$.
   b) Transform the training set samples onto $R(S_T)$ by

$$y_m^i = U^T x_m^i, \qquad i = 1, \ldots, C;\ m = 1, \ldots, N_i. \tag{11}$$

Step 2) *Finding the null spaces of classes in the transformed space:* In the transformed space, the new scatter matrices of the classes will be

$$\begin{aligned}\widetilde{S}_i &= \sum_{m=1}^{N_i} \left(y_m^i - \widetilde{\mu}^i\right)\left(y_m^i - \widetilde{\mu}^i\right)^T \\ &= U^T S_i U, \qquad i = 1, \ldots, C\end{aligned} \tag{12}$$

where $\widetilde{\mu}^i$ is the mean of samples of the ith class in the transformed space. Apply eigendecomposition to each scatter matrix, $\widetilde{S}_i \in \Re^{r \times r}$. Let q_k^i be the eigenvectors corresponding to the zero eigenvalues of $\widetilde{S}_i$. Set $Q^{(i)} = [q_1^i \quad \ldots \quad q_{n_i}^i]$, where n_i is the dimensionality of $N(\widetilde{S}_i)$.

Step 3) *Computation of the final basis vectors of the intersection space $N(S_i) \cap R(S_T)$:* The final basis vectors spanning the intersection subspaces will be

$$W^{(i)} = UQ^{(i)}, \qquad i = 1, \ldots, C. \tag{13}$$

Note that the basis vectors span the intersection subspace $N(S_i) \cap R(S_T)$, and therefore, the following holds:

$$P_{\text{int}}^{(i)} = W^{(i)} W^{(i)^T}, \qquad i = 1, \ldots, C. \tag{14}$$

When the samples of each class are transformed onto their corresponding intersection subspace, the feature vector $\Omega_{\text{com}}^{(i)} = [\langle x_m^i, w_1^i \rangle \quad \cdots \quad \langle x_m^i, w_{n_i}^i \rangle]^T$ of each sample is the same for all samples in that class [14]. These feature vectors are called the CVs, as in the CV method. Furthermore, a CV of any class is different from the CVs of all other classes in the case the data samples are linearly independent. Therefore, this method guarantees 100% recognition accuracy when linearly independent data samples are chosen for training. However, in contrast to the CV method, the dimensionality of the feature vectors is reduced to n_i in this case since the basis vectors are utilized for feature extraction. To recognize a test sample, we compute the Euclidean distances between the test sample feature vector $\Omega_{\text{test}}^{(i)} = [\langle x_{\text{test}}, w_1^i \rangle \quad \cdots \quad \langle x_{\text{test}}, w_{n_i}^i \rangle]^T$ and the CVs of each class using the Euclidean distance. Then, we assign the test sample to the class that minimizes this distance, i.e.,

$$\begin{aligned} g(x_{\text{test}}) &= \arg\min_{i=1,\ldots,C} \left(\left\| \Omega_{\text{test}}^{(i)} - \Omega_{\text{com}}^{(i)} \right\| \right) \\ &= \arg\min_{i=1,\ldots,C} \left(\left\| W^{(i)^T} (x_{\text{test}} - \mu_i) \right\| \right). \end{aligned} \tag{15}$$

Transformation of the training set samples onto $R(S_T)$ can also be done efficiently by employing the so-called difference subspace of the total scatter matrix since the eigenvalues (i.e., an explicit symmetric Schur decomposition) of S_T need not be computed explicitly [19].

The MCV method yields the same results as the CV method; however, the training phase requires more computations as compared to the CV method. Although the method proposed here may not appear particularly advantageous at first, it enables us to extend the CV idea to the nonlinear case. Note that it is not possible to extend the classical CV method to the nonlinear case directly since it cannot be formulated using dot products of the mapped samples. Next, we introduce a new nonlinear method by incorporating the kernel trick into the procedure proposed here.

III. Kernel CV Method

This method consists in mapping the given training set samples into an implicit higher dimensional space $\Im$ using a nonlinear kernel mapping function and then applying the linear MCV method in the mapped space. As in the other kernel methods using the kernel trick, the kernel CV method is also formulated in terms of the dot products of the mapped samples, which are computed using kernel functions. As a result, the mapping function and the mapped samples are not used explicitly, which makes the method computationally feasible.

Let $\Phi = [\Phi^{(1)}, \Phi^{(2)}, \ldots, \Phi^{(C)}]$ represent the matrix whose columns are the mapped training samples in $\Im$, where $\Phi^{(i)} = [\phi(x_1^i), \phi(x_2^i), \ldots, \phi(x_{N_i}^i)]$ is the matrix whose columns are mapped samples of the ith class. The scatter matrix S_i^Φ of each

class and the scatter matrix S_T^Φ of the pooled data in $\Im$ are given by

$$S_i^\Phi = \sum_{m=1}^{N_i} \left(\phi\left(x_m^i\right) - \mu_i^\Phi\right)\left(\phi\left(x_m^i\right) - \mu_i^\Phi\right)^T = \left(\Phi^{(i)} - \Phi^{(i)}1_{N_i}\right)\left(\Phi^{(i)} - \Phi^{(i)}1_{N_i}\right)^T, \qquad i = 1, \ldots, C \tag{16}$$

$$S_T^\Phi = \sum_{i=1}^{C}\sum_{m=1}^{N_i} \left(\phi\left(x_m^i\right) - \mu^\Phi\right)\left(\phi\left(x_m^i\right) - \mu^\Phi\right)^T = (\Phi - \Phi 1_M)(\Phi - \Phi 1_M)^T \tag{17}$$

where μ_i^Φ is the mean of mapped samples in the ith class, and μ^Φ is the mean of all mapped samples. Here, $1_{N_i} \in \Re^{N_i \times N_i}$ is a matrix whose elements are all $1/N_i$, and $1_M \in \Re^{M\times M}$ is a matrix with entries $1/M$. The kernel matrix of the mapped data is given as $K = \Phi^T\Phi = \left(K^{ij}\right)_{\substack{i=1,\ldots,C\\ j=1,\ldots,C}}$, where each submatrix $K^{ij} \in \Re^{N_i\times N_j}$ is defined as

$$K^{ij} = \left(k_{mn}^{ij}\right)_{\substack{m=1,\ldots,N_i\\ n=1,\ldots,N_j}} = \left\langle \phi\left(x_m^i\right), \phi\left(x_n^j\right)\right\rangle = k\left(x_m^i, x_n^j\right)_{\substack{m=1,\ldots,N_i\\ n=1,\ldots,N_j}}. \tag{18}$$

In the aforementioned equation, $k(\cdot)$ represents the kernel function.

Our aim is to find basis vectors for the intersection subspaces $N(S_i^\Phi) \cap R(S_T^\Phi)$, $i = 1, \ldots, C$, for each class. To find these basis vectors, we follow the steps given in the previous section; we first transform all training samples onto $R(S_T^\Phi)$ and then find the null spaces of the classes in the transformed space. The transformation of training set samples onto $R(S_T^\Phi)$ can be done easily by employing the kernel PCA method. Then, we find the vectors spanning the null spaces of the scatter matrices of the transformed samples. The algorithm for the kernel CV method can be summarized as follows.

Step 1) Transform the training set samples onto $R(S_T^\Phi)$ using the kernel PCA. Let $\widetilde{K}$ be the kernel matrix of the centered mapped samples [18]. If we apply eigendecomposition to $\widetilde{K}$, we obtain

$$\widetilde{K} = K - 1_M K - K 1_M + 1_M K 1_M = U\Lambda U^T \in \Re^{M\times M} \tag{19}$$

where Λ is the diagonal matrix of nonzero eigenvalues, and U is the matrix of normalized eigenvectors associated to Λ. The matrix that transforms the training set samples onto $R(S_T^\Phi)$ is $(\Phi - \Phi 1_M)U\Lambda^{-1/2}$.

Step 2) Compute the scatter matrix of each class in the transformed space. The new scatter matrix $\widetilde{S}_i^\Phi \in \Re^{r\times r}$ (r is the rank of $R(S_T^\Phi)$ and that cannot be larger than $M-1$) of each class in the reduced space becomes

$$\widetilde{S}_i^\Phi = \left((\Phi - \Phi 1_M)U\Lambda^{-1/2}\right)^T S_i^\Phi (\Phi - \Phi 1_M)U\Lambda^{-1/2} = \Lambda^{-1/2}U^T\widetilde{K}^{(i)}\widetilde{K}^{(i)^T}U\Lambda^{-1/2}, \qquad i = 1, \ldots, C. \tag{20}$$

Here, the matrix $\widetilde{K}^{(i)} \in \Re^{M\times N_i}$ is written as

$$\widetilde{K}^{(i)} = K^{(i)} - K^{(i)}1_{N_i} - 1_M K^{(i)} + 1_M K^{(i)} 1_{N_i} = \left(K^{(i)} - 1_M K^{(i)}\right)(I - 1_{N_i}) \tag{21}$$

where the matrix $K^{(i)} \in \Re^{M\times N_i}$ is given by $K^{(i)} = \Phi^T\Phi^{(i)} = \left(K^{(i)j}\right)_{j=1,\ldots,C}$, and each submatrix $K^{(i)j} \in \Re^{N_j\times N_i}$ is defined as

$$K^{(i)j} = \left(k_{mn}^{(i)j}\right)_{\substack{m=1,\ldots,N_j\\ n=1,\ldots,N_i}} = \left\langle \phi\left(x_m^j\right), \phi\left(x_n^i\right)\right\rangle = k\left(x_m^j, x_n^i\right)_{\substack{m=1,\ldots,N_j\\ n=1,\ldots,N_i}} \tag{22}$$

(see Appendix II for a formula of $K^{(i)}$).

Step 3) For each class, find a basis of the null space of $\widetilde{S}_i^\Phi$. This can be done by an eigendecomposition. The normalized eigenvectors corresponding to the zero eigenvalues of $\widetilde{S}_i^\Phi$ form an orthonormal basis for the null space of $\widetilde{S}_i^\Phi$. Let $Q^{(i)}$ be a matrix whose columns are the computed eigenvectors corresponding to the zero eigenvalues, such that

$$Q^{(i)^T}\widetilde{S}_i^\Phi Q^{(i)} = 0, \qquad i = 1, \ldots, C. \tag{23}$$

Step 4) The matrix of basis vectors $W^{(i)}$, whose columns span the intersection subspace of the ith class, is

$$W^{(i)} = (\Phi - \Phi 1_M)U\Lambda^{-1/2}Q^{(i)}, \qquad i = 1, \ldots, C. \tag{24}$$

The number of basis vectors spanning the intersection subspaces is determined by the dimensionality of $N(\widetilde{S}_i^\Phi)$ for each class. After performing feature extraction, all training set samples in each class give rise to the CV of that class, which is given as

$$\Omega_{\text{com}}^{(i)} = W^{(i)^T}\phi\left(x_m^i\right) = Q^{(i)^T}\Lambda^{-1/2}U^T\tilde{l}_m^i, \qquad i = 1, \ldots, C;\ m = 1, \ldots, N_i \tag{25}$$

where $\tilde{l}_m^i = (l_m^i - 1_M l_m^i) \in \Re^M$, and $l_m^i \in \Re^M$ is a vector with entries $k(x_n^j, x_m^i)_{\substack{j=1,\ldots,C\\ n=1,\ldots,N_j}}$ (see Appendix II for a formula for l_m^i). Note that the CV given in (25) is independent of the sample index m, and hence, one can choose any sample from a particular class to obtain the corresponding CV.

Tsuda proved that if the kernel matrix K is strictly positive, then all mapped samples are linearly independent in the mapped space [17]. Therefore, although data samples are not linearly independent in the original sample space, if we choose a kernel function that makes K a positive definite matrix, similarly to the linear CV case, a 100% recognition accuracy with respect to the training set data is also guaranteed for this method. To

recognize a given test sample, we compute the feature vector of a test sample by

$$\begin{aligned}\Omega_{\text{test}}^{(i)} &= W^{(i)^T}\phi(x_{\text{test}}) \\ &= Q^{(i)^T}\Lambda^{-1/2}U^T\widetilde{l}_{\text{test}}, \qquad i=1,\ldots,C \end{aligned} \tag{26}$$

where $\widetilde{l}_{\text{test}} = (l_{\text{test}} - 1_M l_{\text{test}}) \in \Re^M$, and $l_{\text{test}} \in \Re^M$ is a vector with entries $k(x_m^i, x_{\text{test}})_{\substack{i=1,\ldots,C \\ m=1,\ldots,N_i}}$. Then, we compare the Euclidean distances between the CVs and the feature vector of the test sample for each class, using (15), and we assign the test sample to the class that minimizes this distance.

A. Comparison of the Linear and Kernel CV Methods

The differences between the MCV and kernel CV methods can be summarized as follows.

1) The MCV method employs linear subspaces to represent classes. Thus, the decision boundaries that are obtained by this method are quadratic. However, the kernel CV method employs nonlinear subspaces for each class since the mapped space is nonlinearly related to the original sample space. Therefore, one can obtain nonlinear decision boundaries using the kernel CV method. Additionally, we have the flexibility of creating different nonlinear decision boundaries by simply changing the kernel functions.
2) The MCV method can only be applied if the dimensionality of the sample space is larger than the rank of the scatter matrix of training samples in each class. Furthermore, as in the other linear subspace classifiers, the dimensionality of the sample space must be large as compared to the number of samples in each class for good recognition rates. However, these limitations do not apply to the proposed kernel method. One can use the kernel CV method even if the number of samples in each class is larger than the dimensionality of the sample space using kernel functions, which ensure the high dimensionality of the implicit mapped space.

However, all these mentioned improvements are achieved at the expense of more intense computations. In particular, the kernel CV method roughly results in additional $(CMd + 10M^3)$ multiplications in the training set, assuming that the linear kernel function is used. However, we are mostly interested in real-time performance of a method, which is determined by the time that is required to classify a new test sample, since the training phase can be performed offline. Assuming that the linear kernel is used, a total of $(dr + \sum_{i=1}^{C} rn_i)$ multiplications are required during classification for the MCV method, whereas the classification phase of the kernel DCV requires $(CMd + M\sum_{i=1}^{C} n_i)$ multiplications. Here, n_i represents the dimensionality of the null space of each class in the reduced space, and $n_i < M-1$. Thus, if $d >> \sum_{i=1}^{C} n_i$, then the MCV method is approximately C times faster than the kernel CV method since the rank r is usually equal to $M-1$. Of course, choosing kernel functions that are different from linear might result in additional computational cost in the kernel CV method.

IV. Experimental Results

The ratio of the dimensionality of the sample space to the training set size is a very important factor that affects recognition performances of subspace classifiers. Therefore, in our experiments, we used seven different real-world data sets having varying ratios of training set sizes to dimensionalities. All databases except the AR face database [22] and the Columbia Object Image Library (Coil-100) database [23] are chosen from the University of California-Irvine repository [24].

An appropriate selection of kernel functions for special recognition tasks is still an open problem since different kernel functions give rise to different constructions of the implicit feature space [25]. We have experimented with polynomial kernels $k(x,y) = (\langle x,y\rangle)^n$ of degree $n = 2, 3$ and the Gaussian kernel $k(x,y) = \exp(-\|x-y\|^2/q)$. A small set of randomly created training and test sets was employed to compute the best Gaussian parameters q for each method. We first globally searched for the best Gaussian parameter over a wide range of the parameter space. Then, we carried out a local search in the neighborhood of the Gaussian parameter that yielded the best recognition rate and computed the final best Gaussian parameter.

Beside the methods proposed here, we also tested the linear CLAFIC method, the ALS method, the linear discriminant analysis classifier utilizing the Mahalanobis distance, the DCV method, the kernel CLAFIC method, the kernel DCV method, and the support vector machine (SVM) classifier for a better assessment of the recognition performances of the proposed methods. Class correlation matrices were used to find the basis vectors spanning the subspaces of classes for the CLAFIC and kernel CLAFIC subspace methods. For these methods, the dimension of each subspace was selected as the value by which the ratio of the sum of the retained eigenvalues to the sum of all eigenvalues exceeds 0.98 [26]. We used the same values for both α and β parameters in the ALS method, as recommended in [2]. To determine the best values for α and β, we followed a similar procedure as in the computation of the Gaussian kernel parameter. We covered the values between 0 and 6 during the selection phase of the best value in all experiments. The maximum number of epochs in the ALS algorithm was chosen to be 8 to avoid the overfitting problem. We adopted the "one-against-one" (OAO) procedure to extend classic two-class SVM problem to the multiclass recognition problem, and the "max wins" voting approach was utilized during the testing phase [27]. The OAO procedure constructs $C(C-1)/2$ binary classifiers where each classifier is trained on data samples from two classes. We assumed that the regularization parameter γ is the same for each binary classifier.

A. Experiments With the Fisher's Iris Database

The Iris flower database [12] contains four measurements on 50 Iris specimens for each of the three species, namely Iris setosa, Iris versicolor, and Iris virginica, for a total of 150 samples

TABLE I
RECOGNITION RATES OF METHODS ON THE FISHER'S IRIS DATABASE

Methods	Recognition Rates (%)
CLAFIC	97.33
ALS	97.33
LDA	**98**
Kernel CLAFIC, $q=1$	96
SVM, $q=3$	**98**
Kernel CV, $q=0.7$	96
Kernel DCV, $q=0.1$	96

in the database. It was reported that the first class is linearly separable from the other two and that the latter two are not linearly separable from each other. The data are not normalized in these experiments. Since the number of samples in each class is much larger than the dimensionality of the sample space, neither the MCV method nor DCV method can be used for this database. However, we can use the kernel CV method by employing the Gaussian kernel function. Note that we cannot employ polynomial kernels in this case since the dimensionality of the transformed feature space is smaller than the number of samples in each class for those kernels. We adopted the leave-one-out strategy to test the generalization performances of the methods. The parameters α and β of the ALS method were set to 0.75, which gave the lowest recognition error. The regularization parameter γ of SVM classifier was chosen as 5. The computed recognition rates and the Gaussian parameters are given in Table I.

As can be seen in the table, the SVM and LDA classifiers achieve the best recognition rate among all methods that are tested here. Both the proposed method and the kernel CLAFIC method yield the same recognition accuracy, which is also the smallest. These experimental results show that the quadratic decision boundaries that are obtained by the linear subspace classifiers do a better job than the nonlinear decision boundaries that are obtained by kernel subspace classifiers for this database. However, the quadratic decision boundaries perform worse than the linear decision boundaries that are obtained by the LDA classifier. The recognition rate of the proposed kernel subspace classifier can be compared to the reported recognition rate of the kernel generalized discriminant analysis (95.33%) [28] on the same database.

B. Experiments With the Wine Database

The wine database [24] includes 178 data samples from three different wine cultivars. Each data sample has 13 numeric attributes that are derived from a chemical analysis of wines. In our experiments, we normalized the data so that the values of the attributes lie between -1 and 1. As in the previous experiment, both the MCV and DCV methods cannot be used for this database since the dimensionality of the sample space is smaller than the training set samples in each class. The leave-one-out strategy was again used to test the generalization performances of the methods. The parameters α and β of the ALS method were set to 3. The SVM regularization parameter was set to 4 for all kernels. The computed recognition rates are shown in Table II.

Among all methods, the best recognition rates were obtained by the kernel CLAFIC and the kernel CV methods, both utilizing the Gaussian kernel. On the other hand, the recognition rates of the proposed method using polynomial kernels were low. This was because employing polynomial kernel functions yielded a smaller transformed space after the kernel PCA step for this database. The sizes of the transformed space were 91 and 170 for the polynomial kernel function with degrees 2 and 3, respectively. As a result, the size of each intersection subspace was small, which diminished the separability among the classes. Thus, our proposed method did not perform satisfactorily. In contrast, employing the Gaussian kernel in the proposed method increased the dimensionality of the transformed space to 176. Consequently, increasing the dimensionality of the transformed space enhanced separation and gave the highest recognition rate among all methods. Note also that the kernel CLAFIC method outperformed its linear counterpart for all kernel functions.

C. Experiments With the Image Segmentation Database

The image segmentation database [24] consists of samples randomly drawn from a database of seven outdoor images. The images were hand segmented to create a classification for every pixel. Each sample has a 3×3 region and 19 attributes. There are a total of seven classes, each having 330 samples. The attributes were normalized to lie in interval $[-1, 1]$.

We used randomly chosen 165 samples of each class for training, and the remaining samples were used for testing. Therefore, both training and test sets consisted of 1155 samples. Note that there is no overlap between the training and test sets. This process was repeated 25 times, and 25 different training and test sets were created. The first five data sets were used for parameter selection, whereas the remaining sets were used to evaluate performance. Thus, the final recognition rates for the experiment were found by averaging these 20 rates that were obtained in each trial. We could not use the MCV and DCV methods since the dimensionality of the sample space, which is equal to 19, is smaller than the number of training set samples of each class, which is equal to 165. Similarly, we could not use the LDA classifier because of the singularity of the within-class scatter matrix. Furthermore, it was not possible to apply the kernel CV method using the polynomial kernel function with degree 2 since the dimensionality of the transformed space via kernel PCA was smaller than the number of samples in each class utilizing this kernel function. We empirically set α and β parameters to value of 0.035 for the ALS method. The SVM regularization parameter was set to 50 for the polynomial kernels and 40 for the Gaussian kernel. The means and the standard deviations of computed recognition rates on this database are given in Table III.

As can be seen in the table, the SVM classifier achieved the highest recognition rates for all kernel functions that were used here. Our proposed method outperformed all other subspace classifiers in all cases. It is interesting that although the kernel CLAFIC method with the Gaussian kernel outperformed the

TABLE II
RECOGNITION RATES OF METHODS ON THE WINE DATABASE

Linear Classifiers	**Recognition Rates (%)**		
CLAFIC	94.38		
ALS	95.51		
LDA	97.75		
Nonlinear Classifiers & Gaussian Kernel Parameters	Polynomial kernel functions		Gaussian kernel function
	$n = 2$	$n = 3$	
Kernel CLAFIC, $q = 2$	**97.19**	**96.63**	**98.88**
SVM, $q = 2$	96.63	**96.63**	97.75
Kernel CV, $q = 0.11$	79.21	79.78	**98.88**
Kernel DCV, $q = 0.10$	not applicable	66.29	98.31

TABLE III
RECOGNITION RATES OF METHODS ON THE IMAGE SEGMENTATION DATABASE

Linear Classifiers	**Recognition Rates (%) and Standard Deviations**	
CLAFIC	87.97, $\sigma = 1.48$	
ALS	89.15, $\sigma = 1.51$	
Nonlinear Classifiers & Gaussian Kernel Parameters	Polynomial kernel function $n = 3$	Gaussian kernel function
Kernel CLAFIC, $q = 0.1$	87.56, $\sigma = 1.59$	96.07, $\sigma = 0.48$
SVM, $q = 0.75$	**96.22**, $\sigma = 0.52$	**96.40**, $\sigma = 0.55$
Kernel CV, $q = 0.46$	95.50, $\sigma = 0.48$	96.32, $\sigma = 0.44$
Kernel DCV, $q = 0.15$	not applicable	94.08, $\sigma = 1.23$

TABLE IV
RECOGNITION RATES OF METHODS ON THE FOURIER COEFFICIENTS DATABASE

Linear Classifiers	**Recognition Rates (%) and Standard Deviations**		
CLAFIC	80.80, $\sigma = 1.05$		
ALS	80.84, $\sigma = 0.98$		
LDA	80.16, $\sigma = 0.83$		
Nonlinear Classifiers & Gaussian Kernel Parameters	Polynomial kernel functions		Gaussian kernel function
	$n = 2$	$n = 3$	
Kernel CLAFIC, $q = 0.58$	82.08, $\sigma = 1.11$	82.93, $\sigma = 0.87$	84.22, $\sigma = 0.72$
SVM, $q = 0.38$	**84.94**, $\sigma = 0.75$	**84.75**, $\sigma = 0.76$	**85.02**, $\sigma = 0.90$
Kernel CV, $q = 0.87$	83.98, $\sigma = 0.82$	84.08, $\sigma = 0.72$	84.35, $\sigma = 0.70$
Kernel DCV, $q = 0.46$	82.80, $\sigma = 0.79$	83.90, $\sigma = 0.74$	84.98, $\sigma = 0.65$

linear subspace classifiers, its recognition rate for the polynomial kernel is worse than those of linear subspace classifiers.

D. Experiments With the Digit Database of Handwritten Numerals

This database [29] includes ten classes, each having 200 patterns. Sample patterns are available in the form of binary images. These characters are represented in terms of different feature sets forming distinct databases. In our experiments, we used only 76-dimensional Fourier coefficients and 240-dimensional pixel averages. The data were not normalized in these experiments. We randomly chose 100 samples from each class for training, and the remaining samples were used for testing. Thus, a training set of 1000 samples and a test set of 1000 samples were created for both databases. This process was repeated 25 times, and 25 different training and test sets were created. The first five data sets were used for parameter selection, and the rest were used for performance evaluation.

We could not use both the MCV and DCV methods for the Fourier coefficients database since the dimensionality of the sample space is smaller than the number of samples in each class. However, we applied the kernel CV method to this database by employing polynomial and the Gaussian kernel functions. The parameters α and β of the ALS method were empirically set to value of 0.025 for this database. The regularization parameter of SVM classifier was set to 5 for polynomial kernels and to 3 for the Gaussian kernel. The means and the standard deviations of computed recognition rates on the Fourier coefficients database are given in Table IV.

For the Fourier coefficients database, the best recognition rate was obtained by the SVM classifier. The proposed kernel CV method achieved the highest recognition rates among all subspace classifiers. Note that it outperformed the kernel CLAFIC method for all kernel functions that were used here. The results also show that both kernel subspace classifiers offer a significant improvement over the recognition rates of the linear subspace classifiers and LDA.

TABLE V
RECOGNITION RATES OF METHODS ON THE PIXEL AVERAGES DATABASE

Linear Classifiers	**Recognition Rates (%) and Standard Deviations**		
CLAFIC	97.38, $\sigma = 0.38$		
ALS	97.38, $\sigma = 0.38$		
LDA	94.53, $\sigma = 0.65$		
MCV	96.83, $\sigma = 0.43$		
Nonlinear Classifiers & Gaussian Kernel Parameters	Polynomial kernel functions		Gaussian kernel function
	$n = 2$	$n = 3$	
Kernel CLAFIC, $q = 4000$	**97.93**, $\sigma = 0.34$	97.95, $\sigma = 0.34$	97.96, $\sigma = 0.42$
SVM, $q = 30$	97.72, $\sigma = 0.49$	97.83, $\sigma = 0.40$	97.92, $\sigma = 0.32$
Kernel CV, $q = 3000$	97.91, $\sigma = 0.38$	97.92, $\sigma = 0.37$	97.96, $\sigma = 0.35$
Kernel DCV, $q = 1200$	97.92, $\sigma = 0.42$	**98.12**, $\sigma = 0.40$	**98.21**, $\sigma = 0.33$

In contrast to the Fourier coefficients database, classification of the pixel averages database is a recognition task, for which we can apply the MCV method. This is because the dimensionality of the sample space, which is 240, is larger than the number of samples in each class, which is 100. However, we still cannot apply the DCV method since the dimensionality of the sample space is smaller than the total number of training samples, which is 1000. The subspaces found by the CLAFIC method did not give rise to any misclassifications in the training set. Thus, the ALS method was equivalent to the CLAFIC method in this case. The regularization parameter of the SVM classifier was chosen to be 10 for the polynomial kernels and 5 for the Gaussian kernel. The computed recognition rates for the pixel averages database are given in Table V.

For the pixel averages database, the best recognition rate was attained by the kernel DCV method. Both the kernel CLAFIC and kernel CV methods outperformed the SVM classifier for all used kernel functions. Similarly, kernel subspace classifiers outperformed the linear subspace classifier methods as well as the LDA classifier. Among the linear subspace classifiers, the CLAFIC method gave better recognition rates than the MCV method because of the low dimensionality of the sample space as compared to the training set samples. Note that the original dimensionality of the sample space was 240. Thus, the size of the intersection subspaces representing classes was much smaller, which caused poor recognition rates for the MCV method. However, employing the kernel functions increased the dimensionality of the intersection subspaces and improved the recognition rates.

For both the Fourier coefficients and the pixel averages databases, the LDA classifier resulted in the smallest recognition rate. Recall that the LDA classifier is built on the assumption that all classes have identical covariance structures. It is apparent from the results that this assumption is not satisfied for these two databases.

E. Experiments With the AR Face Database

The AR face [22] database includes 26 frontal images with different facial expressions, illumination conditions, and occlusions for 126 subjects. Images were recorded in two different sessions, 14 days apart. Thirteen images were recorded under controlled circumstances in each session. The size of the images in the database is 768 $\times$ 576 pixels, and each pixel is represented by 24 bits of red, green, and blue color values.

We randomly selected 50 individuals (30 males and 20 females) for the experiment. Only nonoccluded images [(a)–(g) and (n)–(t) as in Fig. 1] were chosen for every subject. Thus, our face database size was 700, with 14 images per subject. First, these images were converted to grayscale images. Second, we preprocessed these images by aligning and scaling them so that the distances between the eyes were the same for all images and also ensuring that the eyes occurred in the same coordinates of the image. The resulting images were then cropped. The final size of the images was 222 $\times$ 299. Finally, based on empirical observations, we decreased the dimensionality of the sample space to 99 $\times$ 134 by downsampling. The training set consisted of seven images that were randomly selected from each subject, and the rest of the images were used for the test set. Thus, a training set of 350 images and a test set of 350 images were created. Similar to the previous cases, this process was repeated 15 times, and 15 different training and test sets were created. The first five data sets were used for parameter selection, and the rest were used for performance evaluation. The final recognition rates for the experiment were found by averaging these ten rates that were obtained in each trial. Note that we can apply both the MCV and DCV methods here since the dimensionality of the sample space, which is 13 226, is much larger than the total number of samples in the training set. However, it is not possible to apply the LDA classifier since the within-class scatter matrix is rank deficient. Furthermore, we could not apply the ALS method to this database because of computational difficulties. The dimensionality of the sample space was reduced to 349 through PCA, and then, samples were standardized before the application of the SVM classifier to reduce the computational complexity of this method for these experiments. We set the SVM regularization parameter to $\gamma = 10$ for all kernels. The means and the standard deviations of computed recognition rates on the AR face database are given in Table VI.

In terms of classification accuracy, the DCV and kernel DCV methods using the Gaussian kernel achieved the highest recognition rate. Both the MCV and kernel CV methods utilizing the Gaussian kernel achieved the best recognition rates among all subspace classifier methods. However, they did not outperform the SVM, DCV, and kernel DCV methods. The kernel CLAFIC method outperformed its linear counterpart, but it did not

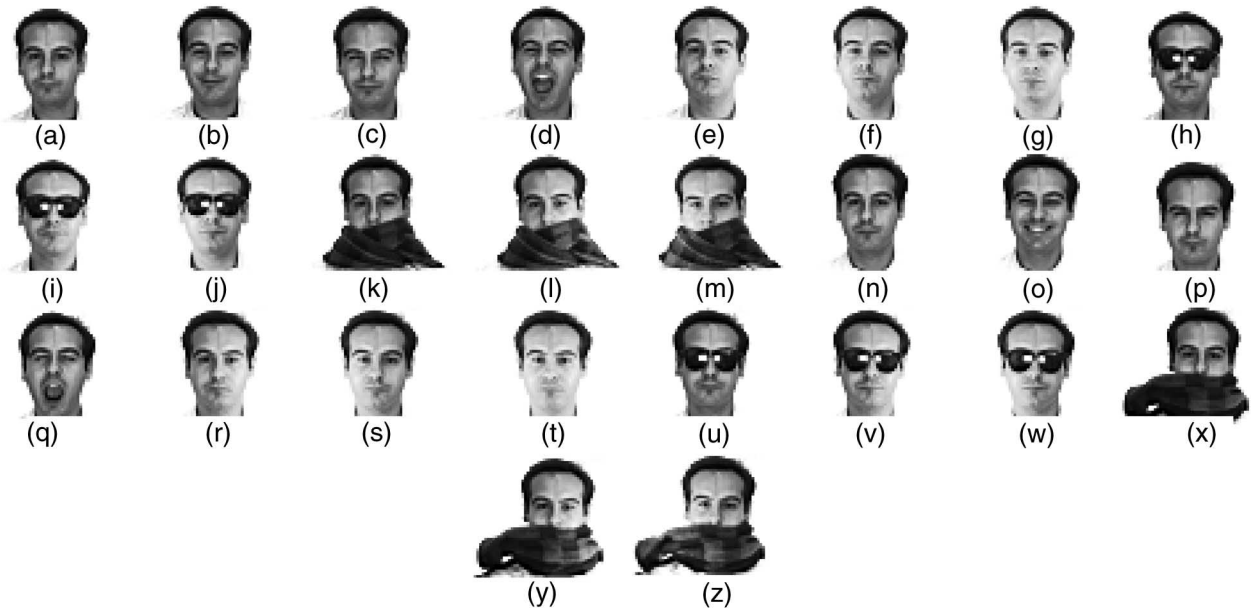

Fig. 1. Images of one subject in the AR face database. First, 13 images (a)–(m) were taken in one session and the others (n)–(z) in another session. Only nonoccluded images (a)–(g) and (n)–(t) were used in our experiments.

TABLE VI
RECOGNITION RATES OF METHODS ON THE AR FACE DATABASE

Linear Classifiers	**Recognition Rates (%) and Standard Deviations**		
CLAFIC	82.77, $\sigma = 2.19$		
MCV	92.26, $\sigma = 1.66$		
DCV	**99.17**, $\sigma = 0.45$		
Nonlinear Classifiers & Gaussian Kernel Parameters	Polynomial kernel functions		Gaussian kernel function
	$n = 2$	$n = 3$	
Kernel CLAFIC, $q = 2.11e8$	88.00, $\sigma = 2.24$	87.57, $\sigma = 1.10$	86.06, $\sigma = 1.87$
SVM, $q = 1$	93.68, $\sigma = 0.95$	**96.51**, $\sigma = 0.86$	97.45, $\sigma = 1.62$
Kernel CV, $q = 3.08e10$	91.71, $\sigma = 1.72$	91.17, $\sigma = 1.74$	92.26, $\sigma = 1.75$
Kernel DCV, $q = 3.10e10$	**98.17**, $\sigma = 0.78$	95.86, $\sigma = 0.99$	**99.17**, $\sigma = 0.52$

offer any improvement over the MCV method. The kernel CV method significantly outperformed the kernel CLAFIC method in all cases. However, the method did not give rise to any improvement over its linear counterpart, which is the MCV method. This can be attributed to the fact that the face image samples are mostly linearly separable since the original dimensionality of the sample space is too large as compared to the number of samples in the classes. Therefore, a further increase of the sample space did not improve the results. Notice that the MCV method significantly outperformed the CLAFIC method in contrast to the comparison results that were obtained on the pixel averages database, which is explained by the fact that the dimensionality of the original sample space was too large.

Similarly to the results reported in [15], the DCV method outperformed the MCV method for the AR face database. The reason for this is that the number of samples in each class was not large enough to model the intersection subspaces representing classes in the MCV method. These classes were better represented by the unique subspace (the null space of the within-class scatter matrix) that were obtained using all the data samples in the training set by the DCV method. These results also reveal the fact that the face images in the AR face database have similar intraclass variations. If the number of samples per class is increased, we expect that the generalization performance of the MCV method would approach to the generalization performance of the DCV for the AR face database, as reported in [15].

F. Experiments With the Coil-100 Database

The aforementioned experiments on the AR face database show that the face classes have similar covariance structures since the DCV method outperformed all other methods. On the other hand, the classes may have very different class covariance structures for object recognition problems. The MCV method will be a better choice for such classification tasks since it is built on the assumption that classes have different intraclass variations. To verify this hypothesis, we tested our proposed methods on the Coil-100 database [23]. This database includes 100 different objects and 72 views of each object taken at 5-degree-apart orientations. All images are converted to gray-scale images, and their original dimensionality 128×128 was reduced to 64×64 by downsampling. Based on the appearances of objects, we chose 40 objects that are more likely to have different covariance structures. These selected object classes are shown in Fig. 2. We randomly selected

Fig. 2. Selected objects from the Coil-100 database.

TABLE VII
RECOGNITION RATES OF METHODS ON THE COIL-100 DATABASE

Linear Classifiers	**Recognition Rates (%) and Standard Deviations**		
CLAFIC	99.12, $\sigma = 0.57$		
MCV	**99.31**, $\sigma = 0.53$		
DCV	92.26, $\sigma = 0.37$		
Nonlinear Classifiers & Gaussian Kernel Parameters	Polynomial kernel functions		Gaussian kernel function
	$n = 2$	$n = 3$	
Kernel CLAFIC, $q = 1.67e5$	99.01, $\sigma = 0.55$	98.67, $\sigma = 0.50$	98.96, $\sigma = 0.55$
SVM, $q = 10$	98.09, $\sigma = 0.46$	94.91, $\sigma = 0.79$	98.26, $\sigma = 0.42$
Kernel CV, $q = 2.51e8$	**99.29**, $\sigma = 0.48$	**99.29**, $\sigma = 0.49$	**99.31**, $\sigma = 0.50$
Kernel DCV, $q = 1.67e7$	98.33, $\sigma = 0.66$	98.16, $\sigma = 0.55$	99.25, $\sigma = 0.51$

36 samples from each object class for training, and the remaining 36 samples for used testing. This process was repeated 15 times, and 15 different training and test sets were created. The first five data sets were used for parameter selection, and the rest were used for performance evaluation. The dimensionality of the sample space was reduced to 1439 through PCA, and then, samples were standardized before the application of the SVM classifier, as in the previous case. We set the SVM regularization parameter to $\gamma = 10$ for all kernels. The means and the standard deviations of computed recognition rates on the Coil-100 database are given in Table VII.

The MCV and kernel CV methods utilizing the Gaussian kernel achieved the best recognition accuracy among all methods as expected. In particular, the DCV method performed very poorly, which indicates that the object classes have different covariance structures. Both the kernel CLAFIC and kernel CV methods did not show any improvement over their linear counterparts. On the other hand, the kernel DCV method significantly outperformed the linear DCV method. Notice that all kernel methods yielded better results than the SVM classifier in all cases.

G. Discussion

The aforementioned classification experiments demonstrate various characteristics of our proposed methods and provide insights concerning suitable environments for the application. The kernel CV method did not show any superiority over the linear subspace classifier methods on the Iris database. Note that the number of training samples in each class is much larger than the dimensionality of the sample space for this database. Therefore, it is better to estimate distribution functions of classes for these situations. Then, more sophisticated classifiers can be constructed by using the estimated density functions. The proposed kernel subspace classifier achieved the highest recognition rate among all subspace classifier methods for the image segmentation, the Fourier coefficients, and the pixel averages databases. Moreover, the proposed method yielded the highest recognition rate along with the kernel CLAFIC for the wine database. Note that the number of samples in each class was not too large as compared to the dimensionality of the sample space for these classification tasks. The estimation of density functions may not be overly reliable for these cases since the dimensionality of the sample space and the number of samples per class are comparable. It has been reported that the number of samples in each class must be at least ten times the dimensionality of the sample space for a reliable density estimation [30], [31]. Thus, our proposed kernel subspace classifier may be a good choice in such cases.

For the AR face database in which the classes have similar covariance structures, both proposed methods achieved the same recognition rate, which was the highest recognition

TABLE VIII
TRAINING AND TEST TIMES (IN SECONDS) OF THE NONLINEAR METHODS USING THE GAUSSIAN KERNEL FUNCTIONS (IS: IMAGE SEGMENTATION; FC: FOURIER COEFFICIENTS; PA: PIXEL AVERAGES; OAO: ONE AGAINST ONE; OAA: ONE AGAINST ALL)

Methods	Kernel CLAFIC		Kernel CV		Kernel DCV		SVM-OAO			SVM-OAA		
Databases	Training Time	Testing Time	Training Time	Testing Time	Training Time	Testing Time	Training Time	Testing Time	Number of SVs	Training Time	Testing Time	Number of SVs
Iris	0.06s	0.01s	0.5s	0.01s	0.5s	0.01s	0.3s	0.01s	38-38	0.79s	0.01s	71-49
Wine	0.07s	0.02s	0.75s	0.02s	0.96s	0.02s	0.43s	0.01s	98-79	1.11s	0.02s	160-93
IS	1.85s	0.09s	141s	0.11s	148s	0.10s	7.46s	0.05s	664-295	28.7s	0.05s	553-323
FC	1.13s	0.09s	146s	0.09s	149s	0.09s	25s	0.11s	4085-733	617s	0.10s	2312-792
PA	1.17s	0.20s	151s	0.23s	160s	0.21s	24s	0.33s	2660-924	303s	0.31s	1385-933

rate among all subspace classifiers. However, the computed recognition rate was inferior to the rates that were obtained by the DCV and kernel DCV methods, which are more suitable methods for data sets having similar intraclass variations. Similarly, both proposed methods yielded the best recognition accuracy among all methods for the Coil-100 database, where the classes have typically different covariance structures. Note that the dimensionality of the sample space is too large as compared to the number of samples for these situations. On the other hand, the MCV method was outperformed by the kernel CV method on the pixel averages database because of the low dimensionality of the original sample space. It has also been observed that the Gaussian kernels give rise to better recognition rates than the polynomial kernels. The dimensionality of the kernel PCA-transformed space that was obtained using the Gaussian kernel was typically higher than the dimensionality that was obtained by the polynomial kernels. All these results are a verification of the fact that the generalization performances of the proposed methods depend on the dimensionality of the sample space in that higher dimensions give better recognition rates. Therefore, we recommend the use of the CV method in recognition tasks with classes having different covariance structures in high-dimensional sample spaces since the kernel CV method is computationally more expensive than the linear CV method. However, if the dimensionality of the sample space is not large enough for satisfactory recognition rates, one should instead use the kernel CV subspace classifier with the Gaussian kernel.

We gave the training and test times of the nonlinear methods for Gaussian kernels in Table VIII. Additionally, the number of total extracted support vectors (SVs) and the number of distinct SVs among them are provided in the table. The regularization parameters of SVM classifiers were available during calculations; thus, one needs additional training time to tune these parameters. Note that we did not give the training and test times on the AR and Coil-100 databases since the SVM classifier was not directly applicable for these data sets. As can be seen in the table, when the number of classes is increased, the training computational complexity of the SVM classifier increases proportionally. Notice that the training times of the SVM classifiers are significantly different for the image segmentation and pixel averages databases, although they have approximately the same number of data samples. On the other hand, the training times of other kernel methods are similar for those databases. It is because the SVM classifier was originally developed for two classes, and adopting it to multiclasses is time consuming [28]. More precisely, the OAO procedure, which was used to extend the SVM classifier to the multiclass problems that were addressed in this paper, requires solving a quadratic optimization problem for $C(C-1)/2$ classifiers, and one must adjust the regularization parameter for each binary classifier. Similarly, the one-against-all (OAA) procedure constructs C classifiers and solves a quadratic optimization problem for each classifier using all data samples. As opposed to the SVM classifier, the proposed kernel CV subspace classifier requires an eigendecomposition of a positive semidefinite matrix, which is easier to solve for moderate sizes. Additionally, our proposed kernel method is directly applicable to the multiclass classification problems, and one does not need to optimize any parameters in this approach. For our proposed nonlinear method, kernel function evaluations that are carried out to form the kernel matrix and eigendecomposition of this matrix constitute the major computational burden of the training phase. The extra computations of the proposed method, which are performed to obtain the null spaces in the transformed kernel PCA space, are negligible when compared to the aforementioned computations. In terms of the training time, assuming that the regularization parameters of SVM classifiers are available, our proposed method was faster than the SVM classifier using OAA procedure for all databases except the image segmentation database. However, the SVM classifier using OAO procedure was more efficient than the proposed method in all cases. Although the OAO procedure generally constructs more classifiers than the OAA procedure, each classifier uses data samples coming from two classes rather than using all data samples. As a result, the OAO procedure works on smaller matrices, which in turn speeds up the training phase. However, it should be taken into account that the SVM algorithm must run several times to compute the best regularization parameters for both approaches. Testing time is the time that is required to classify a given sample. In our proposed nonlinear method, we have to compute the feature vector given in (26) to classify a test sample. This computation requires M online kernel evaluations for each sample in the training set. For the SVM classifiers, the number of kernel evaluations is determined by the extracted SVs. As shown in Table VIII, the number of extracted SVs is less than the total number of data samples in the training set for the Iris, wine, and image segmentation databases, where the number of classes

is less than 10. As a result, SVM classifiers are more efficient than the pr oposed method in terms of testing time for those data sets. On the other hand, as the number of classes is increased, the total number of extracted SVs increases proportionally. Notice that the total number of extracted SVs is bigger than the training set size for the Fourier coefficients and the pixel averages databases, which indicates that some data samples are used as SVs for more than one classifier in the SVM scheme. The testing times of SVM classifier are similar to our proposed method for those databases, as shown in the table. When the number of total extracted SVs is bigger than the training set size, one has to determine unique SVs among all extracted SVs to prevent unnecessary kernel evaluations. Therefore, we have to employ lookup tables during testing to call required kernel evaluations, which brings additional computation. Another interesting observation is that as the number of classes is increased, more data samples contribute as SVs, which makes the SVM classifier less appealing for data sets having large number of classes.

In general, experimental results confirm that the intersection of the range space of the total scatter matrix and the null space of each individual class scatter matrix is typically the best subspace for discrimination. In addition, it is not necessary to optimize parameters such as the subspace dimensions or the parameters α and β. Therefore, the proposed methods are more straightforward and practical as compared to the other subspace classifiers mentioned in this paper. The computational loads of the training and testing times of the linear CV and CLAFIC methods are approximately same, and these methods are more efficient than all discussed kernel methods in terms of training and test times. The computational comparison that is the most interesting to us is between the kernel CLAFIC and kernel CV methods. In terms of training time, the kernel CLAFIC method is more efficient than our proposed kernel method. In fact, the kernel CLAFIC is the most efficient method among all nonlinear methods in terms of training time. It is because C different $N \times N$ kernel matrices are formed and C eigendecompositions are carried out on these smaller matrices in the kernel CLAFIC method, as opposed to the kernel CV method, in which the eigendecomposition is performed on a unique $M \times M$ kernel matrix. On the other hand, the testing time of the kernel CLAFIC method is only slightly better than the proposed method since the kernel CLAFIC method also requires M kernel function evaluations during the testing phase. Experimental results demonstrate that the proposed method typically yields better results than the kernel CLAFIC method. It should be noticed that the transformed space is constructed only using class specific samples in the kernel CLAFIC method. As a result, the relations among classes are ignored. In contrast, the transformed space is more discriminative in our proposed scheme since all pairwise distances are utilized during the kernel PCA step.

The kernel DCV method also includes the kernel PCA step, just as in our proposed method. Since this step constitutes the major computational burden of the algorithm, the training times of the kernel CV and kernel DCV methods are similar, as illustrated in Table VIII. Similarly, both methods require M kernel evaluations during testing phase. Thus, the testing times of those methods are also similar, as given in Table VIII.

As mentioned earlier, kernel evaluations and eigendecomposition of an $M \times M$ kernel matrix constitute the main computational burden of our kernel approach. Furthermore, the eigendecomposition of the kernel matrix may be problematic for large M. There are several approaches to cope with this pitfall. First, we can use a sparse approximation of the kernel matrix, which sufficiently describes the dominant eigenvalues, as described in [32]. A second solution would be to discard some dependent samples so as to reduce the training set size M. Finally, we can apply the proposed method locally. In this scheme, we obtain the closest prototypes to a test sample by using a crude distance metric (e.g., Euclidean distance) in the original sample space. Then, the proposed nonlinear method can be applied to the closest prototype samples.

V. Conclusion

In this paper, we proposed a new nonlinear subspace classifier method, which extends the linear CV method to the nonlinear case using the kernel trick. However, the nonlinearization of the CV method was not trivial. The CV method needed to be modified before the nonlinearization. To represent classes, the proposed nonlinear method employs the intersection subspace of the null space of the covariance matrix of each class and the range space of the covariance matrix of pooled data. When the training set samples are projected onto these intersection subspaces, all training set samples in each class give rise to a unique vector, which is called a CV. Thus, under certain conditions, a 100% recognition rate is guaranteed for the training set samples. Our test results show that the generalization ability of the proposed method compares favorably with other linear subspace classifier methods and also the kernel-based nonlinear subspace method. In particular, the proposed method yields good recognition results when the number of samples is not sufficient for reliable density estimation, which is widely encountered in real-world recognition tasks. Therefore, the proposed method can find broad application areas in pattern recognition field.

Appendix I
Proof of the Theorem

Before we give the Proof of Theorem 1, we need the following auxiliary lemma.

Lemma 1: Let $H^{(1)}$ and $H^{(2)}$ be subspaces of $\Re^d$, $H^{(1)^\perp}$ and $H^{(2)^\perp}$ be their orthogonal complements, and P_1 and P_2 be the orthogonal projection matrices onto $H^{(1)}$ and $H^{(2)}$, respectively. If $H^{(1)^\perp} \perp H^{(2)^\perp}$, then P_1 and P_2 commute, that is, $P_1P_2 = P_2P_1$.

Proof: If $H^{(1)^\perp} \perp H^{(2)^\perp}$, then $(I - P_1)(I - P_2) = 0$ and $(I - P_2)(I - P_1) = 0$, where I is the identity matrix. Thus

$$(I - P_1)(I - P_2) = (I - P_2)(I - P_1) = 0 \quad (27)$$

$$I - P_1 - P_2 - P_1P_2 = I - P_1 - P_2 - P_2P_1 \quad (28)$$

which implies that $P_1P_2 = P_2P_1$. ■

Theorem 1: Let P and $P_{NS}^{(i)}$ be the projection matrices of the subspaces $R(S_T)$ and $N(S_i)$, $i = 1, \ldots, C$, respectively. Then, P and $P_{NS}^{(i)}$ commute, i.e.,

$$P_{NS}^{(i)} P = P P_{NS}^{(i)}, \qquad i = 1, \ldots, C.$$

Proof: Let $H^{(1)} = R(S_T)$ and, for any fixed i, let $H^{(2)} = N(S_i)$. Clearly, $H^{(1)^\perp} = N(S_T)$ and $H^{(2)^\perp} = R(S_i)$. Let S_B denote the between-class scatter matrix, which is defined as $S_B = \sum_{i=1}^{C} N_i(\mu_i - \mu)(\mu_i - \mu)^T$. By using [19, Lemma 1]

$$\begin{aligned} N(S_T) &= N(S_B + S_1 + \ldots + S_C) \\ &= N(S_B) \cap N(S_1) \cap \ldots \cap N(S_C) \end{aligned} \tag{29}$$

and, in particular, $N(S_T) \subset N(S_i)$, which, together with the fact that $N(S_i) \perp R(S_i)$, shows that

$$N(S_T) \perp R(S_i) \quad \text{or} \quad H^{(1)^\perp} \perp H^{(2)^\perp}. \tag{30}$$

The assertion of the theorem now follows from Lemma 1. ■

APPENDIX II
KERNEL MATRIX AND VECTOR FORMULAS

The kernel matrix $K = \Phi^T \Phi = (K^{ij})_{\substack{i=1,\ldots,C \\ j=1,\ldots,C}}$ of the mapped samples can be displayed as

$\mathbf{K}$	1	2	$\cdots$	M
1	$k\left(x_1^1, x_1^1\right)$	$k\left(x_1^1, x_2^1\right)$	$\cdots$	$k\left(x_1^1, x_{N_C}^C\right)$
2	$k\left(x_2^1, x_1^1\right)$	$k\left(x_2^1, x_2^1\right)$	$\cdots$	$k\left(x_2^1, x_{N_C}^C\right)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
M	$k\left(x_{N_C}^C, x_1^1\right)$	$k\left(x_{N_C}^C, x_2^1\right)$	$\cdots$	$k\left(x_{N_C}^C, x_{N_C}^C\right)$

where each submatrix $K^{ij} \in \Re^{N_i \times N_j}$ is

$\mathbf{K}^{ij}$	1	2	$\cdots$	N_j
1	$k\left(x_1^i, x_1^j\right)$	$k\left(x_1^i, x_2^j\right)$	$\cdots$	$k\left(x_1^i, x_{N_j}^j\right)$
2	$k\left(x_2^i, x_1^j\right)$	$k\left(x_2^i, x_2^j\right)$	$\cdots$	$k\left(x_2^i, x_{N_j}^j\right)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
N_i	$k\left(x_{N_i}^i, x_1^j\right)$	$k\left(x_{N_i}^i, x_2^j\right)$	$\cdots$	$k\left(x_{N_i}^i, x_{N_j}^j\right)$

Similarly, the matrix $K^{(i)} = \Phi^T \Phi^{(i)} = (K^{(i)j})_{j=1,\ldots,C} \in \Re^{M \times N_i}$ can be displayed as

$\mathbf{K}^{(\mathbf{i})}$	1	2	$\cdots$	N_i
1	$k\left(x_1^1, x_1^i\right)$	$k\left(x_1^1, x_2^i\right)$	$\cdots$	$k\left(x_1^1, x_{N_i}^i\right)$
2	$k\left(x_2^1, x_1^i\right)$	$k\left(x_2^1, x_2^i\right)$	$\cdots$	$k\left(x_2^1, x_{N_i}^i\right)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
M	$k\left(x_{N_C}^C, x_1^i\right)$	$k\left(x_{N_C}^C, x_2^i\right)$	$\cdots$	$k\left(x_{N_C}^C, x_{N_i}^i\right)$

where each submatrix $K^{(i)j} \in \Re^{N_j \times N_i}$ is

$\mathbf{K}^{(\mathbf{i})\mathbf{j}}$	1	2	$\cdots$	N_i
1	$k\left(x_1^j, x_1^i\right)$	$k\left(x_1^j, x_2^i\right)$	$\cdots$	$k\left(x_1^j, x_{N_i}^i\right)$
2	$k\left(x_2^j, x_1^i\right)$	$k\left(x_2^j, x_2^i\right)$	$\cdots$	$k\left(x_2^j, x_{N_i}^i\right)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
N_j	$k\left(x_{N_j}^j, x_1^i\right)$	$k\left(x_{N_j}^j, x_2^i\right)$	$\cdots$	$k\left(x_{N_j}^j, x_{N_i}^i\right)$

The vector $l_m^i \in \Re^M$ can be displayed as

$\mathbf{l}_\mathbf{m}^\mathbf{i}$	1
1	$k\left(x_1^1, x_m^i\right)$
2	$k\left(x_2^1, x_m^i\right)$
$\vdots$	$\vdots$
M	$k\left(x_{N_C}^C, x_m^i\right)$

REFERENCES

[1] E. Oja, *Subspace Methods of Pattern Recognition*. New York: Res. Stud. Press, 1983.

[2] J. Laaksonen, "Subspace classifiers in recognition of handwritten digits," Ph.D. dissertation, Helsinki Univ. Technol., Espoo, Finland, 1997.

[3] S.-W. Kim and B. J. Oommen, "On utilizing search methods to select subspace dimensions for kernel-based nonlinear subspace classifiers," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 1, pp. 136–141, Jan. 2005.

[4] S. Watanabe, P. F. Lambert, C. A. Kulikowski, J. L. Buxton, and R. Walker, "Evaluation and selection of variables in pattern recognition," in *Computer and Information Sciences II*. New York: Academic, 1967, p. 91.

[5] S. Watanabe and N. Pakvasa, "Subspace method in pattern recognition," in *Proc. 1st Int. Conf. Pattern Recog.*, Washington, DC, 1973, pp. 25–32.

[6] K. Fukunaga and W. L. Koontz, "Application of the Karhunen–Loeve expansion to feature selection and ordering," *IEEE Trans. Comput.*, vol. C-19, no. 4, pp. 311–318, Apr. 1970.

[7] T. Kohonen, G. Nemeth, K. J. Bry, M. Jalanko, and H. Riittinen, "Spectral classification of phonemes by learning subspaces," in *Proc. 5th Int. Conf. Acoust., Speech, Signal Process.*, Washington, DC, 1979, pp. 97–100.

[8] M. Kuusela and E. Oja, "The averaged learning subspace method for spectral pattern recognition," in *Proc. 6th Int. Conf. Pattern Recog.*, Munchen, Germany, 1982, pp. 134–137.
[9] J. Laaksonen and E. Oja, "Subspace dimension selection and averaged learning subspace method in handwritten digit classification," in *Proc. ICANN*, Bochum, Germany, Jul. 1996, pp. 227–232.
[10] C. M. Bishop, *Neural Networks for Pattern Recognition*. New York: Oxford Univ. Press, 1995, pp. 7–9.
[11] M. Turk and A. P. Pentland, "Eigenfaces for recognition," *J. Cogn. Neurosci.*, vol. 3, no. 1, pp. 71–86, 1991.
[12] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Ann. Eugen.*, vol. 7, no. 2, pp. 179–188, 1936.
[13] M. B. Gulmezoglu, V. Dzhafarov, M. Keskin, and A. Barkana, "A novel approach to isolated word recognition," *IEEE Trans. Speech Audio Process.*, vol. 7, no. 6, pp. 620–628, Nov. 1999.
[14] M. B. Gulmezoglu, V. Dzhafarov, and A. Barkana, "The common vector approach and its relation to principal component analysis," *IEEE Trans. Speech Audio Process.*, vol. 9, no. 6, pp. 655–662, Sep. 2001.
[15] H. Cevikalp, B. Barkana, and A. Barkana, "A comparison of the common vector and the discriminative common vector methods for face recognition," in *Proc. 9th World Multi-Conf. Systemics, Cybern. and Inf.*, Orlando, FL, 2005.
[16] T. Balachander and R. Kothari, "Kernel based subspace pattern classification," in *Proc. Int. Joint Conf. Neural Netw.*, 1999, vol. 5, pp. 3119–3122.
[17] K. Tsuda, "Subspace classifier in reproducing kernel Hilbert space," in *Proc. Int. Joint Conf. Neural Netw.*, 1999, vol. 5, pp. 3454–3457.
[18] B. Schölkopf, A. J. Smola, and K. R. Muller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Comput.*, vol. 10, no. 5, pp. 1299–1319, Jul. 1998.
[19] H. Cevikalp, M. Neamtu, M. Wilkes, and A. Barkana, "Discriminative common vectors for face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 1, pp. 4–13, Jan. 2005.
[20] H. Cevikalp, M. Neamtu, and M. Wilkes, "Discriminative common vectors method with kernels," *IEEE Trans. Neural Netw.*, vol. 17, no. 6, pp. 1550–1565, Nov. 2006.
[21] J. Xu and L. Zikatanov, "The method of alternating projections and the method of subspace corrections in Hilbert space," *J. Amer. Math. Soc.*, vol. 15, no. 3, pp. 573–597, 2002.
[22] A. M. Martinez and R. Benavente, "The AR face database," Comput. Vis. Center, Barcelona, Spain, CVC Tech. Rep. 24, 1998.
[23] S. A. Nene, S. K. Nayar, and H. Murase, "Columbia Object Image Library (COIL-100)," Columbia Univ., New York, Tech. Rep., CUCS-006-96, Feb. 1996.
[24] *UCI-Benchmark Repository—A huge collection of artificial and real word data sets*, Univ. California Irvine. [Online]. Available: http://www.ics.edu/~mlearn/MLRepository.html
[25] F. Perez-Cruz and O. Bousquet, "Kernel methods and their potential use in signal processing," *IEEE Signal Process. Mag.*, vol. 21, no. 3, pp. 57–65, May 2004.
[26] Y. Ariki and Y. Motegi, "Segmentation and recognition of hand written characters using subspace method," in *Proc. 3rd Int. Conf. Doc. Anal. and Recog.*, Montreal, PQ, Canada, 1995, pp. 120–123.
[27] C.-W. Hsu and C.-J. Lin, "A comparison of methods for multiclass support vector machines," *IEEE Trans. Neural Netw.*, vol. 13, no. 2, pp. 415–425, Mar. 2002.
[28] G. Baudat and F. Anouar, "Generalized discriminant analysis using a kernel approach," *Neural Comput.*, vol. 12, no. 10, pp. 2385–2404, Oct. 2000.
[29] M. van Breukelen, R. P. W. Duin, D. M. J. Tax, and J. E. den Hartog, "Handwritten digit recognition by combined classifiers," *Kybernetika*, vol. 34, no. 4, pp. 381–386, 1998.
[30] A. K. Jain, R. P. W. Duin, and J. Mao, "Statistical pattern recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 1, pp. 4–37, Jan. 2000.
[31] L. O. Jimenez and D. A. Landgrebe, "Supervised classification in high dimensional space: Geometrical, statistical and asymptotical properties of multivariate data," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 28, no. 1, pp. 39–54, Feb. 1998.
[32] A. J. Smola and B. Schölkopf, "Sparse greedy matrix approximation for machine learning," in *Proc. ICML*, P. Langley, Ed., 2000, pp. 911–918.

Hakan Cevikalp (S'01–M'06) received the M.S. degree from Eskisehir Osmangazi University, Eskisehir, Turkey, in 2001 and the Ph.D. degree from Vanderbilt University, Nashville, TN, in 2005.

He was a Postdoctoral Researcher in the Learning and Recognition in Vision (LEAR), French National Institute for Research in Computer Science and Control (INRIA) Rhone-Alpes, Saint Ismier, France. He is currently with the Department of Electrical and Electronics Engineering, Eskisehir Osmangazi University. His research interests include pattern recognition, neural networks, image and signal processing, optimization, and computer vision.

Marian Neamtu received the M.S. degree in mechanical engineering from the Slovak Technical University, Bratislava, Slovakia, in 1988 and the Ph.D. degree in mathematics from the University of Twente, Enschede, The Netherlands, in 1991.

He is currently an Associate Professor of mathematics at Vanderbilt University, Nashville, TN. His main research interests are in numerical analysis approximation theory, computer-aided geometric design, and related areas of applied mathematics.

Atalay Barkana received the B.S. degree in electrical engineering from Robert College, Istanbul, Turkey, in 1969 and the M.S. and Ph.D. degrees in electrical engineering from the University of Virginia, Charlottesville, in 1971 and 1974, respectively.

From 1974 to 1986, he worked on linear and nonlinear theory. Since 2005, he has been in the Department of Electrical and Electronics Engineering, Anadolu University, Eskisehir, Turkey, where he is currently a Professor. His current interests include speech recognition, pattern analysis, neural networks, and statistical signal processing.

Dr. Barkana is a member of the IEEE Signal Processing Society.