Smart Colonography for Distributed Medical Databases with Group Kernel Feature Analysis

YUICHI MOTAI, Virginia Commonwealth University
DINGKUN MA, Northwestern Polytechnic University
ALEN DOCEF, Virginia Commonwealth University
HIROYUKI YOSHIDA, Massachusetts General Hospital and Harvard Medical School

Computer-Aided Detection (CAD) of polyps in Computed Tomographic (CT) colonography is currently very limited since a single database at each hospital/institution doesn't provide sufficient data for training the CAD system's classification algorithm. To address this limitation, we propose to use multiple databases, (e.g., big data studies) to create multiple institution-wide databases using distributed computing technologies, which we call smart colonography. Smart colonography may be built by a larger colonography database networked through the participation of multiple institutions via distributed computing. The motivation herein is to create a distributed database that increases the detection accuracy of CAD diagnosis by covering many true-positive cases. Colonography data analysis is mutually accessible to increase the availability of resources so that the knowledge of radiologists is enhanced. In this article, we propose a scalable and efficient algorithm called Group Kernel Feature Analysis (GKFA), which can be applied to multiple cancer databases so that the overall performance of CAD is improved. The key idea behind the proposed GKFA method is to allow the feature space to be updated as the training proceeds with more data being fed from other institutions into the algorithm. Experimental results show that GKFA achieves very good classification accuracy.

Categories and Subject Descriptors: I.5 [**Pattern Recognition**]: Applications

General Terms: Algorithms

Additional Key Words and Phrases: Computed tomographic colonography, distributed databases, kernel feature analysis, group learning

ACM Reference Format:
Yuichi Motai, Dingkun Ma, Alen Docef, and Hiroyuki Yoshida. 2015. Smart colonography for distributed medical databases with group kernel feature analysis. ACM Trans. Intell. Syst. Technol. 6, 4, Article 58 (July 2015), 24 pages.
DOI: http://dx.doi.org/10.1145/2668136

58

This work is supported partly by the CAREER award 1054333 from National Science Foundation, CTSA UL1TR000058 from the National Center for Advancing Translational Sciences, Center for Clinical and Translational Research Endowment Fund of Virginia Commonwealth University (VCU), Institutional Research Grant IRG-73-001-31 from the American Cancer Society through the Massey Cancer Center, Presidential Research Incentive Program at VCU, and National Institutes of Health Grants CA095279 and CA166816.
Authors' addresses: Y. Motai (corresponding author), D. Ma, and A. Docef, Department of Electrical and Computer Engineering, Virginia Commonwealth University, 601 West Main Street, Richmond, VA 23284, USA; emails: ymotai@vcu.edu, madingkun@gmail.com, adocef@vcu.edu; H. Yoshida, Department of Radiology, Massachusetts General Hospital and Harvard Medical School, 25 New Chardon Street, Suite 400C, Boston, MA 02114, USA; email: Yoshida.Hiro@mgh.harvard.edu.

DOI: http://dx.doi.org/10.1145/2668136

1. INTRODUCTION

The major concern in traditional medical screening is that it is a limited evaluation by a physician who may diagnose patients based on that physician's knowledge. In the case of colon cancer, although many patients received conventional endoscopic screening, an estimated 50,310 deaths (colon cancer is the second leading cause of cancer death in the United States) are expected to occur in 2014 [American Cancer Society 2014]. For screening of colon cancer, Computed Tomographic (CT) colonography is emerging as an attractive alternative to more invasive colonoscopy because it can find precursor benign polyps that can be removed before cancer has had a chance to develop from them [Levin et al. 2008; Yee et al. 2014]. Improvements that reduce diagnostic error would go a long way toward making CT colonography a more acceptable technique for colon examination. To be a clinically practical means of screening colon cancers, CT colonography must be able to interpret a large number of images in a time-efficient fashion, and it must facilitate the detection of polyps with high accuracy. Currently, however, interpretation of CT colonography is handled by a limited number of specialists at individual hospitals/institutions, and reader performance for polyp detection varies substantially [Rockey et al. 2005]. To overcome these difficulties while providing accurate detection of polyps, Computer-Aided Detection (CAD) schemes are investigated that semi-automatically detect suspicious lesions in CT colonography images [Regge and Halligan 2013; Yoshida and Dachman 2005].

The state-of-the art of CAD is emerging as CT colonography gains popularity for screening of colon cancer. Numerical schemes of image analysis have been developed for individual institutions, where resources and training requirements determine the number of training instances. Thus, if more training data are collected after the initial tumor model is computed, retraining of the model becomes imperative in order to incorporate data from other institutions and to preserve or improve classification accuracy [Yoshida and Nappi 2007]. Thus, we propose a new framework called distributed colonography, in which the colonography database at each institution may be shared and/or uploaded to a common server. The CAD system at each institution can be enhanced by incorporating new data from other institutions using the distributed learning model proposed in this article.

The concept of distributed colonography using networked distributed databases has been discussed in many classification applications, but not yet in the context of CAD in CT colonography [Yoshida et al. 2012]. These existing studies showed that the overall classification performance for larger multiple databases was improved in practical settings [Chang et al. 2008; Khan et al. 2013]. Rather than applying traditional techniques of classification to a very limited number of patients, medical data from multiple institutions can be explored.

The utilization of the proposed distributed colonography framework shown in Figure 1 requires a study to determine whether the overall performance is improved by using multiple databases. The presented work is a first attempt at such a study. The benefit for clinical practice is that different characteristics of CT colonography datasets, which may not exist at a single institution, will be available. Thus, both the CAD algorithm and clinicians can observe and utilize many of the potential cases in the proposed distributed platform.

The primary focus of this article is to find effective ways to associate multiple databases to represent statistical data characteristics. Few existing classification techniques using distributed databases successfully handle big data structures. The new classification method is expected to be capable of learning multiple large databases specifically tailored to the big data of CT colonography. Thus, we propose composite kernel feature analysis to deal with the effective compression of big data from multiple

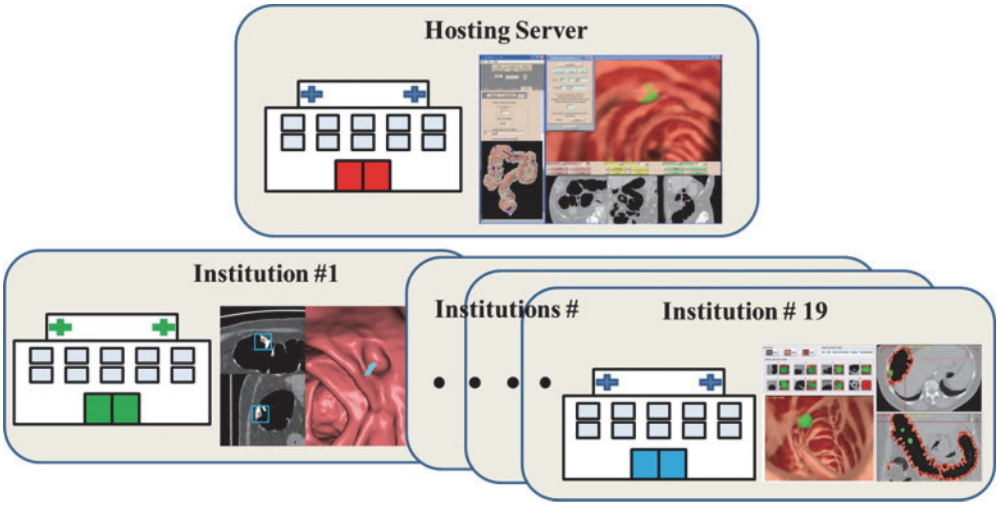

Fig. 1. Distributed colonography with distributed image databases for colon cancer diagnosis. The hosting server collects and analyzes databases from different institutions and groups them into assembled databases, and individual institutions can access them through various interfaces.

databases. The improvement in performance that could be obtained by employing the proposed approach requires a very small additional investment in terms of resources, infrastructure, and operational costs.

We propose Group Kernel Feature Analysis (GKFA) for distributed databases for the group learning method. GKFA can efficiently differentiate polyps from false positives and thus can be expected to improve detection performance. The key idea behind the proposed GKFA method is to allow the feature space to be updated as the training proceeds with more data being fed from other institutions into the algorithm. The feature space can be reconstructed by GKFA by grouping multiple databases. The feature space is augmented with new features extracted from the new data, with the feature space expanded if necessary. We present the first comparative study of these methods and show that the proposed GKFA outperforms existing nonlinear dimensionality reduction methods when different databases for CT colonography become necessary.

The contribution of this study is that the proposed GKFA method works in distributed CT colonography databases. These databases, acquired over a long period of time, can sometimes be highly diverse, and each database is unique in nature; therefore, obtaining a clear distinction among multiple databases is a very challenging task. There is a chance of misinterpreting the database to be either homogeneous or heterogeneous in nature while training the new incoming databases from many institutions. The method was tested using real CT colonography data to show that the proposed GKFA improves CAD performance while achieving a feature space that is comparably similar to the feature space obtained by a separate learning method at each institution.

The rest of this article is organized as follows. Section 2 provides an introduction to kernel methods and a brief review of the existing kernel-based feature extraction method, Kernel Principal Component Analysis (KPCA). In Section 3, we discuss homogeneous and heterogeneous groupings of database subsets to deal with the huge volume of incoming data. Section 4 describes the proposed GKFA for polyp candidates from multiple databases, Section 5 evaluates the experimental results, and conclusions are drawn in Section 6.

2. KERNEL PRINCIPAL COMPONENT ANALYSIS (KPCA)

In the generic field of pattern recognition, there are many existing feature selection methods for classifiers such as stepwise feature selection [Taimouri et al. 2011] and manifold-based methods [Hu et al. 2004]. It is not certain if these existing methods are effective when used in the proposed distributed CT colonography framework. The effective application of group learning using multiple databases to nonlinear spaces is undertaken using kernel-based methods. Kernel-based feature extraction methods tend to perform better than non-kernel-based methods since the actual databases have very nonlinear characteristics. Another issue under consideration is the approach used to handle the larger sized databases obtained by combining multiple databases. Zheng et al. [2005] proposed that the input data be divided into a few groups of similar size and KPCA be applied to each group. A set of eigenvectors was obtained for each group, and the final set of features was obtained by applying KPCA to a subset of these eigenvectors. The application of KPCA is a promising method of compressing all the databases and extracting the salient features (principal components) [Kivinen et al. 2004; Ozawa et al. 2008; Zhao et al. 2006; Li 2004; Kim 2007]. KPCA has already shown computational effectiveness in many image processing applications and pattern classification systems [Kim et al. 2003; Kim and Kim 2003; Hoegaerts et al. 2007; Chin and Suter 2007].

For efficient feature analysis, the extraction of the salient features of a polyp is essential because of the size and 3D nature of the polyp databases [Näppi and Yoshida 2002, 2003]. Moreover, the distribution of the image features of polyps is nonlinear. The problem is how to select a nonlinear, positive-definite kernel $K: R^d \times R^d \rightarrow R$ of an integral operator in the d-dimensional space. The kernel K, which is a Hermitian and positive semi-definite matrix, calculates the inner product between two finite sequences of inputs $\{x_i : i \in n\}$ and $\{x_j : j \in n\}$, defined as $K := (K(x_i, x_j)) = (\Phi(x_i).\Phi(x_j) : i, j \in n)$. Here, x is a gray-level CT image, n is the number of image databases, and $\Phi: R^d \rightarrow H$ denotes a nonlinear embedding (induced by K) into a possibly infinite dimensional Hilbert space H. Some of the commonly used kernels are the Linear Kernel, the Polynomial Kernel, the Gaussian Radial Basis Function (RBF) Kernel, the Laplace RBF Kernel, the Sigmoid Kernel, and the ANOVA RB Kernel [Schölkopf and Smola 2002].

Kernel selection is heavily dependent on data specifics. For instance, the linear kernel is important in large, sparse data vectors, and it implements the simplest of all kernels, whereas the Gaussian and Laplace RBFs are general-purpose kernels used when prior knowledge about data is not available. The Gaussian kernel avoids the sparse distribution, which is obtained when a high-degree polynomial kernel is used. The polynomial kernel is widely used in image processing, while the ANOVA RBF is usually adopted for regression tasks. A more thorough discussion of kernels can be found in Schölkopf and Smola [2002], Fröhlich et al. [2003], Chen [2003], Park and Cho [2003], and Sadjadi [2008]. Our GKFA for CT colonographic images is a dynamic extension of KPCA following Kim and Kim [2003], Kim et al. [2004], Hoegaerts et al. [2007], Chin and Suter [2007], Jiang et al. [2006], and Jayawardhana et al. [2009].

KPCA uses a Mercer kernel to perform a linear principal component analysis of the transformed image. Without loss of generality, we assume that the image of the data has been centered so that its scatter matrix in S is given by $S = \sum_{i=1} \Phi(x_i)(x_i)\Phi(x_i)^T$. The eigenvalues λ_j and eigenvectors e_j are obtained by solving the following equation, $\lambda_j e_j = S e_j = \sum_{i=1} \Phi(x_i)\Phi(x_i)^T e_j = \sum_{i=1} < e_j, \Phi(x_i) > \Phi(x_i)$. If K is an $n \times n$ Gram matrix, with the element $k_{ij} = < \Phi(x_i), \Phi(x_j) >$, and $a_j = [a_{j1}, a_{j2}, \ldots a_{jn}]$ are the eigenvectors associated with eigenvalues λ_j, then the dual eigenvalue problem equivalent to the problem can be expressed as $\lambda_j a_j = K a_j$.

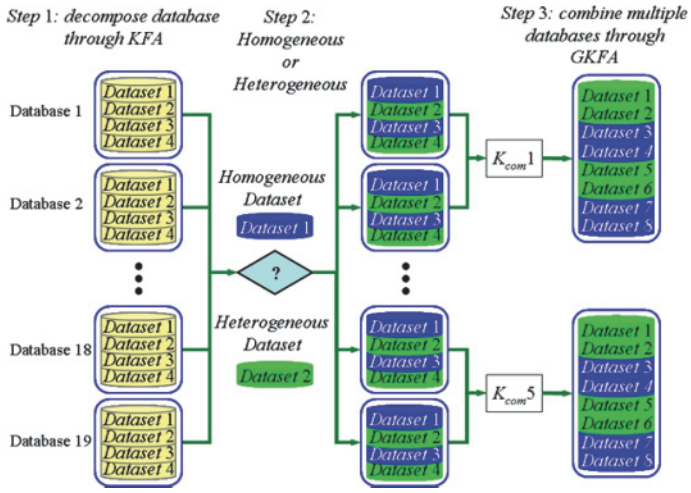

Fig. 2. The concept of Group Kernel Feature Analysis. The proposed criteria are to determine the nature of database by (1) decomposition, (2) classification by heterogeneity, and (3) combination.

KPCA can now be presented as follows:

1. Compute the Gram matrix that contains the inner products between pairs of image vectors.
2. Solve $\lambda_j a_j = K a_j$ to obtain the coefficient vectors a_j for $j = 1, 2, \ldots, n$.
3. The projection of a test point $x \in R^d$ along the j-th eigenvector is $< e_j, \Phi(x) > = \sum_{i=1}^{n} a_{ji} < \Phi(x_i), \Phi(x) > = \sum_{i=1}^{n} a_{ji} k(x, x_i)$.

This method implicitly contains an eigenvalue problem of rank n, so the computational complexity of KPCA is $O(n^3)$. The total computational complexity is given by $O(ln^2)$, where l stands for the number of features to be extracted and n stands for the rank of the Gram matrix K [Jiang et al. 2006; Jayawardhana et al. 2009]. Once the Gram matrix is computed, we can apply these algorithms to our database to obtain a higher dimensional feature space. This idea is discussed in the following sections.

3. KERNEL FEATURE ANALYSIS (KFA) FOR DISTRIBUTED DATABASES

Non-shareable data have not yet been addressed in any clinical applications of colonography CAD. In the current environment, the problem is that each platform is independently operated in a closed manner—that is, none of the CAD platforms contributes to other CAD platforms. To avoid this limitation, the proposed distributed databases for colonography attempt to make larger data-driven CAD more prominent through the use of data aggregation. To handle data aggregation by synthesizing each platform, instead of handling data independently, we herein propose a machine learning technique called GKFA that adjusts the classification criteria by extending KPCA for distributed colonography databases. We want to validate the performance of GKFA when applied to a distributed database, specifically CT colonography.

As shown in Figure 2, we introduce the concept of training the algorithm by analyzing the data received from other databases. Step 1 in Figure 2 illustrates the decomposition of each database through KFA. Each database consists of several datasets. For example Database 1 is decomposed into four datasets. We will describe Step 1 in Section 3.1 and Step 2 in Section 3.2 to reconstruct each database using KFA. Specifically, Section 3.1 describes how to extract the data-dependent kernels for each database using KFA.

In Section 3.2, we propose that each database be classified as either homogeneous or heterogeneous by the proposed criteria so that each database can be decomposed into a heterogeneous dataset. We will describe the details of Step 3 separately in Section 4 as GKFA.

3.1. Extract Data-Dependent Kernels Using KFA

We exploit the idea of the data-dependent kernel to select the most appropriate kernels for a given database. Let $\{x_i, x_j\}(i, j = 1, 2, \ldots, n)$ be n training samples of the given d-dimensional data, and $x_j = \{+1, -1\}$ represents the class labels of the samples (i.e., the data are labeled true-positive representing $x_j = \{+1\}$). A data-dependent kernel is adopted using a composite form as follows: the kernel k_l, for $l \in \{1, 2, 3, 4\}$ is formulated as $k_l(x_i, x_j) = q_l(x_i)q_l(x_j)p_l(x_i, x_j)$, where $x \in R^d$, $p_l(x_i, x_j)$ is one kernel among four chosen kernels. Here, $q_l(.)$ is the factor function, $q_l(x_i) = a_{l0} + \sum_{m=1}^{n} a_{lm}k_l(x_i, a_{lm})$, where $k_l(x_i, x_j)$ and α_{lm} are the combination coefficient. In matrix form, we can write $q_l = K_l a.$, where $q_l = \{q_l(x_1), q_l(x_2), \ldots, q_l(x_n)\}^T$ and K_l is an $n \times (n+1)$ matrix defined as:

$$K_l = \begin{pmatrix} 1 & k_l(x_1, a_{l1}) & \cdots & k_l(x_1, a_{ln}) \\ 1 & k_l(x_2, a_{l1}) & \cdots & k_l(x_2, a_{ln}) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & k_l(x_n, a_{l1}) & \cdots & k_l(x_n, a_{ln}) \end{pmatrix}. \tag{1}$$

Let the kernel matrices corresponding to kernels k_l, and p_l be K_l, and P_l. Therefore, we can denote the data-dependent kernel matrix K_l as $K_l = [q_l(x_i)q_l(x_j)p_l(x_i, x_j)]_{n\times(n+1)}$. Defining $Q_l = \{1, q_l(x_1), q_l(x_2), \ldots, q_l(x_n)\}^T$, we obtain $K_l = Q_l P_l Q_l^T$. We decompose each database by maximizing the Fisher scalar for our kernel optimization. The Fisher scalar is used to measure the class separability J of the training data in the mapped feature space. It is formulated as $J = \mathrm{tr}(\sum_{l=1} S_{bl})/\mathrm{tr}(\sum_{l=1} S_{wl})$, where S_{bl} represents "between-class scatter matrices," and S_{wl} represents "within-class scatter matrices."

Suppose that the training data are clustered; that is, the first n_1 data belong to one class (class label equals -1), and the remaining n_2 data belong to the other class (class label equals $+1$). Then, the basic kernel matrix P_l can be partitioned to represent each class shown as:

$$P_l = \begin{matrix} \\ \end{matrix} \begin{matrix} P_{11}^l & P_{12}^l \\ P_{21}^l & P_{22}^l \end{matrix}, \tag{2}$$

where P_{11}^l, P_{12}^l, P_{21}^l, and P_{22}^l are the submatrices of P_l in the order of $n_1 \times n_1$, $n_1 \times n_2$, $n_2 \times n_1$, $n_2 \times n_2$, respectively. According to Xiong et al. [2007], the class separability by Fisher scalar can be expressed as $J_l(a_l) = a_l^T M_l a_l / a_l^T N_l a_l$, where $M_l = K_l^T B_l K_l$, $N_l = K_l^T W_l K_l$, and

$$W_l = \mathrm{diag}\left(P_{11}^l, P_{22}^l\right) - \begin{matrix} P_{11}^l/n_1 & 0 \\ 0 & P_{22}^l/n_2 \end{matrix} \qquad B_l = \begin{matrix} P_{11}^l/n_1 & 0 \\ 0 & P_{22}^l/n_2 \end{matrix} - P_l/n. \tag{3}$$

To maximize $J_l(a_l)$, the standard gradient approach is followed. If the matrix N_{0i} is nonsingular, the optimal a_l that maximizes $J_l(a_l)$ is the eigenvector that corresponds to the maximum eigenvalue of $M_l a_l = \lambda_l N_l a_l$. The criterion to select the best kernel function is to find the kernel that produces the largest eigenvalue:

$$\lambda_{l^*} = \arg\max_{\lambda_l} N_l^{-1} M_l \tag{4}$$

Choosing the eigenvector that corresponds to the maximum eigenvalue can maximize the $J_l(a_l)$ to achieve the optimum solution. Once we determine the eigenvectors (i.e.,

the combination coefficients of all four different kernels), we now proceed to construct q_l and Q_l to find the corresponding Gram matrices of kernel K_l.

The optimization of the data-dependent kernel k_l consists in selecting the optimal combination coefficient vector a_l so that the class separability of the training data in the mapped feature space is maximized. Once we have computed these Gram matrices, we can find the optimum kernels for the given database. To do this, we arrange the eigenvalues (that determined the combination coefficients) for all kernel functions in descending order. The first kernel corresponding to the largest eigenvalue is used in the construction of a composite kernel that is expected to yield the optimum classification accuracy [Motai and Yoshida 2013]. If we apply one of the four kernels for the entire database, we cannot achieve the desired classification performance.

3.2. Decomposition of Database through Data Association via Recursively Updating Kernel Matrices

As shown in Step 2 of Figure 2, in each database, we apply the data association using class separability as a measure to identify whether the data are either heterogeneous or homogeneous. Obtaining a clear distinction between the heterogeneous and homogeneous data of each database is a very challenging task. The data acquired in clinical trials can sometimes be highly diverse, and there is no concrete method to differentiate between heterogeneous and homogeneous data. We would like to decompose each database so that the decomposed database has homogeneous data characteristics. If the data are homogeneous, then separability is improved; conversely, heterogeneous data degrade the class separability ratio. Let us introduce a variable ξ, which is the ratio of class separabilities:

$$\xi = \arg\max_{r}(J'_*(a'_r)/ J_*(a'_r)), \tag{5}$$

where $J'_*(a'_l) = a_l'^T M'_l a'_l / a_l'^T N'_l a'_l$ denotes the class separability yielded by the most dominant kernel, which is chosen from the four different kernels for the dataset, and $J'_*(a'_l) = a_l'^T M_l a'_l / a_l'^T N_l a'_l$ is the class separability yielded by the most dominant kernel for the entire database. Because of the properties of class separability, Equation (5) can be rewritten as $\xi = \lambda'_*/\lambda_*$, where λ'_* corresponds to the most dominant eigenvalue of in the maximization of Equation (4) under the kernel choice, given all clusters noted as r clusters. The newly computed λ'_* is the latest eigenvalue of another r set of "re-calculated" clusters. If ξ is less than a threshold value 1, then the database is heterogeneous; otherwise, it is homogeneous. If the data are homogeneous, we keep the Gram matrix, as defined in Section 3.1. Conversely, if the data are heterogeneous, we update the Gram matrix depending on the level of sub-dataset heterogeneity, as described later.

We propose to quantify the data's heterogeneity by introducing a criterion called the Residue Factor by extending Equation (5) $\xi = \lambda'_*/\lambda_*$ into the Residue Factor r f, defined as:

$$\mathrm{r\,f} = (a'_* - \bar{a}_*) \cdot \lambda'_*/\lambda_*, \tag{6}$$

where we use only the most dominant kernel for determining the Residue Factor. The class separability of the most dominant kernel for the newly decomposed data is directly dependent on both the maximum combination coefficient a'_* (this is the maximum combination coefficient of four different kernels), as well as the maximum eigenvalue λ'_*. Let us denote by $\bar{a}_*$ the mean of the combination coefficients of all databases and by a'_* the most dominant kernel among the subsets of the newly decomposed database, respectively. Using these values, we determine the type of update for the Gram matrix by evaluating disparities between composite eigenvalues, iteratively.

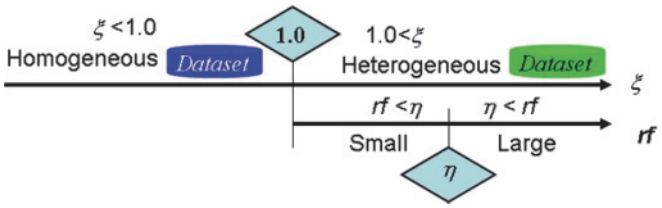

Fig. 3. Relationship of the criteria to determine homogenous and heterogeneous degree.

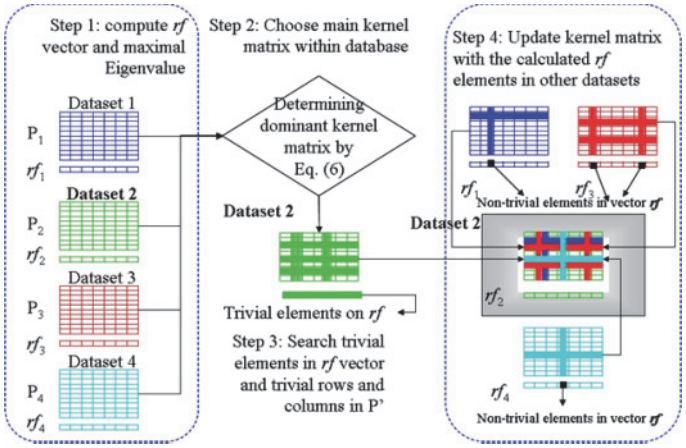

Fig. 4. The steps to choose the basic kernel matrix P' and updating process according to the elements rf.

We observed that there exists a chance of misinterpreting heterogeneous data while updating the newly clustered databases. Specifically, as shown in Figure 3, we consider the two cases of the updates for small/large heterogeneous data as follows:

Case 1: Partially update for small heterogeneous data

If the Residue Factor rf in Equation (6) is less than a threshold value η, then that means the heterogeneous degree between the previous eigenvectors and the new eigenvectors is relatively small. Hence, the dimensions of the Gram matrix have to remain constant. We replace the trivial rows and columns of the dominant kernel Gram matrix with those of the newly decomposed data. The trivial rows/columns are calculated by the minimum difference vector. Since we assigned the decomposed data to one of the existing datasets, we just compare the combination coefficient values of that class with the combination coefficient of new decomposed data to yield the difference vectors that determine the trivial combination vector to be replaced. This process is repeated for all the kernel matrices.

The input matrices P_l' and Q_l' should also be updated by removing rows and columns, by applying the four steps shown in Figure 4.

In Step 1, we compute the individual residue factor rf corresponding to each dataset and decompose one matrix corresponding to one database into kernel matrices for several datasets. In Step 2, we choose the main kernel matrix among the datasets by maximizing ξ. In Step 3, we search trivial elements in the rf vector in the main kernel matrix according Equation (6) that minimizes rf. In Step 4, we substitute the corresponding parts in the main kernel matrix with the calculated element of rf in the other datasets. We compute $Q_l' = \mathrm{diag}(a_l')$. Hence, in the update for small heterogeneous data, the Gram matrix can be given as $K_l' = Q_l' P_l' Q_l'^{\mathrm{T}}$.

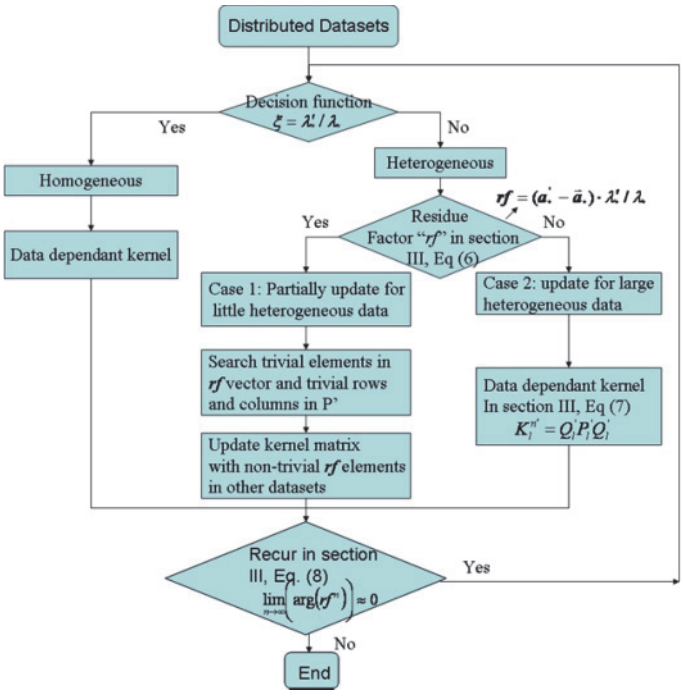

Fig. 5. The training flowchart of reclustered databases.

Case 2: Update for large heterogeneous data

If the Residue Factor $\mathrm{r\,f}$ in Equation (6) is greater than a threshold value η, that means the heterogeneous degree between the previous eigenvectors and the new eigenvectors is relatively large. So, it is very important for us to retain this highly heterogeneous data for efficient classification. Instead of replacing the trivial rows and columns of the previous data, we simply retain them. Hence, the size of the Gram matrix is increased by the size of the newly decomposed data. We already calculated the new combination coefficient a_{l}', and input matrix P' and Q' are the same as in Section 3.1. Then, the kernel Gram matrix can be newly calculated as:

$$\mathrm{K}_{\mathrm{l}}^{\mathrm{n}'} = \mathrm{Q}_{\mathrm{l}}'\,\mathrm{P}_{\mathrm{l}}'\,\mathrm{Q}_{\mathrm{l}}'^{\mathrm{T}}. \tag{7}$$

Once we have our kernel Gram matrices, we can determine the composite kernel that gives us the optimum classification accuracy when the existing database is incorporated with newly decomposed data.

We perform the entire algorithm to see if there is an improvement in the difference of $\vec{\mathrm{a}}_*$ and a_*' between the current and previous steps. If the heterogeneous degree is still large, then the decomposed data has to be further reduced, and the recursive algorithm described herein is performed again. This entire process is summarized in the flowchart in Figure 5. This process is repeated until the Residue Factor finds an appropriate size of data that would allow for all the decomposed datasets to be homogeneous. That means the Residue Factor is expected to converge to zero by recursively updating clusters:

$$\lim_{\mathrm{n}\to\infty} \vec{\arg}(\mathrm{r\,f}^{\mathrm{n}}) \approx 0 \tag{8}$$

After the training of the Gram matrix is finished to incorporate the heterogeneous/homogeneous features of the newly decomposed data, the KFA algorithm is applied to the kernel Gram matrix. Obtaining a higher dimensional feature space for the huge volume of data with greater classification power depends on how effectively we update the Gram matrix. As the size of distributed databases increases, it is often very important to re-evaluate and change the criteria established using an enhanced algorithm to correctly train the big data. In the following section, we explore how the re-clustered multiple databases must be aligned.

4. GROUP KERNEL FEATURE ANALYSIS (GKFA)

There is currently insufficent true cancer data in any single database to validate the accuracy of CT colonography. Due to the limitations of training cases, few radiologists have clinical access to the full variety of true-positive cases encountered in actual clinic practices of CT colonography. Therefore, we propose combining the data from several databases to improve classification by increasing available colon cancer cases and the diversity of patients. There are as yet few data centers that render data into a form that can be readily reused, shoulder curatorial responsibilities, or build new data management tools and services.

In this section, as shown in Step 3 of Figure 2, we illustrate how to handle multiple databases through GKFA.

4.1. Composite Kernel: Kernel Combinatory Optimization

In this section, we propose a composite kernel function to define the weighted sum of the set of different optimized kernel functions, which corresponds to multiple clustered databases. To obtain the optimum classification accuracy, we define the composite kernel $K^s{}_{com}(\rho)$, using a composite coefficient ρ, as

$$K^s_{com}(\rho) = \rho_{l_1} Q_{l_1} P_{l_1} Q^T_{l_1} + \rho_{l_2} Q_{l_2} P_{l_2} Q^T_{l_2} \tag{9}$$

where $K^s_{com}(\rho)$ is a composite kernel obtained by combining two of the four basic kernel functions (linear, polynomial, Gaussian RBF, and Laplace RBF). Thus, the number of possible combinations is six: $\binom{4}{2} = 4!/(2! * 2!) = 6$ cases, where s represents one of six composite kernels. Through this approach, the relative contribution of a single kernel to the composite kernel can be varied over the multiple databases by the value of the composite coefficient ρ. Instead of using K_l as the kernel matrix, we use $K^s_{com}(\rho)$. According to Chang et al. [2008], this composite kernel matrix $K^s_{com}(\rho)$ satisfies Mercer's condition. Each term should be positive or zero; thus, the combined two terms in Equation (9) are guaranteed for the semi-definiteness of the composite kernel.

The problem becomes how to determine this composite coefficient such that the classification performance is optimized. To this end, we used the concept of kernel alignment to determine the best $\hat{\rho} = [\rho_{l1}, \rho_{l2}]$, which gives optimum performace. The alignment measure was proposed by Cristianini and Kandola [Cristianini et al. 2001] to compute the adaptability of a kernel to the target data and provide a practical method to optimize the kernel. It is defined as the normalized Frobenius inner product between the kernel matrix and the target label matrix. The empirical alignment between kernel k_1 and kernel k_2 with respect to the training set is given as:

$$A(k_1, k_2) = \langle K_1, K_2 \rangle_F \,/\, \|K_1\|_F \, \|K_2\|_F \,, \tag{10}$$

where K_1 and K_2 are the kernel matrix for the training set using kernel function k_1 and k_2, $\|K_1\|_F = \sqrt{\langle K_1, K_1 \rangle_F}$, $\|K_2\|_F = \sqrt{\langle K_2, K_2 \rangle_F}$. $\langle K_1, K_2 \rangle_F$ is the Frobenius inner product between K_1 and K_2. If $K_2 = yy^T$, then the empirical alignment between kernel K^s_{com}

and target vector y is:

$$A\,K^s_{com}, yy^T = K^s_{com}, yy^T{}_F \quad K^s_{com}, K^s_{com}{}_F \quad yy^T, yy^T{}_F = y^T K^s_{com} y \quad n \quad K^s_{com}, K^s_{com}{}_F \tag{11}$$

If the kernel is well-adapted to the target information, separation of the data has a low bound on the generalization error [Cristianini et al. 2001]. So, we can optimize the kernel alignment by training data to improve the generalization performance on the testing data. Let us consider the optimal composite kernel corresponding to Equation (9) as: $K^s_{com}(\hat{\rho}) = \hat{\rho}_{l_1} Q_{l_1} P_{l_1} Q^T_{l_1} + \hat{\rho}_{l_2} Q_{l_2} P_{l_2} Q^T_{l_2}$. We can change ρ to maximize the empirical alignment between $K^s_{com}(\hat{\rho})$ and the target vector yy^T. Hence,

$$\hat{\rho} = \arg_\rho \max A\,K^s_{com}, yy^T$$
$$= \arg\max_\rho \left(\sum_{l=1} \rho_l K_l, yy^T \quad n \sum_{l_1=1} \rho_{l_1} K_{l_1} \quad , \quad \sum_{l_2=1} \rho_{l_2} K_{l_2} \right) \tag{12}$$

$$= \arg\max_\rho \left(\sum_{l=1} \rho_l u_l \quad n^2 \sum_{l_1=1, l_2=1} \rho_{l_1} \rho_{l_2} v_{l_1 l_2} \right) = \arg\max_\rho \rho^T U \rho \; n^2 \rho^T V \rho, \tag{13}$$

where $u_l = \langle K_l, yy^T \rangle$, $v_{l_1 l_2} = \langle K_{l_1}, K_{l_2} \rangle$, $U_{l_1 l_2} = u_{l_1} u_{l_2} V_{l_1 l_2} = v_{l_1} v_{l_2}$, $\rho = (\sqrt{\rho_{l_1}}, \sqrt{\rho_{l_2}})$. Let the generalized Raleigh coefficient be $J(\rho) = \rho^T U \rho \; \rho^T V \rho$. Therefore, we can obtain $\hat{\rho}$ by solving the generalized eigenvalue problem:

$$U\rho = \delta V \rho, \tag{14}$$

where δ denotes the eigenvalues of kernel alignment. Once we find this optimum composite coefficient $\hat{\rho}$, which will be the eigenvector corresponding to the maximum eigenvalue $\bar{\delta}$, we can compute the composite data-dependent kernel matrix $K^s_{com}(\rho)$ according to Equation (9) by changing data clusters. That means that eigenvectors $\hat{\rho}$ for $U\rho = \delta V \rho$ provide the optimum coefficients for the composite kernel in Equation (9).

This composite kernel process provides an optimal data-dependent kernel. We can now proceed with to train the multiple databases for the re-clustered database, as described in the subsequent section.

4.2. Multiple Databases Using Composite Kernel

We extend Section 4.1 to multiple databases after the composite kernels have been identified. Four basic kernels are considered to combine and represent the six $K_{com}(\rho)$ shown in Figure 6: $K^1_{com}(\rho)$, $K^2_{com}(\rho)$, $K^3_{com}(\rho)$, and $K^5_{com}(\rho)$. We assign each database (there are 19 databases in our colonography experiment described in Section 5) one of six composite kernel cases. Database 1, for example, is labeled by kernel K^1_{com}, Databases 8 and 9 use kernel K^2_{com}. Then we assemble the database according to the kernel label.

To assemble the databases, we further optimize combining coefficients by assembling databases. Our goal is to find a composite kernel that will best fit these newly assembled databases K^s_{group}. Since the composite kernel with the coefficients was calculated in Section 4.1 starting from Equation (9), the desired calculation of the assembled databases utilizes the precalculated values K^s_{group}. Let us define the newly calculated group kernel K^s_{group} by the weighted sum of the composite kernel previously calculated

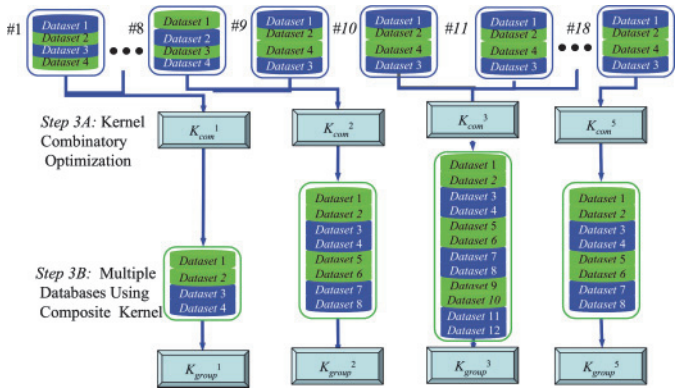

Fig. 6. Group kernel feature analysis (GKFA). Steps 1 and 2 are the same as in Figure 2. Step 3 of Figure 2 is illustrated here for assembled databases through kernel choice in the composite kernel as in Section 4.1 (Kernel Combinatory Optimization) and Section 4.2 (Multiple Databases Using Composite Kernel).

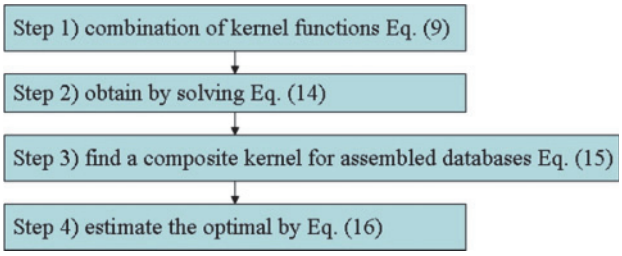

Fig. 7. The overall GKFA steps for the newly assembled databases.

in Equation (14) in Section 4.1:

$$K^{s}_{group}(\hat{\rho}_{group}) = \sum_{g=1}^{D_g} \hat{\rho}_g (K^{s}_{com})_g, \tag{15}$$

where K^{s}_{group} is the weighted sum of data-dependent kernel optimized by assembling databases up to the number D_g, representing K^{s}_{group}. Dg is defined as the total database number assembled for grouping under the identical composite kernel K^{s}_{group}. We do not directly calculate K^{s}_{group} using Equation (15) since we already have calculated $\hat{\rho}$ and d corresponding to K^{s}_{group} for the individual databases in Section 4.1. We would like to estimate the optimal $\hat{\rho}_g$ shown in Equation (15) by the weighted sum of eigenvectors through the eigenvalues δ previously calculated in Equation (14):

$$\hat{\rho}_{group} = \sum_{g=1}^{D_{rcs}} \delta_g \hat{\rho}_g \Big/ \sum_{g=1}^{D_{rcs}} \delta_g, \tag{16}$$

where the value d_g denotes the eigenvalues of the database g in Equation (14), corresponding to the individual database in Section 4.1. The newly grouped kernel, corresponding to the largest eigenvalue d_g, is used in the construction of an assembled database to yield the optimum classification accuracy. This entire process (detailed in Sections 4.1 and 4.2) is summarized in the flowchart in Figure 7.

Table I. Databases

	# Patients		Total #	# Database		Total #
Databases	# TP patients	# FP patients	Patients	# TP	# FP	Database
1	5	30	35	12	155	167
2	3	29	32	5	217	222
3	3	10	13	7	213	220
4	3	27	30	5	206	211
5	7	35	42	12	196	208
6	3	25	28	6	198	204
7	3	24	27	6	208	214
8	1	28	29	4	200	204
9	3	17	20	8	190	198
10	3	22	25	7	198	205
11	4	23	27	8	181	189
12	3	29	32	4	191	195
13	2	11	13	8	208	216
14	3	27	30	5	188	193
15	3	15	18	7	147	154
16	3	5	8	8	221	229
17	3	12	15	7	169	176
18	2	25	27	12	169	181
19	2	11	13	5	183	188
Average	3.1	21.3	24.4	7.15	190.1	197.3

Rather than a single database, the proposed GKFA approach uses more than one database to improve cancer classification performance, as shown in the experimental results.

5. EXPERIMENTAL RESULTS

5.1. Cancer Databases

We evaluated the performance of the proposed GKFA based on a retrospectively established database of clinical cases obtained from several multicenter screening CT Colonography (CTC) trials [Pickhardt et al. 2003; Rockey et al. 2005; Regge et al. 2009]. The database consisted of 464 CTC cases that were obtained from a total of 19 medical centers in the United State and Europe. The current study was a post-analysis of earlier clinical trial data conducted in compliance with Health Insurance Portability and Accountability Act regulations and was approved by our institutional review board.

Our previously developed CAD scheme [Yoshida and Nappi 2001; Näppi and Yoshida 2007; Yoshida et al. 2002] was applied to the CTC cases, which yielded a total of 3,774 detections (polyp candidates) consisting of 136 True-Positive (TP) detections and 3,638 False-Positive (FP) detections. The supine and prone CTC volumes of a patient were treated as independent in the detection process. A Volume of Interest (VOI) of 963 pixels was placed at each candidate to cover the entire region of the candidate. The collection of the VOIs for all the candidates consisted of the databases used for the performance evaluation as shown in Table I. We applied up to 40%-fold cross-validation for testing with the training data.

The proposed statistical analysis by use of GKFA was applied to the databases in Table I, which showed that the CTC data were highly biased toward FPs (the average ratio between TP and FP is 1: 26.6) due to the limited number of TPs caused by an asymptomatic patient cohort. The proposed statistical analysis using GKFA is expected to compensate for the lack of TPs by incorporating the multiple databases.

Table II. Eigenvalues of Four Kernels for Offline Databases

Databases	The First Kernel	The Second Kernel
1	Sigmoid Kernel	Gauss Kernel
	$\lambda = 107$, d $= 3.334{*}10^{-4}$, Offset $= 0$	$\lambda = 3.93$, $\sigma = 0.7$
2	Sigmoid Kernel	Polynomial Kernel
	$\lambda = 31.6$, d $= 2.223{*}10^{-4}$, Offset $= 0$	$\lambda = 2.41$, d $= 5$, Offset $= 0.1$
3	Sigmoid Kernel	Polynomial Kernel
	$\lambda = 96.7$, d $= 1.0{*}10^{-7}$, Offset $= 0$	$\lambda = 5.59$, d $= 1$, Offset $= 0.1$
4	Sigmoid Kernel	Linear Kernel
	$\lambda = 105$, d $= 1.112{*}10^{-4}$, Offset $= 0$	$\lambda = 9.21$, d $= 3$
5	Sigmoid Kernel	Gauss Kernel
	$\lambda = 328$, d $= 3.334{*}10^{-4}$, Offset $= 0$	$\lambda = 4.44$, $\sigma = 0.1$
6	Sigmoid Kernel	Gauss Kernel
	$\lambda = 80.1$, d $= 2.223{*}10^{-4}$, Offset $= 0$	$\lambda = 9.82$, $\sigma = 0.8$
7	Sigmoid Kernel	Linear Kernel
	$\lambda = 38.3$, d $= 2.223{*}10^{-4}$, Offset $= 0$	$\lambda = 17.1$, d $= 5$
8	Sigmoid Kernel	Gauss Kernel
	$\lambda = 127$, d $= 5.556{*}10^{-4}$, Offset $= 0$	$\lambda = 8.31$, $\sigma = 0.1$
9	Sigmoid Kernel	Gauss Kernel
	$\lambda = 35.9$, d $= 1.112{*}10^{-4}$, Offset $= 0$	$\lambda = 20.5$, $\sigma = 0.9$
10	Sigmoid Kernel	Linear Kernel
	$\lambda = 18.6$, d $= 1.112{*}10^{-4}$, Offset $= 0$	$\lambda = 2.28$, d $= 3$
11	Sigmoid Kernel	Gauss Kernel
	$\lambda = 52$, d $= 2.223{*}10^{-4}$, Offset $= 0$	$\lambda = 2.53$, $\sigma = 0.1$
12	Sigmoid Kernel	Gauss Kernel
	$\lambda = 88.9$, d $= 4.445{*}10^{-4}$, Offset $= 0$	$\lambda = 14.8$, $\sigma = 0.6$
13	Sigmoid Kernel	Gauss Kernel
	$\lambda = 40.8$, d $= 2.223{*}10^{-4}$, Offset $= 0$	$\lambda = 1.78$, $\sigma = 0.1$
14	Polynomial Kernel	Sigmoid Kernel
	$\lambda = 29.6$, d $= 1$, Offset $= 0.1$	$\lambda = 6.3$, d $= 0.000001$ offset $= 0$
15	Sigmoid Kernel	Gauss Kernel
	$\lambda = 280$, d $= 3.334{*}10^{-4}$, Offset $= 0$	$\lambda = 11.0$, $\sigma = 0.7$
16	Sigmoid Kernel	Gauss Kernel
	$\lambda = 48.1$, d $= 3.334{*}10^{-4}$, Offset $= 0$	$\lambda = 1.82$, $\sigma = 0.1$
17	Sigmoid Kernel	Polynomial Kernel
	$\lambda = 89.0$, d $= 3.334{*}10^{-4}$, Offset $= 0$	$\lambda = 1.28$, d $= 7$, Offset $= 0.1$
18	Sigmoid Kernel	Gauss Kernel
	$\lambda = 179$, d $= 2.223{*}10^{-4}$, Offset $= 0$	$\lambda = 1.38$, $\sigma = 0.1$
19	Sigmoid Kernel	Gauss Kernel
	$\lambda = 46.2$, d $= 3.334{*}10^{-4}$, Offset $= 0$	$\lambda = 1.35$, $\sigma = 0.1$

5.2. Optimal Selection of Data-Dependent Kernels

We used the method proposed in Section 3.1 to create four different data-dependent kernels and select the kernel that best fit the data and achieved optimum classification accuracy for each database. We determined the optimum kernel depending on the eigenvalue that yielded maximum separability. The performance measure used to evaluate the experimental results was defined as the ratio between the number of successfully classified polyps and the total number of polyps. Table II lists the eigenvalues λ and parameters of four kernels for each database calculated in Equation (5).

Table II shows the maximal eigenvalues corresponding to the data-dependent kernels of an individual database. Among the four data-dependent kernels, the Sigmoid kernel was observed to achieve the best performance for most databases except for Database 14. The kernel with the maximum eigenvalue is highlighted for each database in Table II.

Table III. The Value of $\hat{\rho}$ for Each of the Composite Kernels

Databases	Two Most Dominant Kernels	ρ_1	ρ_2	K-NN(k)	Performance (%)
1	Sigmoid and Gauss	0.73	0.27	1	98.00
2	Sigmoid and Poly	0.27	0.73	8	90.00
3	Sigmoid and Poly	0.68	0.32	7	86.27
4	Sigmoid and Linear	0.94	0.06	3	94.23
5	Sigmoid and Gauss	0.30	0.70	1	92.16
6	Sigmoid and Gauss	0.72	0.28	1	97.78
7	Sigmoid and Linear	0.27	0.73	3	90.20
8	Sigmoid and Gauss	0.31	0.69	1	96.00
9	Sigmoid and Gauss	0.65	0.35	1	92.16
10	Sigmoid and Linear	0.27	0.73	1	92.00
11	Sigmoid and Gauss	0.28	0.72	1	98.04
12	Sigmoid and Gauss	0.24	0.76	3	94.12
13	Sigmoid and Gauss	0.27	0.73	3	92.16
14	Sigmoid and Poly	0.43	0.57	5	90.38
15	Sigmoid and Gauss	0.73	0.27	1	95.35
16	Sigmoid and Gauss	0.27	0.73	1	97.96
17	Sigmoid and Poly	0.31	0.69	1	91.84
18	Sigmoid and Gauss	0.74	0.26	1	94.00
19	Sigmoid and Gauss	0.27	0.73	4	90.91

5.3. Kernel Combinatory Optimization

Once we find the kernel that yields the optimum eigenvalue, we select the two largest kernels to form the composite kernel. For example, for Database 1, we combined the Sigmoid and Gauss kernels to form the composite kernel. We observed that each database had different combinations for the composite kernels. We adopted the KFA algorithm to obtain the feature vectors, and we classified them using the K-Nearest Neighbor (K-NN) method with a metric of Euclidean distance. Table III shows how the two kernel functions are combined according to the composite coefficients listed in the table. These composite coefficients were obtained in Section 4.2. For all the databases, the most dominant kernels kept varying, and the second most dominant kernel was the Sigmoid kernel. As a result, the contribution of the Sigmoid kernel was lower when compared to other kernels in forming a composite kernel.

5.4. Composite Kernel for Multiple Databases

We used the method proposed in Section 4.2 to obtain the group kernel by the weighted sum of the composite kernels, then assembled 19 individual databases according to kernel type. As for the Sigmoid and Gauss group kernel, we sorted 12 databases in order as follows: $\lambda_*^9 < \lambda_*^{13} < \lambda_*^{11} < \lambda_*^{16} < \lambda_*^{19} < \lambda_*^6 < \lambda_*^{12} < \lambda_*^1 < \lambda_*^8 < \lambda_*^{18} < \lambda_*^{15} < \lambda_*^5$; then, they are divided into three assembled databases by maximal eigenvalue, as in Equation (4). For each assembled database, we applied the GKFA method with the K-NN classifier, and assembled database parameters.

Table IV shows the assembled databases by group kernel. Performance can be compared with the values in Table III. Table IV shows that most databases can be categorized into the Sigmoid and Gauss group kernel. We divided 12 databases into three assembled databases based on the order of eigenvalues. The classification performance for combined databases is 98.49% on average, compared to the performance of KFA with individual databases which is 95.22%. The second database assembled was the Sigmoid and Poly group kernel. A classification rate of 95.33% was achieved, which outperformed KFA with individual Databases 2, 3, 14, and 17 by an average of 89.62%. The last database assembled was the Sigmoid and Linear group. We obtained a 97.35%

Table IV. GKFA for Assembled Database

Kernel Type	Database Assembled	First Kernel	Second Kernel	Performance (%)
Sigmoid and Gauss	Database9 Database13 Database11 Database16	Sigmoid $d = 6.67*10^{-4}$ Offset = 0. $\rho_1 = 0.49$	Gauss $\sigma = 0.4$ $\rho_2 = 0.51$	97.54
	Database19 Database6 Database12 Database1	Sigmoid $d = 8.89*10^{-4}$ Offset = 0. $\rho_1 = 1.00$	Gauss $\sigma = 0.2$ $\rho_2 = 0.12$	100.00
	Database8 Database18 Database15 Database5	Sigmoid $d = 7.78*10^{-4}$ Offset = 0. $\rho_1 = 0.95$	Gauss $\sigma = 0.3$ $\rho_2 = 0.05$	97.94
Sigmoid and Poly	Database2 Database3 Database14 Database17	Sigmoid $d = 3.34*10^{-4}$ Offset = 0. $\rho_1 = 0.71$	Linear d = 3, Offset = 0.1 $\rho_2 = 0.20$	95.33
Sigmoid and Linear	Database4 Database7 Database10	Sigmoid $d = 3.34*10^{-4}$ Offset = 0. $\rho_1 = 1.00$	Linear d = 13 $\rho_2 = 0.12$	97.35

performance rate, which was 5.2% higher than KFA for Databases 4, 7, and 10. Therefore, we can conclude from the comparison of Tables III and IV that classification performance can be improved by using GKFA for the assembled databases over KFA for a single database.

5.5. K-NN Classification Evaluation with ROC

We demonstrate the advantage of GKFA in terms of the Receiver Operating Characteristic (ROC) by comparing GKFA for assembled databases to KFA for a single database. We evaluated the classification accuracy shown in Table III and ROC using the sensitivity and specificity criteria as statistical measures of performance. The True Positive Rate (TPR) defines diagnostic test performance for classifying positive cases correctly among all positive instances available during the test. The False Positive Rate (FPR) determines how many incorrect positive results occur among all negative samples available during the test [Fawcett 2006]:

$$\text{TPR} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}} \tag{17}$$

$$\text{FPR} = \frac{\text{False Positives (FP)}}{\text{False Positives (FP)} + \text{True Negatives (TN)}} \tag{18}$$

The classification performance was evaluated by the Area Under the Curve (AUC) by calculating the integral of the ROC plot in the range between 0 and 1. We used the K-NN method as classifier with parameter k, according to the performance of TPR and specificity with respect to the variable k in Figures 8, 9, and 10 and corresponding to the kernel types of Table IV.

Figure 8 shows the ROC results for the Sigmoid and Gauss group kernels. We compared the performance of assembled databases by AUC, with a single database (Databases 9, 11, 13, and 16) in Figure 8(a), a single database (Databases 1, 6, 12, and 19) in Figure 8(b), and a single database (Databases 5, 8, 15, and 18) in Figure 8(c), respectively. In Figure 8(a), GKFA for the assembled database outperformed the KFA

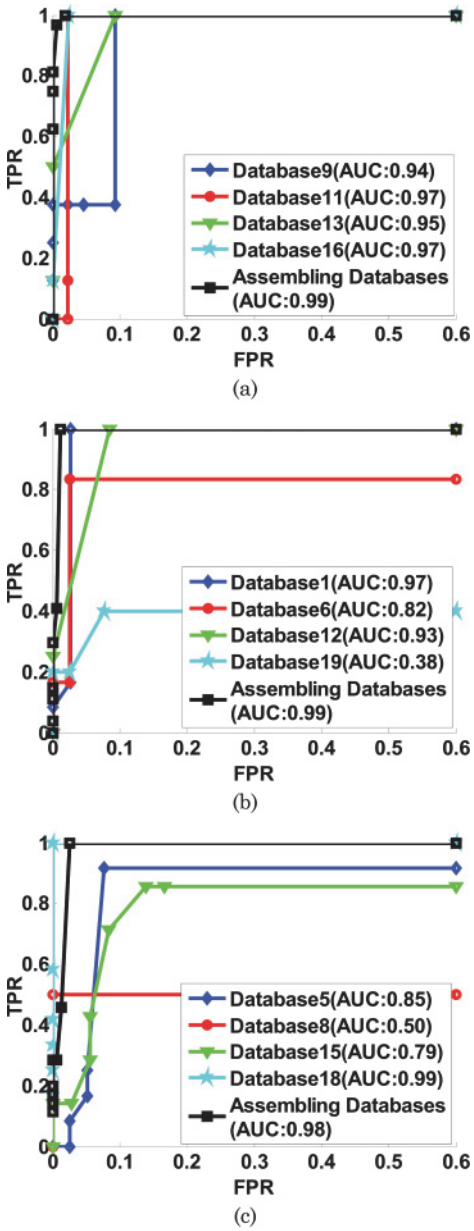

Fig. 8. ROC with Sigmoid and Gauss Group kernel (a) comparison between assembled databases and databases 9, 11, 13, and 16; (b) comparison between assembled databases and databases 1, 6, 12, and 19; (c) comparison between assembled databases and databases 5, 8, 15, and 18.

for a single database by achieving a higher TPR and a lower FPR. In Figure 8(b), although Database 1 had a smaller FPR, GKFA for the assembled database had a better gradient. In Figure 8(c), the performance of GKFA for the assembled database was not as good as KFA for Database 18, but it performed better than the other three databases in terms of the ROC.

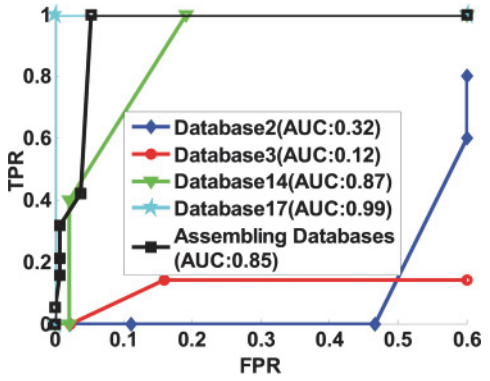

Fig. 9. ROC with Sigmoid and Poly Group kernel.

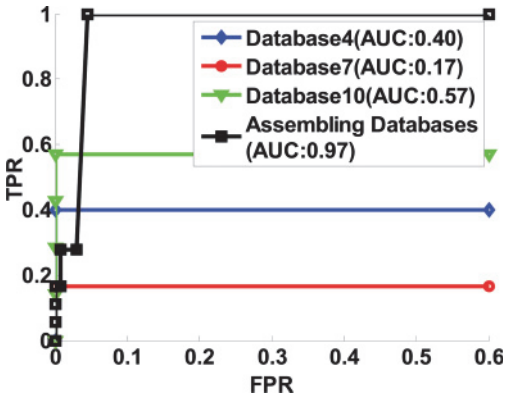

Fig. 10. ROC with Sigmoid and Linear Group kernel.

Figure 9 shows that KFA with Database 17 had the best performance in ROC, and the second best one was the performance of GKFA for the assembled database. Regarding KFA for Databases 2, 3, and 14, FPR was a bit high, and, at the same time, TPR was not good enough. In the Sigmoid and Poly group kernel experiment, GKFA for the assembled databases was not the best one, but better than KFA for the rest of the single databases.

In Figure 10, we see that GKFA for the assembled database outperformed KFA for Databases 4, 7, and 10, although Database 10 had a small FPR, but the maximal TPR only reached 0.58. On the other hand, GKFA for the assembled databases reached 0.98, although the FPR was bigger than for Database 10. In most cases, GKFA for the assembled databases had the advantage over KFA by an average of 22.2% in TPR and 3.8% in FPR for a single database with respect to FPR and TPR from Figures 8, 9, and 10.

5.6. Comparison of Results with Other Studies on Colonography

In this section, we compared results for the Sigmoid and Gauss group kernels for four databases (Databases 1, 6, 12, and 19) and the assembled databases with different classifiers, which include RBF Neural Networks (RNN), Back-propagation Neural Networks (BNN), Support Vector Machines (SVM), Decision Trees (DT), and the K-NN method [Stork and Yom-Tov 2004]. Classification performance is shown in Table V through a Matlab implementation, and the resulting ROC curves are shown in

Table V. Comparison of Different Classification Methods

Classifier Method	Classifier Parameters	Performance (%)
RNN	Number of iterations given 2,500	90.00
BNN	Number of iterations given 1,500	82.11
SVM	Gauss kernel, Gaussian width 0.001, slack 0.1	87.37
DT	Incorrectly assigned samples at a node 10%	96.32
K-NN	**K = 1**	**97.94**

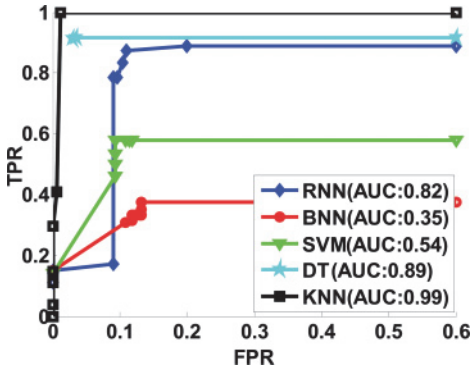

Fig. 11. Classification comparison with different methods.

Table VI. Classification Accuracy Compared to Other Methods

Studies	Performance (%)	Datasets	Methods
Yoshida et al. [2001]	95.0	Colonic Polyps	Shape Index
Yao et al. [2004]	91.8	Colonic Polyps	Fuzzy Clustering and Deformable Models
Nappi and Yoshida [2007]	95.0	Colonic Polyps	Adaptive Density Correction and Mapping
van Ravesteijn et al. [2010]	95.0	Colonic Polyps	Logistic Regression
Awad et al. [2010]	93.4	Colonic Polyps	Weighted Proximal Support Vector Machines
Cai et al. [2011]	94.6	Phantom and Colonic Polyps	Mosaic Decomposition
Multiple CTC	97.6	**Colonic Polyps**	**Group Kernel Feature Analysis(GKFA)**

Figure 11. In Table V, we see that the K-NN method yielded the best performance, followed by DT, RNN, SVM, and BNN methods in descending order. Each classifier is listed with its corresponding parameters. The construction of NN is a three-layer network with a fixed-increment single-sample perceptron for BNN, and RNN has a radial basis function in the middle layer.

In Figure 11, we also see that the assembled databases with the K-NN method achieved the best performance, followed by DT with less TPR. Thus, we conclude that the K-NN method was the most appropriate classifier for the distributed medical imaging databases with GKFA in the experiment.

Other studies that investigated the classification of CTC showed comparable performance [Yoshida and Nappi 2001; Yao et al. 2004; Nappi and Yoshida 2007; van Ravesteijn et al. 2010; Awad et al. 2010; Cai et al. 2011], as shown in Table VI, along with the classification performance based on the proposed GKFA method applied to multiple CTC databases.

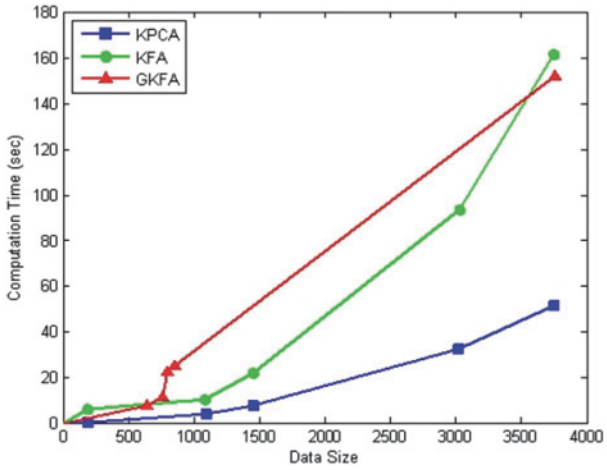

Fig. 12. The computation time comparison among KPCA, KFA, and GKFA as data size increases.

Note that, in this comparison table, the CTC datasets were different from each other; thus, a direct comparison of classification performance is not possible. However, the fact that the proposed GKFA classification yielded an overall performance of 97.6% on average with a standard deviation of 2.77% implies that the proposed classifier may potentially outperform the previous classifiers.

This is the first study that demonstrates that multiple databases may improve classification performance. Future works propose to increase the size of databases and the number of institutions and to extend the interface module in the CAD system.

5.7. Computational Speed and Scalability Evaluation of GKFA

The computational efficiency of the proposed GKFA method was evaluated by comparing its runtime with KPCA and KFA for the selected datasets. The algorithms were implemented in Matlab R2007b using the Statistical Pattern Recognition Toolbox for the Gram matrix calculation and kernel projection. The processor was a 3.2GHz Intel® Pentium 4 CPU with 3GB of RAM. Runtime was determined using the cputime command. For each algorithm, computation time increases with increasing training data size (n), as expected. All three methods required the computation of a Gram matrix whose size increases as the data size increased. The results from Figure 12 clearly indicated that GKFA and KFA required more computational time than KPCA because the composite data-dependent kernels needed further calculations of a Gram matrix and optimization of coefficient parameters. These overhead computations were the main causes of the increase in runtime.

Typically, overall scalability is estimated by computational complexity. If the KPCA algorithm contains an eigenvalue problem of rank n, the computational complexity of KPCA is $O(n^3)$. In addition, each resulting eigenvector is represented as a linear combination of n terms; the l features depend on n image vectors of X_n. Thus, all data contained in X_n must be retained, which is computationally cumbersome and unacceptable for our distributed applications. If the KFA algorithm contains an eigenvalue problem of rank n, the computational complexity of KFA is expected to be $O(n^2)$. If the GKFA algorithm contains a distributed problem of D_g databases, the computational complexity of GKFA is expected to be $O(D_g n_g{}^2)$.

6. CONCLUSION

In this article, we proposed a new framework for health informatics: computer-aided detection of colonic polyps using distributed colonography, where distributed databases from multiple institutions are considered for participation. We showed how to merge the information-centric characteristics and node-centric physical world connectivity to develop a smart healthcare system. This is the first pilot study on the evaluation of how the proposed machine-supported diagnosis system can handle multiple colonography databases.

The size of TPs is usually small at a single institution due to the screening nature of the colonoscopies. The CAD algorithm and clinicians both can observe all the potential cases in the proposed distributed platform. For handling multiple CTC databases, GKFA was developed in this article. When GKFA was used with assembled databases, it achieved an average improvement of 22.2% in TPR and 3.8% in FPR compared to single databases. GKFA has the potential to be a core classifier in the distributed computing framework for a CAD scheme, which will yield high detection performance of polyps using multiple distributed databases. Successful development of CAD in the distributed computing environment may advance the clinical implementation of cancer screening and promote the early diagnosis of colon cancer. Such a CAD scheme can make CTC a viable option for screening large patient populations, resulting in early detection of colon cancer and leading to reduced mortality due to colon cancer.

APPENDIX

Acronym Definitions

CAD	Computer-Aided Detection
NN	Neural Network
GKFA	Group Kernel Feature Analysis
ROC	Receiver Operating Characteristics
CT	Computed Tomography
KPCA	Kernel Principal Component Analysis
RBF	Radial Basis Function
KFA	Kernel Feature Analysis
FP	False Positive
TP	True Positive
PCA	Principal Component Analysis
VOI	Volumes Of Interest
TPR	True Positive Rate
FPR	False Positive Rate
K-NN	K-Nearest Neighbor
SVM	Support Vector Machine

Symbol Definitions

x_i	Input Data
y_i	Output Class Label
$k_{ij} = \langle \Phi(x_i), \Phi(x_j) \rangle$	Element of Gram Matrix
K	Kernel Gram Matrix
$q_l(.)$	Factor Function
L	Base Kernel Label

$p_s = (\cdot, \cdot)$	Base Kernel
α_{lm}	Combination Coefficient
K_l	Data-Dependent Kernel
Q_l	Factor Function Matrix
Q_l'	Updated Ql Matrix
$P_{\cdot\cdot}^{l}$	Base Kernel Matrix
$P_{\cdot\cdot}^{l\prime}$	Updated Pl.. Matrix
J	Fisher Scalar
S_{br} / S_{wr}	Between-class / Within-class Scatter Matrices
λ_l	Eigenvalue of Fisher Scalar
λ_*	Largest Eigenvalue
ξ	Ratio of the Class Separability
r f	Residue Factor
n_r	Number of Data for the r-th Cluster
$K_{com}^{s}(\rho)$	Composite Kernel
ρ	Composite Coefficient
s	One of Six Composite Kernels
$A(k_1, k_2)$	Empirical Alignment between Kernels k1 and k2
δ	Eigenvalues of Kernel Alignment
$\hat{\rho}$	Optimum Composite Coefficient
K_g^s	Group Kernel
δ_d	Eigenvalues of the Database d
$J_*'(a_l')$	Class Separability Yielded by the Most Dominant Kernel for Dataset(subsets) of Database
$J_*(a_l')$	Class Separability Yielded by the Most Dominant Kernel for the Entire Database
a_*'	Combination Coefficients of the Most Dominant Kernel among the Subsets
$\bar{a}_*$	Mean of Combination Coefficients of All Databases

REFERENCES

American Cancer Society. 2014. Cancer Facts & Figures 2014. American Cancer Society.

M. Awad, Y. Motai, J. Näppi, and H. Yoshida. 2010. A clinical decision support framework for incremental polyps classification in virtual colonoscopy. Algorithms 3, 1–20.

C. L. Cai, J. G. Lee, M. E. Zalis, and H. Yoshida. 2011. Mosaic decomposition: An electronic cleansing method for inhomogeneously tagged regions in noncathartic CT colonography. IEEE Transactions on Medical Imaging 30 (2011), 559–570.

T. T. Chang, J. Feng, H. W. Liu, and H. Ip. 2008. Clustered microcalcification detection based on a multiple kernel support vector machine with grouped features. In Proceedings of the 19th International Conference on Pattern Recognition. 8, 1–4.

X. W. Chen. 2003. Gene selection for cancer classification using bootstrapped genetic algorithms and support vector machines. In Proceedings of the IEEE International Computational Systems, Bioinformatics Conference. 504–505.

T. J. Chin and D. Suter. 2007. Incremental kernel principal component analysis. IEEE Transactions on Image Processing 16, 6, 1662–1674.

N. Cristianini, J. Kandola, A. Elisseeff, and Shawe-Taylor, J. 2001. On kernel target alignment. In Proceedings of the Neural Information Processing Systems. 367–373.

T. Fawcett. 2006. An introduction to ROC analysis. Pattern Recognition Letters 27, 8, 861–874.

H. Fröhlich, O. Chapelle, and B. Scholkopf. 2003. Feature selection for support vector machines by means of genetic algorithm. In Proceedings of the 15th IEEE International Conference on Tools with Artificial Intelligence. 142–148.

L. Hoegaerts, L. De Lathauwer, I. Goethals, J. A. K. Suykens, J. Vandewalle, and B. De Moor. 2007. Efficiently updating and tracking the dominant kernel principal components. Neural Networks 20, 2, 220–229.

C. Hu, Y. Chang, R. Feris, and M. Turk. 2004. Manifold based analysis of facial expression. In Proceedings of the Computer Vision and Pattern Recognition Workshop. 27, 81–85.

L. Jayawardhana, Y. Motai, and A. Docef. 2009. Computer-aided detection of polyps in CT colonography: On-line versus off-line accelerated kernel feature analysis. Special Issue on Processing and Analysis of High-Dimensional Masses of Image and Signal Data, Signal Processing 1–12.

X. Jiang, R. Snapp, Y. Motai, and X. Zhu. 2006. Accelerated kernel feature analysis In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 109–116.

A. Khan, J. A. Doucette, and R. Cohen. 2013. Validation of an ontological medical decision support system for patient treatment using a repository of patient data: Insights into the value of machine learning. ACM Transactions on Intelligent Systems Technology 4, 4, Article 68 (September 2013).

B. J. Kim and I. K. Kim. 2004. Incremental nonlinear PCA for classification. In Proceedings in Knowledge Discovery in Databases. 3202, 291–300.

B. J. Kim, J. Y. Shim, C. H. Hwang, I. K. Kim, and J. H. Song. 2003. Incremental feature extraction based on empirical kernel map. Foundations of Intelligent Systems. 2871, 440–444.

Y. Kim. 2007. Incremental principal component analysis for image processing. Optics Letters 32, 1, 32–34.

J. Kivinen, A. J. Smola, and R. C. Williamson. 2004. Online learning with kernels. IEEE Transactions on Signal Processing 52, 8, 2165–2176.

B. Levin, D. A. Lieberman, B. McFarland, K. S. Andrews, D. Brooks, J. Bond, C. Dash, F. M. Giardiello, S. Glick, D. Johnson, C. D. Johnson, T. R. Levin, P. J. Pickhardt, D. K. Rex, R. A. Smith, A. Thorson, and S. J. Winawer. 2008. Screening and surveillance for the early detection of colorectal cancer and adenomatous polyps, 2008: A joint guideline from the American Cancer Society, the US Multi-Society Task Force on Colorectal Cancer and the American College of Radiology. CA: A Cancer Journal for Clinicians 58, 130–160.

Y. M. Li. 2004. On incremental and robust subspace learning. Pattern Recognition 37, 7, 1509–1518.

Y. Motai and H. Yoshida. 2013. Principal composite kernel feature analysis: Data-dependent kernel approach. IEEE Transactions on Knowledge and Data Engineering 25, 8, 1863–1875.

J. Näppi and H. Yoshida. 2002. Automated detection of polyps in CT colonography: Evaluation of volumetric features for reduction of false positives. Academic Radiology 9, 386–397.

J. Näppi and H. Yoshida. 2003. Feature guided analysis for reduction of false positives in CAD of polyps for CT colonography. Medical Physics 30, 1592–1601.

J. Näppi and H. Yoshida. 2007. Fully automated three-dimensional detection of polyps in fecal-tagging CT colonography. Academy of Radiology 14 (March 2007), 287–300.

S. Ozawa, S. Pang, and N. Kasabov. 2008. Incremental learning of chunk data for online pattern classification systems. IEEE Transactions on Neural Networks 19, 6, 1061–1074.

C. Park and S. B. Cho. 2003. Genetic search for optimal ensemble of feature-classifier pairs in DNA gene expression profiles. In Proceedings of the International Joint Conference on Neural Networks. 3, 1702–1707.

P. J. Pickhardt, J. R. Choi, I. Hwang, J. A. Butler, M. L. Puckett, H. A. Hildebrandt, R. K. Wong, P. A. Nugent, P. A. Mysliwiec, and W. R. Schindler. 2003. Computed tomographic virtual colonoscopy to screen for colorectal neoplasia in asymptomatic adults. New England Journal of Medicine 349 (December 4, 2003), 2191–2200.

D. Regge and S. Halligan. 2013. CAD: How it works, how to use it, performance. European Journal of Radiology 82, 1171–1176.

D. Regge, C. Laudi, G. Galatola, Della P. Monica, L. Bonelli, G. Angelelli, R. Asnaghi, B. Barbaro, C. Bartolozzi, D. Bielen, L. Boni, C. Borghi, P. Bruzzi, M. C. Cassinis, M. Galia, T. M. Gallo, A. Grasso, C. Hassan, A. Laghi, M. C. Martina, E. Neri, C. Senore, G. Simonetti, S. Venturini, and G. Gandini. 2009. Diagnostic accuracy of computed tomographic colonography for the detection of advanced neoplasia in individuals at increased risk of colorectal cancer. Journal of the American Medical Association 301, 23, 2453–2461.

D. C. Rockey. 2010. Computed tomographic colonography: Ready for prime time? Gastroenterology Clinics of North America 39, 901–909.

D. C. Rockey, E. Paulson, D. Niedzwiecki, W. Davis, H. B. Bosworth, L. Sanders, J. Yee, J. Henderson, P. Hatten, S. Burdick, A. Sanyal, D. T. Rubin, M. Sterling, G. Akerkar, M. S. Bhutani, K. Binmoeller, J. Garvie, E. J. Bini, K. McQuaid, W. L. Foster, W. M. Thompson, A. Dachmanm, and R. Halvorsen. 2005. Analysis of air contrast barium enema, computed tomographic colonography, and colonoscopy: Prospective comparison. Lancet 365 (January 22, 2005), 305–311.

F. A. Sadjadi. 2008. Polarimetric radar target classification using support vector machines. Optical Engineering 47, 4.

B. Schölkopf and A. J. Smola. 2002. Learning with kernels: support vector machines, regularization, optimization, and beyond. Adaptive Computation and Machine Learning. MIT Press.

D. G. Stork and E. Yom-Tov. 2004. Computer Manual in MATLAB to Accompany Pattern Classification, 2nd ed. New York: John Wiley & Sons.

V. Taimouri, L. Xin, L. Zhaoqiang, L. Chang, P. Darshan, and H. Jing. 2011. Colon segmentation for prepless virtual colonoscopy. IEEE Transactions on Information Technology Biomedicine 15 (2011), 709–715.

V. F. van Ravesteijn, C. van Wijk, F. M. Vos, R. Truyen, J. F. Peters, J. Stoker, and L. J. van Vliet. 2010. Computer-aided detection of polyps in ct colonography using logistic regression. IEEE Transactions on Medical Imaging 29 (January 2010), 120–131.

H. Xiong, Y. Zhang, and X. W. Chen. 2007. Data-dependent kernel machines for microarray data classification. IEEE/ ACM Transactions on Computational Biology and Bioinformatics 4, 4.

J. H. Yao, M. Miller, M. Franaszek, and R. M. Summers. 2004. Colonic polyp segmentation in CT colonography-based on fuzzy clustering and deformable models. IEEE Transactions on Medical Imaging 23, 1344–1356.

J. Yee, D. H. Kim, M. P. Rosen, T. Lalani, L. R. Carucci, B. D. Cash, B. W. Feig, K. J. Fowler, D. S. Katz, M. P. Smith, and V. Yaghmai. 2014. ACR appropriateness criteria: Colorectal cancer screening. Journal of the American College of Radiologists 11, 543–551.

H. Yoshida and A. H. Dachman. 2005. CAD techniques, challenges and controversies in computed tomographic colonography. Abdominal Imaging 30, 26–41.

H. Yoshida and J. Nappi. 2001. Three-dimensional computer-aided diagnosis scheme for detection of colonic polyps. IEEE Transactions on Medical Imaging 20 (December 2001), 1261–1274.

H. Yoshida and J. Näppi. 2007. CAD in CT colonography without and with fecal tagging: Progress and challenges. Computerized Medical Imaging and Graphics 31, 267–284.

H. Yoshida, Y. Masutani, P. MacEneaney, D. Rubin, and A. H. Dachman. 2002. Computerized detection of colonic polyps at CT colonography on the basis of volumetric features: Pilot study. Radiology 222, 327–336.

H. Yoshida, Y. Wu, and W. Cai. 2012. Scalable, high-performance 3D image computing platform: System architecture and application to virtual colonoscopy. In Proceedings of the IEEE Engineering and Medical Biology Society. 3994–3997.

H. T. Zhao, P. C. Yuen, and J. T. Kwok. 2006. A novel incremental principal component analysis and its application for face recognition. IEEE Transactions on Systems, Man, and Cybernetics, Part B 36, 4, 873–886.

W. Zheng, C. Zou, and L. Zhao. 2005. An improved algorithm for kernel principal component analysis. Neural Processing Letters 49–56.

Received March 2014; revised July 2014; accepted September 2014