International Journal of Pattern Recognition
and Artificial Intelligence
Vol. 17, No. 3 (2003) 459–486


# A SURVEY ON PATTERN RECOGNITION APPLICATIONS OF SUPPORT VECTOR MACHINES

HYERAN BYUN*

*Department of Computer Science, Yonsei University,*
*Shinchon-dong, Seodaemun-gu, Seoul 120-749, Korea*
*hrbyun@cs.yonsei.ac.kr*

SEONG-WHAN LEE

*Department of Computer Science and Engineering, Korea University,*
*Anam-dong, Seongbuk-ku, Seoul 136-701, Korea*
*swlee@image.korea.ac.kr*

In this paper, we present a survey on pattern recognition applications of Support Vector Machines (SVMs). Since SVMs show good generalization performance on many real-life data and the approach is properly motivated theoretically, it has been applied to wide range of applications. This paper describes a brief introduction of SVMs and summarizes its various pattern recognition applications.

*Keywords*: Support Vector Machines; pattern recognition; face detection; face recognition; object recognition; handwritten character recognition; speech recognition.

## 1. Introduction

The SVM is a new type of pattern classifier based on a novel statistical learning technique that has been recently proposed by Vapnik and his co-workers.[13,19,101] Unlike traditional methods such as neural networks, which minimize the empirical training error, SVMs aim at minimizing an upper bound of the generalization error through maximizing the margin between the separating hyperplane and the data.[3] Since SVMs are known to generalize well even in high dimensional spaces under small training sample conditions[48] and have been shown to be superior to the traditional empirical risk minimization principle employed by most of neural networks,[65] SVMs have been successfully applied to a number of pattern recognition applications involving face detection, verification, and recognition,[1,2,10,15,24,33,38,40,44,48,53,58,59,61,62,64,65,67,68,75–77,79,84,87,92,97,99,104,106–108] object detection and recognition,[26,49,63,73,80,83,89,90] handwritten digit and character recognition,[9,18,30,74,98,116] speech and speaker verification, and

*Author for correspondence.

recognition,[11,22,29,66,102] information and image retrieval,[21,34,41,100,115] gender classification,[71,103,110] prediction,[23,25,27,69,96] text detection and categorization,[6,45,46,47,93] and so on.[4,7,8,20,36,39,51,56,70,85,109,112–114]

This paper is organized as follows. We give a brief explanation on SVMs in Sec. 2 and a survey on pattern recognition applications of support vector machines in Sec. 3. Section 4 describes the limitations of SVMs and conclusion is given in Sec. 5.

## 2. Support Vector Machines

Classical learning approaches are designed to minimize error on the training dataset and it is called the Empirical Risk Minimization (ERM). Those learning methods follow the ERM principle and neural networks are the most common example of ERM. On the other hand, the SVMs are based on the Structural Risk Minimization (SRM) principle rooted in the statistical learning theory. The SVM has better generalization abilities for unseen test data and achieves SRM through a minimization of the upper bound which is the sum of the training error rate and a term that depends on VC dimension of the generalization error.[13,16,19,35,101,116]

### 2.1. *Linear support vector machines for linearly separable case*

The basic idea of the SVMs is to construct a hyperplane as the decision plane, which separates the positive (+1) and negative (−1) classes with the largest margin, which is related to minimizing the VC dimension of SVM. In a binary classification problem where feature extraction is initially performed, let us label the training data $\mathbf{x}_i \in \mathbf{R}^d$ with a label $y_i \in \{-1, +1\}$, for all $i = 1, \cdots, l$, where $l$ is the number of data, and $d$ is the dimension of the problem. When the two classes are linearly separable in $\mathbf{R}^d$, we wish to find a separating hyperplane which gives the smallest generalization error among the infinite number of possible hyperplanes. Such an optimal hyperplane is the one with the maximum margin of separation between the two classes, where the margin is the sum of the distances from the hyperplane to the closest data points of each of the two classes. These closest data points are called Support Vectors (SVs). The solid line on Fig. 1(a) represents the optimal separating hyperplane.

Let us suppose they are completely separated by a $d$-dimensional hyperplane described by

$$\mathbf{w} \cdot \mathbf{x} + b = 0 \tag{1}$$

The separation problem is to determine the hyperplane such that $\mathbf{w} \cdot \mathbf{x}_i + b \geq +1$ for positive examples and $\mathbf{w} \cdot \mathbf{x}_i + b \leq -1$ for negative examples. Since the SVM finds the hyperplane which has the largest margin, it can be found by maximizing $1/||\mathbf{w}||$. The optimal separating hyperplane can thus be found by minimizing Eq. (2) under the constraint (3) to correctly separate the training data.

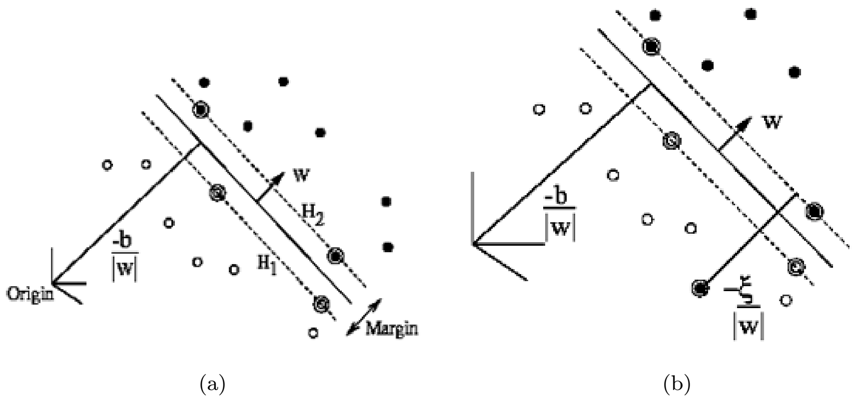



Fig. 1. (a) Linear separating hyperplanes for the separable case. (b) Linear separating hyperplane for the nonseparablecase. The support vectors are circled.[16]

$$\min_{\mathbf{w}} \ \tau(\mathbf{w}) = \frac{1}{2}\,||w||^2 \tag{2}$$

$$y_i(\mathbf{x_i} \cdot \mathbf{w} + b) - 1 \geq 0\,, \quad \forall\, i \tag{3}$$

This is a Quadratic Programming (QP) problem for which standard techniques (Lagrange Multipliers, Wolfe dual) can be used.[35,50,77,81] The detailed explanation on QP problems and alternative researches are described in Sec. 2.4.

### 2.2. *Linear support vector machines for nonseparable case*

In practical applications for real-life data, the two classes are not completely separable, but a hyperplane that maximizes the margin while minimizing a quantity proportional to the misclassification errors can still be determined. This can be done by introducing positive slack variables $\xi_i$ in constraint (3), which then becomes

$$y_i(\mathbf{x_i} \cdot \mathbf{w} + b) \geq 1 - \xi_i\,, \quad \forall\, i \tag{4}$$

If an error occurs, the corresponding $\xi_i$ must exceed unity, so $\sum_i \xi_i$ is an upper bound for the number of misclassification errors. Hence the objective function $\tau(\cdot)$ in (2) to be minimized can be changed into

$$\tau(\mathbf{w},\, \xi) = \frac{1}{2}\,||w||^2 + C\sum_{i=1}^{l} \xi_i \tag{5}$$

where $C$ is a parameter chosen by the user that controls the tradeoff between the margin and the misclassification errors $\xi = (\xi_1, \cdots, \xi_l)$. A larger $C$ means that a higher penalty to misclassification errors is assigned. Minimizing Eq. (5) under constraint (4) gives the *Generalized Separating Hyperplane*. This still remains a QP problem. The nonseparable case is illustrated in Fig. 1(b).

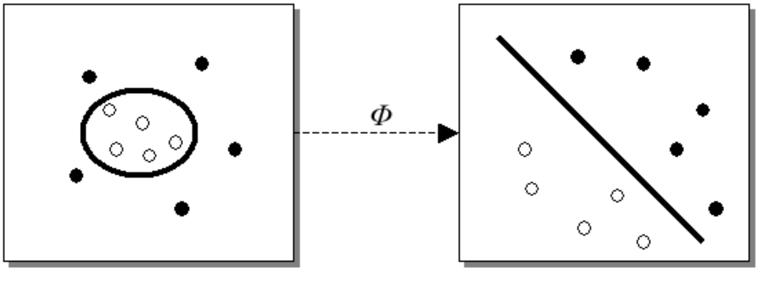
## 2.3. *Nonlinear support vector machines and kernels*

### 2.3.1. *Nonlinear support vector machines*

An extension to nonlinear decision surfaces is necessary since real-life classification problems are hard to be solved by a linear classifier.[96] When the decision function is not a linear function of the data, the data will be mapped from the input space into a high dimensional feature space by a nonlinear transformation. In this high dimensional featured space, the generalized optimal separating hyperplane shown in Fig. 2 is constructed.[35] Cover's theorem states that if the transformation is nonlinear and the dimensionality of the feature space is high enough, then input space may be transformed into a new feature space where the patterns are linearly separable with high probability.[37] This nonlinear transformation is performed in implicit way through so-called kernel functions.

### 2.3.2. *Inner-product kernels*

In order to accomplish nonlinear decision function, an initial mapping $\Phi$ of the data into an (usually significantly higher dimensional) Euclidean space $H$ is performed as $\Phi : \mathbf{R}^n \to H$, and the linear classification problem is formulated in the new space with dimension $d$. The training algorithm then only depends on the data through dot product in $H$ of the form $\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$. Since the computation of the dot products is prohibitive if the dimension of transformed training vectors $\Phi(\mathbf{x}_i)$ is very large, and since $\Phi$ is not known *a priori*, the Mercer's theorem[16] for positive definite functions allows to replace $\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$ by a positive definite symmetric kernel function $k(\mathbf{x}_i, \mathbf{x}_j)$, that is, $k(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$. In training phase, we need only kernel function and $\Phi$ does not need to be known since it is implicitly defined by the choice of kernel. The data can become linearly separable in feature space although original input is not linearly separable in the input space. Hence kernel substitution provides a route for obtaining nonlinear algorithms from algorithms previously restricted to handling linear separable datasets.[17] The use of




(a) Input space (b) Feature space

Fig. 2. Feature space is related to input space via a nonlinear map, causing the decision surface to be nonlinear in the input space.[34]

Table 1. Summary of inner-product kernels.[37]

| Kernel Function | Inner Product Kernel $K(\mathbf{x}, \mathbf{x}_i), i = 1, 2, \ldots, N$ |
|---|---|
| Polynomial kernel | $K(\mathbf{x}, \mathbf{x}_i) = (\mathbf{x}^T \mathbf{x}_i + 1)^d$ |
| Gaussian kernel | $K(\mathbf{x}, \mathbf{x}_i) = \exp(-\Vert\mathbf{x} - \mathbf{x}_i\Vert^2 / 2\sigma^2)$ |
| Multi-layer perceptron (sigmoid) | $K(\mathbf{x}, \mathbf{x}_i) = \tanh(\beta_0 \mathbf{x}^T \mathbf{x}_i + \beta_1)$, $\beta_0$ and $\beta_1$ are decided by the user |

implicit kernels allows reducing the dimension of the problem and overcoming the so-called "dimension curse".[101] Variant learning machines are constructed according to the different kernel function $k(\mathbf{x_i}, \mathbf{x_j})$ and thus construct different hyperplanes in feature space. Table 1 shows three typical kernel functions.

## 2.4. *Quadratic programming problem of SVMs*

### 2.4.1. *Dual problem*

In Eqs. (2) and (3), the optimization goal $\tau(\mathbf{w})$ is quadratic and the constraints are linear, so it is a typical QP. Given such a constrained optimization problem, it is possible to construct another problem called the dual problem. We may now state the dual problem: *given the training sample* $\{(\mathbf{x}_i, d_i)\}_{i=1}^N$, *find the Lagrange multipliers* $\{\alpha_i\}_{i=1}^N$ *that maximize the objective function*

$$Q(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j d_i d_j \mathbf{x}_i^T \mathbf{x}_j \tag{6}$$

subject to the constraints

$$\sum_{i=1}^N \alpha_i d_i = 0\,, \tag{7}$$

$$\alpha_i \geq 0, \quad \text{for all } i = 1, \ldots, N\,. \tag{8}$$

We may also formulate the dual problem for nonseparable patterns using the method of Lagrange multipliers. Given the training sample $\{(\mathbf{x}_i, d_i)\}_{i=1}^N$, *find the Lagrange multipliers* $\{\alpha_i\}_{i=1}^N$ *that maximize the objective function*

$$Q(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j d_i d_j \mathbf{x}_i^T \mathbf{x}_j \tag{9}$$

subject to the constraints

$$\sum_{i=1}^N \alpha_i d_i = 0\,, \tag{10}$$

$$0 \leq \alpha_i \leq C, \quad \text{for all } i = 1, \ldots, N\,, \tag{11}$$

where $C$ is a user-chosen positive parameter. The objective function $Q(\alpha)$ to be maximized for the case of nonseparable problems in the dual problem is the same as the case for the separable problems except for a minor but important difference. The difference is that the constraints $\alpha_i \geq 0$ for the separable case is replaced with the more stringent constraint $0 \leq \alpha_i \leq C$ for the nonseparable case.[37]

### 2.4.2. *How to solve the quadratic problem*

A number of algorithms have been suggested for solving the dual problems. Traditional QP algorithms[91,95] are not suitable for large size problems because of the following reasons[50]:

- They require that the kernel matrix is computed and stored in memory, so it requires extremely large memory.
- These methods involve expensive matrix operations such as the Cholesky decomposition of a large submatrix of the kernel matrix.
- For practitioners who would like to develop their own implementation of an SVM classifier, coding these algorithms is very difficult.

A few attempts have been made to develop methods that overcome some or all of these problems. Osuna *et al.*[77] proved a theorem, which suggests a whole new set of QP problems for SVM. The theorem proves that the large QP problem can be broken down into a series of smaller QP subproblems. As long as at least one example that violate the Karush–Kuhn–Tucker (KKT) conditions is added to the examples for the previous subproblem, each step will reduce the cost of overall objective function and maintain a feasible point that obeys all of the constraints. Therefore, a sequence of QP subproblems that always add at least one violator will be guaranteed to converge.[77]

Platt proposed a Sequential Minimal Optimization (SMO) to quickly solve the SVM QP problem without any extra matrix storage and without using numerical QP optimization steps at all. Using Osuna's theorem to ensure convergence, SMO decomposes the overall QP problem into QP subproblems. The difference of the Osuna's method is that SMO chooses to solve the smallest possible optimization problem at every step. At each step, (1) SMO chooses two Lagrange multipliers to jointly optimize, (2) finds the optimal values for these multipliers, and (3) updates the SVMs to reflect the new optimal values. The advantage of SMO is that numerical QP optimization is avoided entirely since solving for two Lagrange multipliers can be done analytically. In addition, SMO requires no extra matrix storage at all. Thus, very large SVM training problems can fit inside the memory of a personal computer or workstation.[81] Keerthi *et al.*[50] pointed out an important source of confusion and inefficiency in Platt's SMO algorithm that is caused by the use of single threshold value. Using clues from the KKT conditions for the dual problem, two threshold parameters are employed to derive modifications of SMO.

### 2.5. *SVMs applied to multiclass classification*

The basic SVM is for two-class problem. However it should be extended to multi-class to apply to the multi-class problems . There are two basic strategies for solving $q$-class problems with SVMs : one-to-others and tree-structure (pairwise SVMs and DDAG).

#### 2.5.1. *One-to-others multiclass SVMs*[116]

Take the training samples with the same label as one class and the others as the other class, then it becomes a two-class problem.[116] For the $q$-class problem ($q > 2$), $q$ classifiers are formed and denoted by $\mathrm{SVM}_i$, $i = 1, 2, \cdots, q$. As for the testing sample $\mathbf{x}$, $d_i(\mathbf{x}) = \mathbf{w}_i^* \cdot \mathbf{x} + b_i^*$ can be obtained by using $\mathrm{SVM}_i$. The testing sample $\mathbf{x}$ belongs to the class $j$ where

$$d_j(\mathbf{x}) = \max_{i=1,\cdots,q} d_i(x) \tag{12}$$

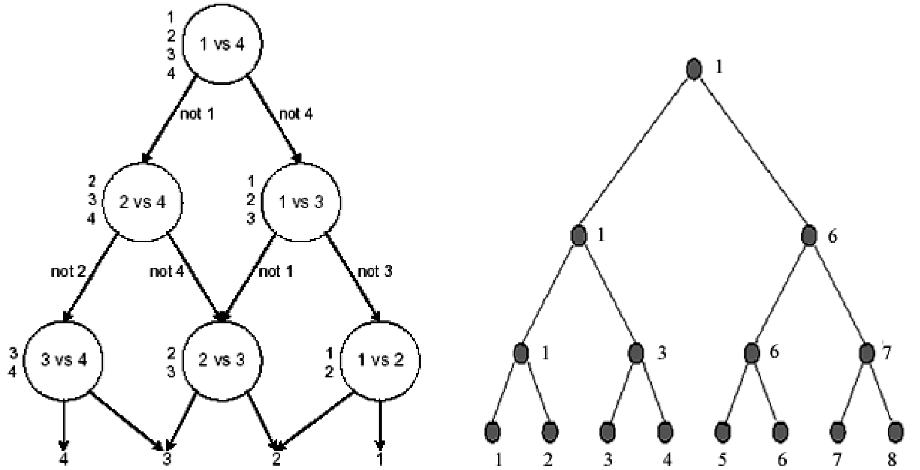
#### 2.5.2. *Tree structured multiclass SVMs*: *pairwise SVMs and DDAG SVMs*

In the pairwise approach, machines are trained for $q^2$-class problem.[83] All these SVM classifiers must be used for classifying the testing samples and the synthesizing result is gotten. The pairwise classifiers are arranged in trees, where each tree node represents a SVM. A bottom-up tree which is similar to the elimination tree used in tennis tournaments was originally proposed by Pontil and Verri[83] for recognition of 3D objects and was applied to face recognition by Guo *et al.*.[32,33] A top-down tree structure called Decision Directed Acyclic Graph (DDAG) has been recently proposed in Platt *et al.*'s paper.[82] There is no theoretical analysis of the two strategies with respect to the classification performance.[38] Regarding the training effort, the one-to-others approach is preferable since only $q$ binary SVMs in one-to-others have to be trained compared to $q(q-1)/2$ binary SVMs in the pairwise approach. However, at runtime both strategies require the evaluation of $q$ SVMs.[38] Recent experiments on person recognition show similar classification performance for the two strategies: one-to-others and tree-structured methods.[73]

Also Hsu and Lin[42] compared the above methods based on three types of binary classification: one-to-others, pairwise and DDAG SVM. Their experiments indicated that pairwise and DDAG SVM methods are more suitable for practical use than the one-to-others method.

## 3. Pattern Recognition Applications of SVMs

In this section, we survey applications of pattern recognition using SVMs. We classify existing methods into roughly five categories according to their aims. Some methods, which are not included in these categories, are summarized in Sec. 3.6. Osuna *et al.*[77] first demonstrated the applicability of SVM by embedding SVM in




(a) Example of top-down tree structure (DDAG)

(b) Example of bottom-up tree structure (Pairwise SVM)

Fig. 3. Tree structure for multi-class SVMs. (a) The DDAG for finding the best class out of four classes. (b) The binary tree structure for eight classes. For a coming test image, it is compared with each two pairs, and the winner will be tested in an upper level until the top of the tree. The numbers 1–8 encode the classes.[32,33]

face detection system which performs comparable recognition results to the state-of-the-art system. The reason to investigate the use of SVM is the fact that (1) SVMs are very well founded from the mathematical point of view, being an approximate implementation of the Structural Risk Minimization induction rule and (2) the adjustable parameters are only $C$ and the kernel functions.[77] There are many publicly available free software such as SVMFu, SVMLight, LIBSVM, SVMTorch, etc. and a brief summary of these software is described in Table 2.

### 3.1. *Face detection and recognition*

Face detection, verification and recognition are one of the popular issues in biometrics, identity authentication, access control, video surveillance and human–computer interfaces. There are many active researches in this area for all these applications using different methodologies. However, it is very difficult to achieve a reliable performance. The reasons are due to the difficulty of distinguishing different persons who have approximately the same facial configuration and wide variations in the appearance of a particular face. These variations are because of changes in pose, illumination, facial makeup and facial expression.[104] Also glasses or a moustache makes difficult to detect and recognize faces. Recently many researchers applied SVMs to face detection, facial feature detection, face verification and recognition and compared their results with other methods. Each method used different input features, different databases, and different kernels for SVM classifier.

Table 2. Some examples of publicly available SVM software.

| Software | Developer | Language | Environment | Algorithms | URL |
|---|---|---|---|---|---|
| SVMFu | R. Rifkin M. Nadermann (MIT) | C++ | Unix-like system | Osuna *et al.*, SMO (Platt) | http://www.ai. mit.edu |
| LIBSVM | C. C. Chang, C.H. Lin (National Taiwan Univ.) | C++, Java | Python, R, Matlab, Perl | SMO (Platt), SVMLight (Joachims) | http://www.csie. ntu.edu.tw/ libsym |
| SVMLight | T. Joachims, (Univ. of Dortmund) | C | Solaris, Linux, IRIX, Windows NT | T. Joachims | http://www.svmlight. joachims.org |
| SVMTorch | R. Collobert, (IDIAP, Switzerland) | C, C++ | Windows | R. Collobert | http://www.idap.ch /learning/SVMTorch. html |

**Face Detection**: A SVM can be used to distinguish face and non-face images since a face detection problem is a binary classification problem. The application of SVM for frontal face detection in image was first proposed by Osuna *et al.*[77] The proposed algorithm scanned input images with a $19 \times 19$ window and a SVM with a second degree polynomial as a kernel function is trained with a novel decomposition algorithm, which guarantees global optimality. This system was able to handle up to a small degree of non-frontal views of faces.

Although non-face examples are abundant, non-face examples that are useful from a learning point of view are very difficult to characterize and define.[77] To solve this problem, bootstrap was the most popular method and also hierarchical linear SVMs were used to exclude non-face images step by step and more complex non-linear SVM verified the face in the last step.[2,67,68] By Ma *et al.*,[68] five hierarchical linear SVMs to exclude non-face used different $C$'s with $C_{\text{face}}$ being 100, 50, 10, 5, 5 times of $C_{\text{non-face}}$ indicating different cost-sensitivity.

To avoid the scanning of the whole image to decide face or non-face, many papers applied their methods on the skin-color segmented region.[58,79,84,99] Kumar and Poggio[58] recently incorporated Osuna *et al.*'s SVM algorithm in a system for real-time tracking and analysis of faces on skin region and also to detect eyes. In Qi *et al.*'s paper,[84] SVMs used the ICA features as an input after applying skin color filter for face detection and they showed that the used ICA features gave better generalization capacity than by training SVM directly on the whole image data. In Terrillon *et al.*,[99] they applied SVM to invariant Orthogonal Fourier–Mellin Moments as features for binary face or non-face classification on skin color-based segmented image and compared the performance of SVM face detector to multi-layer perceptron in terms of Correct Face Detector (CD) and Correct Face Rejection

(CR). In Xi and Lee's paper,[108] LH and HL subimages of wavelet decomposition are used as a feature vector for face detection. Also, in order to speed up the face detection, Ai *et al.*[1] have used two templates of eyes-in-whole and face in filtering out face candidates for SVMs to classify face and non-face classes. Another method to improve the speed of the SVM algorithm found a set of reduced support vectors (RVs) which are calculated from support vectors.[87] RVs are used to speed up the calculation sequentially.

For the input feature vectors to SVMs, Xi *et al.*[106–108] and Huang *et al.*[44] used component-based feature vectors such as eye brows, eyes, mouth as an input to SVMs and showed their component-based face detection performed well to the whole input image. In order to extract the 14 facial components by SVMs, Bileschi and Heisele[12] trained each facial component only on positive examples of face images. The negative training data for each component is extracted from four random crops to overlap the component by no more than 35% of the area of each component in face images. The performance of complete system using SVM classifiers trained on facial negatives for each facial component detection outperformed.

SVMs have also been used for multi-view face detection by constructing separate SVMs specific to different views based on the pose estimation. For face recognition, frontal view SVM-based face recognizer is used if the detected face is in frontal view after head pose estimation.[62,75,76] Also combined methods are tried to improve the performance for face detection. Li *et al.*[61] tested the performance of three face detection algorithms, eigenface method, SVM method and combined method in terms of both speed and accuracy for multi-view face detection. The combined method consisted of a coarse detection phase by eigenface method, and then the ambiguous outputs of eigenface methods are tested by a fine SVM phase so that an improved performance could be achieved by speeding up the computation and keeping the accuracy. Buciu *et al.*[15] attempted to improve the performance of face detection by majority voting on the outputs of five different kernels of SVM. Papageorgio *et al.*[79] applied SVM to overcomplete wavelet representation as input data to detect faces and people and Richman *et al.*[86] applied SVMs to find nose cross-section for face detection. The summary of face detection using SVM is given in Table 3 in terms of feature vectors, different databases, detection rate, different kernels, and SVM software used. The benchmark test sets[111] for face detection are MIT data set, CMU-set A, CMU-set B, Kodak Data Set, M2VTS, etc. The descriptions are followings:

- **MIT data set**: two sets of high (301 frontal and near frontal mugshots of 71 different people) and low (23 images with 149 faces) gray-scale images with multiple faces in complex background.
- **CMU frontal face (set A, B, C, D)**: 130 gray scale images with a total of 507 frontal faces.
- **CMU profile face set**: 208 gray-scale images with faces in profile views.
- **CMU-PIE database**: pose, illumination, expression face database.

Table 3. Summary of face detection by SVMs.

| Method | Feature Vectors | Database | Detection Rate/ False Alarms | SVM Software | Kernel |
|---|---|---|---|---|---|
| *(baseline)* *Rowley et al.*[88] | *System 5* *System 11* | *CMU-set A* *CMU-set B* | *90.5% (570)* *86.2% (23)* | *– Neural Network Methods* *– Used for baseline algorithm* | |
| Osuna *et al.*[77] | 19×19 gray image | 313: single face 23: multi-faces with a total of 155 faces | 97.4% (4) 74.2% (20) | decomposition algorithm | 2nd polynomial (C = 200) |
| Qi *et al.*[84] | ICA features | 820 face images: LAMP (Univ. of Maryland) and Essex face database | 92.5% (54) | N/A | N/A (C = 230) |
| Terrillon *et al.*[99] | Invariant Orthogonal Fourier-Mellin Moments | Own database 100 images with 144 face images | CD: 93.1% CR: 72.8% | N/A | RBF |
| Romdhani *et al.*[87] | 20×20 gray image | CMU-set A | 81.9% (465) | SMO | RBF (2207 SVs, C = 200) |
| Bassiou *et al.*[10] | mosaic image (multireolution images) | M2VTS best case (set 4) | FAR: 0% FRR: 0% | N/A | Polynomial, RBF, Linear, Sigmoid (C = 1000) |
| Xi and Lee[108] | Wavelet LH, HL sub-images | Own database, Set A: 325 images (one face per image) Set B: 136 images (with more than 2 faces) | Set A: 98.1% (0.3% FR) Set B: 75.4% (13.4% FR) | N/A N/A | N/A |
| Ma and Ding[68] | 20×20 gray images | CMU face data | 88.9% (23) | N/A | 5 linear SVMs 2nd polynomial |
| Ma and Ding[67] | 20×20 gray images | Nokia test set CMU face data | 90.1% (148) 87.2% (156) | N/A | 5 linear SVMs (C = 10) Nonlinear SVM (C = 200) |

- **Kodak data set**: faces of multiple size, pose and various illumination in color images.
- **M2VTS**: video sequences of 37 different subjects in four different shots.
- **XM2VTS**: video sequences of 295 different subjects taken over a period of four months.

**Face Recognition and Authentication**: The recognition of a face is a well-established field of research and a large number of algorithms have been proposed in the literature. Machine recognition of faces yields problems that belong to the following categories whose objectives are briefly outlined[97]:

- Face Recognition: Given a test face and a set of reference faces in a database, find the $N$ most similar reference faces to the test face.
- Face Authentication: Given a test face and a reference one, decide if the test face is identical to the reference face.

Guo *et al.*[32,33] proposed multi-class SVM with pairwise bottom-up tree strategy for face recognition and compared SVM result with Nearest Center (NC), Hidden Markov Model (HMM), Conventional Neural Network (CNN), and Nearest Feature Line (NFL). Normalized feature extracted by PCA was the input of the SVM classifier. Error rates of NC, HMM, CNN, NFL, and SVM are 5.25%, 5%, 3.83%, 3.125%, and 3.0% respectively on ORL face database.

Heisele *et al.*[38,44] proposed a component-based method and compared the performance with two global methods for face recognition by one-to-others SVMs. Huang *et al.*[44] generated a large number of synthetic face images to train the system by rendering the 3D models under various poses and illumination. In component-based system, they extracted facial components and combined them into a single feature vector, which is classified by SVMs. The global systems used SVMs to recognize faces by classifying a single feature vector consisting of the gray values of the whole face image. One global method used single SVM and the other used view-based SVMs. Their results showed that the component-based method outperformed the global methods.

Kim *et al.*[53] used modified SVM local correlation kernel to explore spatial relationships among potential eye, nose, and mouth objects and compared their kernel with existing kernels with error rate of 2% on ORL database. Wang *et al.*[104] proposed a face recognition algorithm based on both 3D range and 2D gray-level facial images. 2D texture (Gabor Coefficient) and 3D shape features (Point Signature) are projected onto PCA subspace and then integrated 2D and 3D features are as an input of SVM to recognize faces.

For face authentication and recognition, Jonsson *et al.*[48] presented that SVMs could extract the relevant discriminative information from the training data and the performance of SVMs was relatively insensitive to the representation space and preprocessing steps. To prove this, they performed a number of experiments with different decision rules (Euclidean distance, normalized correlation, SVMs), subspaces (PC, LD), and preprocessing (no preprocessing, zero-mean and unit variance, histogram equalization). A SVM with histogram equalization and LD subspace showed the best performance of EER = 1.00, FAR = 1.37, FRR = 0.75. Tefas *et al.*[97] reformulated Fisher's discriminant ratio to a quadratic optimization problem subject to a set of inequality constraints to enhance the performance of morphological elastic graph matching (MEGM) for frontal face authentication. SVMs which find the optimal separating hyperplane are constructed to solve the reformulated quadratic optimization problem for face authentication. These optimal coefficients by SVMs are used to weigh the raw similarity vectors that are provided by the MEGM and the best performance of the frontal face verification on M2VTS face database is EER = 2.4. The summary of face verification and recognition performance is given in Table 4.

Table 4. Summary of face verification and recognition by SVMs.

| Category | Feature Vectors | Database | Recognition Rate | Type of Multi-Class (Kernel) | Remarks |
|---|---|---|---|---|---|
| Face verification[48] | LDA | M2VTS | EER = 1.00<br>FAR = 1.37<br>FRR = 0.75 | N/A<br>(RBF kernel) | SVM is relatively insensitive to the representations space and preprocessing steps |
| | PCA | M2VTS | EER = 1.50<br>FAR = 2.19<br>FRR = 1.50 | N/A<br>(RBF kernel) | |
| Face verification[97] | Weighted MEGM | M2VTS | EER = 2.4 | N/A<br>(Nonlinear kernel) | |
| Face recognition[33] | PCA | ORL database | 97% | Pairwise<br>(N/A) | NC (94.75%),<br>CNN (96.17%)<br>HMM (95%),<br>NFL (96.88%) |
| Face recognition[38,44] | Component based | Own database | ROC curve | One-to-others<br>(linear kernel) | Component based method (outperformed) global based method |
| | Whole image | Own database | ROC curve | One-to-others<br>(2nd order polynomial) | |
| Face recognition[53] | Feature-based (component based) | ORL database | 98% | One-to-others<br>(N/A) | Eigenface (90%) |
| Face recognition[104] | PCA of PS + GC | Own database | about 90% | DDAG<br>(N/A) | |

### 3.2. *Object detection and recognition*

Object detection or recognition aims to find and track moving people or traffic situation for surveillance or traffic control. Nakajima *et al.*[73] developed people recognition and pose estimation as a multi-class classification problem. This paper used pairwise and DDAG multi-class SVMs with linear kernel and the two types of SVM classifiers showed very similar performance. They recognized four people and four poses (left, right, front, back side) with 640 images of 40 images per each person at each pose of their own data. Two features (color histogram and local shape) are tested for people and pose recognition and local shape feature gave better performance. 3D object recognition was developed by Roobaert and Van Hulle,[89] and Pontil and Verri.[83] Both of them used COIL object database, which contained 7200 images of 100 objects with 72 different views per each object. Roobaert and Van Hulle[89] proposed 3D object recognition with SVMs to illustrate the potential of SVMs in terms of the number of training views per object. Their result showed that the performance was decreased much when the number of training views was less than 18 views. Pontil and Verri[83] used linear SVMs for aspect-based 3D object recognition from a single view without feature extraction, data reduction and estimating pose. They tested SVM method on the synthesized images of COIL database with noise, occlusion, and pixel shifts and got very good performance. The result showed that SVMs are well-suited for aspect-based recog-

Table 5. Summary of object detection and recognition by SVMs.

| Category | Input | Database | Recognition Rate | Type of Multi-Class | Remarks |
|---|---|---|---|---|---|
| People recognition[73] | Color feature<br>Shape feature<br>Color feature<br>Shape feature | Own Database | 91.6%<br>99.5%<br>91.4%<br>99.5% | Pairwise linear SVMs<br>DDAG linear SVMs | Shape feature is better<br>No performance difference between pairwise and DDAG SVMs |
| Pose estimation[73] | Color feature<br>Shape feature<br>Color feature<br>Shape feature | Own Database | 68.2%<br>84.3%<br>68.0%<br>84.5% | Pairwise linear SVMs<br>DDAG linear SVMs | |
| 3D object recognition[83,89] | 32×32 image | COIL database | 99.7% (plain images)<br>99.7% (with noise)<br>99.4% (3 pixel shift) | Pairwise linear SVMs | Tested on the most difficult 32 objects out of 72 objects |
| Pedestrian detection[49] | 32×32 vertical edges | Own database | 100% with FD = 0.01% | 3rd order polynomial | |

nition. Pittore *et al.*[80] developed VIDERE (VIsual Dynamic Event REcogniton) system. They proposed a system that was able to detect the presence of moving people, represented the event by using a SVM for regression, and recognized trajectory of visual dynamic events from an image sequence by SVM classifier. Gao *et al.*[26] proposed a shadow and head-lights elimination algorithm by considering this problem as a two-class problem. That is, the SVM classifier was used to detect real moving vehicles from shadows. Some other object recognitions were on radar target recognition,[63] pedestrian detection[49] and recognition.[105] Kang *et al.*[49] used $32 \times 64$ size of vertical edges as features to detect pedestrians by a SVM. Their system could detect pedestrians in different size, pose, gait, clothing and occulusions. The brief summary of object detection and recognition is given in Table 5.

### 3.3. *Handwritten character recognition*

Among the SVM-based applications, SVMs have shown to largely outperform all other learning algorithms for handwritten digit and character recognition problem. A major problem in handwriting recognition is the huge variability and distortions of patterns. To absorb these problems, Choisy and Belaid[18] used NSPH-HMM for local view and SVM for global view to recognize French bank check words. For handwritten digit recognition, SVMs are used by Gorgevik *et al.*,[30] Teow *et al.*[98] and Zhao *et al.*[116] Gorgevik *et al.*[30] used two different feature families (structural features and statistical features) for handwritten digit recognition using SVM classifier. They tested single SVM classifier applied on both feature families as one set. Also two feature sets are forwarded to two different SVM classifiers and the obtained results are combined by rule-based reasoning. The paper showed that the single SVM classifier gave better performance than rule-based reasoning which combined two individual SVM classifiers. Teow and Loe[98] had developed a vision-based handwritten digit recognition system, which extracts features that are biologically plausible, linearly separable and semantically clear. In this system, they showed that their extracted features were linearly separable features over a large set of training data in a highly nonlinear domain by using linear SVM classifier. Zhao *et al.*[116] showed the recognition performance of handwritten digits according to (1) the effect of input dimension, (2) effect of kernel functions, (3) comparison of different classifiers (ML, MLP, SOM+LVQ, RBF, SVM) and (4) comparison of three types of multi-class SVMs (one-to-others, pair-wise, DDAG).

To recognize hand-printed Hiragana, Naruyama *et al.*[74] proposed the combination of two multi-class SVM methods which are one-to-others SVMs with max-win and DDAG for cumulative recognition rate. They found some pairs which are difficult to discriminate and then combined these pairs into the same group. They first applied DDAG and then one-to-others SVMs. If the result of DDAG was in the same group, then they applied one-to-others with majority voting for the pairs of the same group. The result showed that the proposed modified DDAG is 30 times faster than one-to-others and almost equivalent to one-to-others with max-win in terms of the cumulative recognition rate. The brief summary of handwritten digit and character recognition is given in Table 6.

### 3.4. *Speaker recognition and speech recognition*

In speaker or speech recognition problem, the two most popular techniques are discriminative classifiers and generative model classifiers. The methods using discriminative classifiers consist of decision tree, neural network, SVMs, etc. The well-known generative model classification approaches include Hidden Markov models (HMM) and Gaussian Mixture models (GMM).[22] For training and testing data, there are text dependent and text independent data. Bengio and Mariethoz,[11]

Table 6. Summary of handwritten character recognition by SVMs.

| Category | Database | Features | Recognition rate | SVM software |
|---|---|---|---|---|
| Handwritten digit recognition[30] | NIST; 16×16 binary image | 54 structural features + 62 statistical features | 97.53% (Gaussian) 95.06% (linear) | SVM Torch |
| Handwritten digit recognition[116] | NIST; various input resolutions | Whole image | 97.21% (best case) | pairwise SVM |
| Handwritten digit recognition[98] | NIST; 36×36 gray image | Biologically motivated features | 99.18% | SVMLight; pairwise linear SVM |
| Handwritten Hiragana recognition[74] | JEITA-HP; 64×64 gray image | 256 directional features | 94.00% | SMO |

and Wan and Campbell[102] used SVMs for speaker verification on different data sets. They experimented on text dependent and text independent data and replaced the classical thresholding rule with SVMs to decide accept or reject. Text independent tasks gave significant performance improvements. Wan and Campbell[102] proposed a new technique for normalizing the polynomial kernel to use with SVMs and tested on YOHO database. Dong and Zhaohui[22] reported on the development of a natural way of achieving combination of discriminative classifier and generative model classifiers by embedding GMM in SVM outputs, and thus created a continuous density support vector machine (CDSVM) for text independent speaker verification. For utterance verification which is essential to accept keywords and reject non-keywords on spontaneous speech recognition, Ma *et al.*[66] trained and tested SVM classifier as the confidence measurement problem in speech recognition.

SVM is also applied to the visual speech recognition which recognizes speech by their lipreading. Viseme is defined by a mouth shape and mouth dynamics corresponding to the production of a phone or a group of phones indistinguishable in the visual domain. Each viseme is described by SVM and Vitterbi algorithm used SVMs as nodes for modeling the temporal character of speech. To evaluate the performance, they experimented on audio-visual data Tuplip 1 to solve the task of recognizing the first four digits in English.[28,29] The brief summary of speaker and speech recognition is given in Table 7.

Table 7. Summary of speaker and speech recognition by SVMs.

| Category | Database | Input | Recognition rate | Remarks |
|---|---|---|---|---|
| Speaker verification[11] — text independent | PolyVar telephone database | 39 coefficient features: 12LPC + deltas (HMM: generative model) | HTER = 1/2 (% FA+%FR) 4.73% (RBF kernel) 5.55 % (Bayes decision) | SVM Torch |
| Speaker verification[11] — text dependent | CAVE | 39 coefficient features: 12LPC + deltas (GMM: generative model) | HTER 3.34 % (RBF kernel) 3.40 % (Bayes decision) | SVM Torch |
| Speaker verification[102] — text independent | YOHO | 24 coefficient features: 12th order LPC + deltas | EER: 0.34% (seen) EER: 0.59% (unseen); normalized 10th order polynomial kernel EER: 1.86% (unseen); unnormalized RBF kernel | N/A |
| Utterance verification[66] | Own data | 4 features: normalized score, score per frame, word duration, speech rate | ER: 1% with 7% rejection rate | N/A |
| Visual speech recognition (lipreading)[29] | Tulips1 | 16×16 mouth region gray image and delta features | Word recognition rate: 90.6% (3rd order polynomial kernel) | SVM Light |

### 3.5. *Information and image retrieval*

Content-based image retrieval is emerging as an important research area with applications to digital libraries and multimedia databases.[34] Guo *et al.*[34] proposed a new metric, distance-from-boundary to retrieve the texture image. The boundaries between classes are obtained by SVM. To retrieve more images or information relevant to the query image, SVM classifier is used to separate two classes of relevant images and irrelevant images.[21,100,115] Drucker *et al.*[21], Tian *et al.*[100] and Zhang *et al.*[115] proposed that SVMs automatically generated preference weights for relevant images or information. The weights were determined by the distance of the two separating hyperplanes, which was trained by SVMs using positive examples (+1) and negative examples (−1). The brief summary of image and information retrieval is given in Table 8.

### 3.6. *Other applications*

There are many more other applications of SVMs for pattern recognition problems. Moghaddam and Yang[71,110] used nonlinear SVM implemented by SVMFu software to classify gender on FERET face database with 1496 training images (793 males and 713 females) and 259 test images (133 males and 126 females). Then they trained and tested each classifier with the face images using five-fold cross-validation. The performance of SVM (3.4% error rate) was shown to be superior to traditional pattern classifiers (linear, quadratic, FLD, RBF, ensemble-RBF). They experimented from $21 \times 12$ low resolution images to $84 \times 48$ high reso-

Table 8. Summary of information and image retrieval by SVMs.

| Category | Database | Input | Recognition Rate | Remarks |
|---|---|---|---|---|
| Image retrieval [34] | Bordatz texture database | Mean and variance of 24 garbor filter banks (3 scales, 4 orientations) | 87.61% retrieval performance in top 5 images | GRBF kernel (sigma = 0.3, C = 200) |
| Information retrieval with relevance feedback[21] | Reuters corpus of news articles | TF-IDF<br>TF<br>TF-IDF<br>TF | 100% (Topic: Earn)<br>100% (Topic: Earn)<br>95% (Topic: Grain)<br>87% (Topic: Grain)<br>on 10 iterations | SVMLight |
| Image retrieval with relevance feedback[41,100] | Correl database | Color, texture, structure | 90% (Category 5) in top 20 | Linear kernel |
| Image retrieval with relevance feedback[115] | Correl database | Autocorrelogram of $4 \times 4 \times 4$ quantized, R,G,B color images | 0.75 recall after 5 iterations | Gaussian Kernel |

lution images with various different kernels. From the experiments, female gave more errors because they have less significant features.[71] Also gender classification is done by gait data analysis using a SVM. Human body is segmented by adaptive background elimination and the body is divided into seven parts. Ellipse was fitted to these seven regions and centroid, aspect ratio of major and minor axes of the ellipse, the orientation of major axis of the ellipse are the feature vectors. They experimented with the best six features selected using ANOVA out of full 57 features. They experimented under the random-sequence and the random-person test and showed that the linear kernel performed at least as well as the polynomial and Gaussian kernels.[60] Walawalkar *et al.*[103] performed gender classification using visual and audio cues. The feature vectors of the visual cue were (1) $20 \times 20$ whole images of recognition rate with 95.31% and (2) top 50 PCs with recognition rate of 90.94% implemented by SVMLight software using Gaussian RBF kernel. Their own data was used for their experiments with 1640 images (883 males and 757 females). The feature vectors of the audio cues was Cepstral feature with recognition rate of 100% on ISOLET Speech Corpus data with total of 447 utterances (255 males and 192 females).

Gutta *et al.*[36] have applied SVMs to face pose classification on FERET database and their results yielded 100% accuracy. Also Huang *et al.*[43] applied SVMs to classify face poses into three categories. Fingerprint type classification algorithms based on SVMs into five classes were proposed by Yao *et al.*[112] SVMs were trained on combining flat and structured representation and showed good performance and promising approach for fingerprint classification. Also, SVM is used to recognize intrusion detection and trained with 41 features to classify attack and normal patterns. The reason why SVM is used is the speed for real time performance and scalability: SVMs are relatively insensitive to the number of data points and the classification complexity does not depend on the dimensionality of the feature space. RBF kernel and SVM light are used.[72]

SVM for texture classification is designed to receive the raw gray-value pixels instead of feature extraction. This paper is not needed for a carefully designed feature extraction because the feature extraction is reduced to the problem of training the SVMs, and SVM has the capability of learning in high-dimensional spaces. For multi-class classification, one-to-others SVM is used with Neural Network arbitrator for the final decision. The experiments are done on Brodatz and MIT Vision Texture (VisTex) database with different kernels, fifth order polynomial, Gaussian, and Tangent Hyperbolic kernels.[55] SVM is also used to solve text detection and categorization problem.[52,54]

The aim of many nonlinear forecasting methods[23,27,69,96] is to predict next points of time series. Tay and Cao[96] proposed $C$-ascending SVMs by increasing the value of $C$, the relative importance of the empirical risk with respect to the growth of regularization term. This idea is based on the assumption that it is better to give more weights on recent data than distant data. Their results showed that $C$-ascending SVMs gave better performance than standard SVM in financial time series forecasting. Fan and Palaniswami[23] had adopted SVM approach to the problem of predicting corporate distress from financial statements. For this problem, the choice of input variables (financial indicators) affects the performance of the system. This paper had suggested selecting suitable input variables that maximize the distance of vectors between different classes, and minimize the distance within the same class. Euclidean distance-based input selection provided a choice of variables that tends to discriminate within the SVM kernel used.

In addition, SVM had been applied to many other applications such as data condensation,[70] goal detection,[4] and bullet-hole image classification.[109] Data condensation[70] was to select a small subset from huge databases and the accuracy of a classifier trained on such reduced data set were comparable to results from training with the entire data sets. The paper extracted data points lying close to the class boundaries, SVs, which form a much reduced but critical set for classification using SVM. But the problem of large memory requirements for training SVM's in batch mode was solved so that the training would preserve only the SVs at each incremental step, and add them to the training set for the next step, called incremental learning. Goal detection for a particular event, ghost goals, using SVMs was proposed by Ancona *et al.*[4] Xie *et al.*[109] focused on the application of SVM for classification of bullet hole images in an auto-scoring system. The image was classified into one, two or more bullet-hole images by multi-class SVMs. Other applications are — white blood cells classification,[78] spam categorization,[42] cloud and Typhoon classification,[8,56] and soon.[31,39] There will be some more pattern recognition applications of SVMs which are not listed in this paper.

## 4. Performance Issues

The performance of SVMs largely depends on the choice of kernels. SVMs have only one user-specified parameter $C$, which controls the error penalty when the kernel is fixed, but the choice of kernel functions, which are well suited to the specific problem is very difficult.[16] Smola *et al.*[94] explained the relation between the SVM kernel method and the standard regularization theory. However, there are no theories concerning how to choose good kernel functions in a data-dependent way.[3] Amari and Wu[3] proposed a modified kernel to improve the performance of SVMs classifier. It is based on information-geometric consideration of the structure of the Riemannian geometry induced by the kernel. The idea is to enlarge the spatial resolution around the boundary by a conformal transformation so that the separability of classes is increased.

Speed and size is another problem of SVM both in training and testing. In terms of running time, SVM is slower than other neural networks for a similar generalization performance.[37] Training for very large datasets with millions of SVs is an unsolved problem.[16] Recently, even though Platt[81] and Keerthi *et al.*[50] proposed SMO (Sequential Minimization Optimization) and modified SMO to solve the training problem, it is still an open problem to improve. The issue of how to control the selection of SVs is another difficult problem, particularly when the patterns to be classified are nonseparable and the training data are noisy. In general, attempts to remove known errors from the data before training or to remove them from the expansion after training will not give the same optimal hyperplane because the errors are needed for penalizing nonseparability.[37] Lastly, although some researches have been done on training a multi-class SVM, the work for multi-class SVM classifiers is an area for further research.[16]

## 5. Conclusions

We have presented a brief introduction on SVMs and several applications of SVMs in pattern recognition problems. SVMs have been successfully applied to a number of applications ranging from face detection and recognition, object detection and recognition, handwritten character and digit recognition, speaker and speech recognition, information and image retrieval, prediction, etc. because they have yielded excellent generalization performance on many statistical problems without any prior knowledge and when the dimension of input space is very high. In this paper, we tried to summarize the comparison of the performance results for the same application as much as possible.

Some researches compared the performance of different kinds of SVM kernels to solve their problems and most results showed that RBF kernel was usually better than linear or polynomial kernels. RBF kernel performs usually better than others for several reasons such as (1) it has better boundary response as it allows extrapolation and (2) most high dimensional data sets can be approximated by Gaussian-like distributions similar to that used by RBFs.[43]

Among the application areas, the most popular research fields to apply SVMs are for face detection, verification and recognition. SVMs are binary class classifiers and it was first applied for verification or two-class classification problems. But SVMs had been used for multi-class classification problems since one-to-others and pairwise bottom-up, DDAG top-down multi-class classification methods were developed.

Most applications using SVMs showed SVMs-based problem solving outperformed other methods. Although SVMs do not have long histories, it has been applied to a wide range of machine learning tasks and used to generate many possible learning architectures through an appropriate choice of kernels. If some limitations related with the choice of kernels, training speed and size are solved, it can be applied to more real-life classification problems.

## Acknowledgments

This research was supported by the Brain Neuroinfomatics Research Program and the Creative Research Initiative Program of the Ministry of Science and Technology, Korea.

## References


1. H. Ai, L. Liang and G. Xu, "Face detection based on template matching and support vector machines," *Proc. IEEE Int. Conf. Image Processing*, 2001, pp. 1006–1009.
2. H. Ai, L. Ying and G. Xu, "A subspace approach to face detection with support vector machines," *Proc. 16th Int. Conf. Pattern Recognition*, Quebec City, Canada, 2002, pp. 45–48.
3. S. Amari and S. Wu, "Improving support vector machine classifiers by modifying kernel functions," *Proc. Int. Conf. Neural Networks*, 1999, pp. 783–789.
4. N. Ancona, G. Cicirelli, A. Branca and A. Distante, "Goal detection in football by using support vector machines for classification," *Proc. IEEE Int. Joint Conf. Neural Networks*, Vol. 1, 2001, pp. 611–616.
5. N. Ancona, G. Cicirelli, E. Stella and A. Distante, "Object detection in images: runtime complexity and parameter selection of support vector machines," *Proc. 16th Int. Conf. Pattern Recognition*, Quebec City, Canada, 2002, pp. 426–429.
6. S. Antani, U. Gargi, D. Crandall, T. Gandhi and R. Kasturi, "Extraction of text in video," Dept. of Computer Science and Eng., Pennsylvania Stat Univ., Technical Report, CSE-99-016, 1999.
7. E. Ardizzone, A. Chella and R. Pirrone, "Pose classification using support vector machines," *Proc. IEEE Int. Joint Conf. Neural Networks*, Vol. 6, 2000, pp. 317–322.
8. M. R. Azimi-Sadjadi and S. A. Zekavat, "Cloud classification using support vector machines," *Proc. IEEE on Geoscience and Remote Sensing Symp.*, Vol. 2, 2000, pp. 669–671.
9. C. Bahlmann, B. Hassdonk and H. Burkhardt, "On-line handwriting recognition with support vector machines — a kernel approach," *Proc. 8th Int. Workshop on Frontiers in Handwriting Recognition*, Ontario, Canada, 2002, pp. 49–54.
10. N. Bassiou, C. Kotropoulos, T. Kosmidis and I. Pitas, "Frontal face detection using support vector machines and backpropagation neural networks," *Proc. Int. Conf. Image Processing*, 2001, pp. 1026–1029.
11. S. Bengio and J. Mariethoz, "Learning the decision function for speaker verification," *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Vol. 1, 2001, pp. 425–428.
12. S. M. Bileschi and B. Heisele, "Advances in component-based face detection," *Pattern Recognition with Support Vector Machines*, LNCS 2388, eds. S.-W. Lee and A. Verri, Springer, 2002, pp. 135–143.
13. B. Boser, I. Guyon and V. Vapnik, "A training alogorithm for optimal margin classifiers," *Proc. 5th Annual Workshop on Computational Learning Theory*, NY, 1992, pp. 144–152.
14. I. Buciu, C. Kotropoulos and I. Pitas, "Combining support vector machines for accurate face detection," *Proc. Int. Conf. Image Processing*, Vol. I, 2001, pp. 1054–1057.
15. I. Buciu, C. Kotropoulos and I. Pitas, "On the stability of support vector machines for face detection," *Proc. Int. Conf. Image Processing*, Vol. III, 2002, pp. 121–124.
16. C. C. Burges, "A tutorial on support vector machines for pattern recognition," *Proc. Int. Conf. Data Mining and Knowledge Discovery*, Vol. 2, 1998, pp. 121–167.
17. C. Campbell, "An introduction to kernel methods," *Radial Basis Function Networks: Design and Applications*, eds. R. J. Howlett and L. C. Jain, Springer-Verlag, 2000.
18. C. Choisy and A. Belaid, "Handwriting recognition using local methods for normalization and global methods for recognition," *Proc. 6th Int. Conf. Document Analysis and Recognition*, 2001, pp. 23–27.

19. C. Cortes and V. Vapnik, "Support vector networks," *Mach. Learn.* **20** (1995) 273–297.
20. H. Drucker, D. Wu and V. Vapnik, "Support vector machines for spam categorization," *IEEE Trans. Neural Networks* **10**, 5 (1999) 1048–1054.
21. H. Drucker, B. Shahrary and D. C. Gibbon, "Support vector machines: relevance feedback and information retrieval," *Inform. Process. Manag.* **38**, 3 (2002) 305–323.
22. X. Dong and W. Zhaohui, "Speaker recognition using continuous density support vector machines," *Electron. Lett.* **37**, 17 (2001) 1099–1101.
23. A. Fan and M. Palaniswami, "Selecting bankruptcy predictors using a support vector machine approach," *Proc. IEEE Int. Joint Conf. Neural Networks*, Vol. 6, 2000, pp. 354–359.
24. B. Froba and C. Kublbeck, "Robust face detection at video frame rate based on edge orientation features," *Proc. 5th IEEE Int. Conf. Automatic Face and Gesture Recognition*, Washington DC, 2002, pp. 327–332.
25. T. Frontzek, T. Navin Lal and R. Eckmiller, "Predicting the nonlinear dynamics of biological neurons using support vector machines with different kernels," *Proc. IEEE Int. Joint Conf. Neural Networks*, Vol. 2, 2001, pp. 1492–1497.
26. D. Gao, J. Zhou and L. Xin, "SVM-based detection of moving vehicles for automatic traffic monitoring," *Proc. IEEE Int. Conf. Intelligent Transportation System*, 2001, pp. 745–749.
27. T. V. Gestel, J. A. K. Suykens, D. E. Baestaens, A. Lambrechts, G. Lanckriet, B. Vandaele, B. De Moor and J. Vandewalle, "Financial time series prediction using least squares support vector machines within the evidence framework," *IEEE Trans. Neural Networks* **12**, 4 (2001) 809–821.
28. M. Gordan, C. Kotropoulos and I. Pitas, "Application of support vector machines classifiers to visual speech recognition," *Proc. IEEE Int. Conf. Image Processing*, Vol. III, 2002, pp. 129–132.
29. M. Gordan, C. Kotropoulos and I. Pitas, "Visual speech recognition using support vector machines," *Proc. Digital Signal Processing*, 2002, pp. 1093–1096.
30. D. Gorgevik, D. Cakmakov and V. Radevski, "Handwritten digit recognition by combining support vector machines using rule-based reasoning," *Proc. 23rd Int. Conf. Information Technology Interfaces*, 2001, pp. 139–144.
31. A. Gretton, M. Davy, A. Doucet and P. J. W. Rayner, "Nonstationary signal classification using support vector machines," *Proc. 11th IEEE Workshop on Statistical Signal Processing*, 2001, pp. 305–308.
32. G. Guo, S. Z. Li and K. L. Chan, "Face recognition by support vector machines," *Proc. 4th IEEE Int. Conf. Automatic Face and Gesture Recognition*, Grenoble, France, 2000, pp. 196–201.
33. G. Guo, S. Z. Li and K. L.Chan, "Support vector machines for face recognition," *J. Imag Vis. Comput.* **19**, 9 & 10 (2001) 631–638.
34. G. Guo, H. J. Zhang and S. Z. Li, "Distance-from-boundary as a metric for texture image," *Proc. Int. Conf. Acoustics, Speech, and Signal Processing*, 2001, pp. 1629–1632.
35. B. Gutschoven and P. Verlinde, "Multi-modal identity verification using support vector machines (SVM)," *Proc. 3rd Int. Conf. Information Fusion*, 2000, pp. 3–8.
36. S. Gutta, J. R. J. Huang, P. Jonathon and H. Wechsler, "Mixture of experts for classification of gender, ethnic origin, and pose of human," *IEEE Trans. Neural Networks* **11**, 4 (2000) 948–960.
37. S. Haykin, *Neural Networks*, Prentice Hall Inc., 1999.

38. B. Heisele, P. Ho and T. Poggio, "Face recognition with support vector machines: global versus component-based approach," *Proc. 8th IEEE Int. Conf. Computer Vision*, Vol. 2, 2001, pp. 688–694.
39. S. I. Hill, P. J. Wolfe and P. J. W. Rayner, "Nonlinear perceptual audio filtering using support vector machines," *Proc. 11th IEEE Int. Workshop on Statistical Signal Processing*, 2001, pp. 305–308.
40. E. Hjelams and B. K. Low, "Face detection: a survey," *Comput. Vis. Imag. Underst.* **83**, 3 (2001) 236–274.
41. P. Hong, Q. Tian and T. S. Huang, "Incorporate support vector machines to content-based image retrieval with relevance feedback," *Proc. IEEE Int. Conf. Image Processing*, Vol. 3, 2000, pp. 750–753.
42. C. W. Hsu and C. J. Lin, "A comparison of methods for multiclass support vector machines," *IEEE Trans. Neural Networks* **13**, 2 (2002) 415–425.
43. J. Huang, X. Shao and H. Wechsler, "Face pose discrimination using support vector machines (SVM)," *Proc. IEEE Int. Conf. Image Processing*, 1998, pp. 154–156.
44. J. Huang, V. Blanz and B. Heisele, "Face recognition using component-based SVM classification and morphable models," *Pattern Recognition with Support Vector Machines*, LNCS 2388, eds. S.-W. Lee and A. Verri, Springer, 2002, pp. 334–341.
45. A. K. Jain and B. Yu, "Automatic text location in images and video frames," *Patt. Recogn.* **31**, 12 (1998) 2055–2976.
46. I. Jang, B. C. Ko and H. Byun, "Automatic text extraction in news images using morphology," *Proc. SPIE Visual Communication and Image Processing*, 2002, pp. 4671–4648.
47. T. Joachims, "Text categorization with support vector machines: learning with many relevant features," *Proc. 10th European Conf. Machine learning*, 1999, pp. 137–142.
48. K. Jonsson, J. Kittler and Y. P. Matas, "Support vector machines for face authentication," *J. Imag. Vis. Comput.* **20**, 5 & 6 (2002) 369–375.
49. S. Kang, H. Byun and S.-W. Lee, "Real-time pedestrian detection using support vector machines," *Pattern Recognition with Support Vector Machines*, LNCS 2388, eds. S.-W. Lee and A. Verri, Springer, 2002, pp. 268–277.
50. S. S. Keerthi, S. K. Shevade, C. Bhattacharyya and K. R. K. Murthy, "Improvements to platt's SMO algorithm for SVM classifier design," Technical Report, Dept. of CSA, IISc, Bangalore, India, 1999.
51. H. C. Kim, S. Pang, H. M. Je, D. Kim and S. Y. Bang, "Pattern classification using support vector machine ensemble," *Proc. 16th Int. Conf. Pattern Recognition*, Quebec City, Canada, 2002, pp. 160–163.
52. K. I. Kim, K. Jung, S. H. Park and H. J. Kim, "Supervised texture segmentation using support vector machines," *Electron. Lett.* **35**, 22 (1999) 1935–1937.
53. K. I. Kim, J. Kim and K. Jung, "Recognition of facial images using support vector machines," *Proc. 11th IEEE Workshop on Statistical Signal Processing*, 2001, pp. 468–471.
54. K. I. Kim, K. Jung, S. H. Park and H.J. Kim, "Support vector machine-based text detection in digital video," *Patt. Recogn.* **34**, 2 (2001) 527–529.
55. K. I. Kim, K. Jung, S. H. Park and H. J. Kim, "Support vector machines for texture classification," *IEEE Trans. Patt. Anal. Mach. Intell.* **24**, 11 (2002) 1542–1550.
56. A. Kitamoto, "Typhoon analysis and data mining with kernel methods," *Pattern Recognition with Support Vector Machines*, LNCS 2388, eds. S.-W. Lee and A. Verri, Springer, 2002, pp. 237–248.

57. J. Kivinen, M. Warmuth and P. Auer, "The perceptron algorithm versus winnow: linear versus logarithmic mistake bounds when few input variables are relevant (Technical Note)," *Artif. Intell.* **97**, 1 & 2 (1997) 325–343.
58. V. Kumar and T. Poggio, "Learning-based approach to real time tracking and analysis of faces," *Proc. 4th IEEE Int. Conf. Auomatic Face and Gesture Recognition*, Grenble, France, 2000, pp. 96–101.
59. T. Kurita and T. Taguchi," A modification of kernel-based Fisher discriminant analysis for face detection," *Proc. 5th IEEE Int. Conf. Automatic Face and Gesture Recognition*, Washington DC, 2002, pp. 285–290.
60. L. Lee and W. E. L. Grimson, "Gait analysis for recognition and classification," *Proc. IEEE Int. Conf. Atomatic Face and Gesture Recognition*, Washington DC, 2002, pp. 148–155.
61. Y. Li, S. Gong, J. Sherrah and H. Liddell, "Multi-view face detection using support vector machines and eigenspace modelling," *Proc. 4th Int. Conf. Knowledge-Based Intelligent Engineering Systems & Allied Technologies*, 2000, pp. 241–244.
62. Y. Li, S. Gong and H. Liddell, "Support vector regression and classification based multi-view face detection and recognition," *Proc. 4th IEEE Int. Conf. Automatic Face and Gesture Recogntion*, Genoble, France, 2000, pp. 300–305.
63. Z. Li, Z. Weida and J. Licheng, "Radar target recognition based on support vector machine," *Proc. 5th Int. Conf. Signal Processing*, Vol. 3, 2000, pp. 1453–1456.
64. Z. Li, S. Tang and S. Yan, "Multi-class SVM classifier based on pairwise coupling," *Pattern Recognition with Support Vector Machines*, LNCS 2388, eds. S.-W. Lee and A. Verri, Springer, 2002, pp. 321–333.
65. J. Lu, K. N. Plataniotis and A. N. Ventesanopoulos, "Face recognition using feature optimization and v-support vector machine," *Proc. IEEE Neural Networks for Signal Processing XI*, 2001, pp. 373–382.
66. C. Ma, M. A. Randolph and J. Drish, "A support vector machines-based rejection technique for speech recognition," *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Vol. 1, 2001, pp. 381–384.
67. Y. Ma and X. Ding, "Face detection based on hierarchical support vector machines," *Proc. Int. Conf. Pattern Recognition*, Quebec City, Canada, 2002, pp. 222–225.
68. Y. Ma and X. Ding, "Face detection based on cost-sensitive support vector machines," *Pattern Recognition with Support Vector Machines*, LNCS 2388, eds. S.-W. Lee and A. Verri, Springer, 2002, pp. 260–267.
69. D. Mckay and C. Fyfe, "Probability prediction using support vector machines," *Proc. Int. Conf. Knowledge-Based Intelligent Engineering Systems and Allied Technologies*, Vol. 1, 2000, pp. 189–192.
70. P. Mitra, C.A. Murthy and S. K. Pal, "Data condensation in large database by incremental learning with support vector machines," *Proc. 15th Int. Conf. Pattern Recognition*, Barcelona, Spain, Vol. 2, 2000, pp. 708–711.
71. B. Moghaddam and M. H. Yang, "Learning gender with support faces," *IEEE Trans. Patt. Anal. Mach. Intell.* **24**, 5 (2002) 707–711.
72. S. Mukkamala, G. Janoski and A. Sung, "Intrusion detection using neural networks and support vector machines," *Proc. IEEE Int. Joint Conf. Neural Network*, 2002, pp. 1702–1707.
73. C. Nakajima, M. Pontil and T. Poggio, "People recognition and pose estimation in image sequences," *Proc. IEEE Int. Joint Conf. Neural Networks*, Vol. 4, 2000, pp. 189–194.

74. K. I. Naruyama, M. Maruyama, H. Miyao and Y. Nakano, "Handprinted hiragana recognition using support vector machines," *Proc. 8th Int. Workshop on Frontiers in Handwriting Recognition*, Ontario, Canada, 2002, pp. 55–60.
75. J. Ng and S. Gong, "Composite support vector machines for detection of faces across views and pose estimation," *Imag. Vis. Comput.* **20**, 5 & 6 (2002) 359–368.
76. J. Ng and S. Gong, "Performing multi-view face detection and pose estimation using a composite support vector machine across the view sphere," *Proc. IEEE Int. Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*, 1999, pp. 26–27.
77. E. Osuna, R. Freund and F. Girosi, "Training support machines: an application to face detection," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1997, pp. 130–136.
78. C. Ongun, U. Halici, K. Leblebicioglu, V. Atalay, M. Beksac and S. Beksac, "Feature extraction and classification of blood cells for an automated differential blood count system," *Proc. IEEE Int. Joint Conf. Neural Networks*, 2001, pp. 2461–2466.
79. C. P. Papageorgiou, M. Oren and T. Poggio, "A general framework for object detection," *Proc. Int. Conf. Computer Vision*, 1998, pp. 555–562.
80. M. Pittore, C. Basso and A. Verri, "Representing and recognizing visual dynamic events with support vector machines," *Proc. Int. Conf. Image Analysis and Processing*, 1999, pp. 18–23.
81. J. Platt, "Sequential minimal optimization: a fast algorithm for training support vector machines," Microsoft Research Technical Report MSR-TR-98-14, 1998.
82. J. Platt, N. Christianini and J. Shawe-Taylor, "Large margin DAGs for multiclass classification," *Adv. Neural Inform. Process. Syst.* **12**, (2000) 547–553.
83. M. Pontil and A. Verri, " Support vector machines for 3D object recognition," *IEEE Trans. Patt. Anal. Mach. Intel.* **20**, 6 (1998) 637–646.
84. Y. Qi, D. Doermann and D. DeMenthon, "Hybrid independent component analysis and support vector machine," *Proc. Int. Conf. Acoustics, Speech and Signal Processing*, Vol. 3, 2001, pp. 1481–1484.
85. L. Ramirez, W. Pedrycz and N. Pizzi, "Severe storm cell classification using support vector machines and radial basis approaches," *Proc. Canadian Conf. Electrical and Computer Engineering*, Vol. 1, 2001, pp. 87–91.
86. M. S. Richman, T. W. Parks and H. C. Lee, "A novel support vector machine-based face detection method," *Proc. Record 33rd Asilomar on Signals, Systems, and Computers*, 1999, pp. 740–744.
87. S. Romdhani, B. Schokopf and A. Blake, "Computationally efficient face dectection," *Proc. Int. Conf. Computer Vision*, 2001, pp. 695–700.
88. H. A. Rowley, S. Baluja and T. Kanade, "Neural network-based face detection," *IEEE Trans. Patt. Anal. Mach. Intell.* **20**, 1 (1998) 23–38.
89. D. Roobaert and M. M. Van Hulle, "View-based 3D object recognition with support vector machines," *Proc. IX IEEE Workshop on Neural Networks for Signal Processing*, 1999, pp. 77–84.
90. E. M. Santos and H. M. Gomes, "Appearance-based object recognition using support vector machines," *Proc. XIV Brazilian Symp. Computer Graphics and Image Processing*, 2001, 399 pp.
91. B. Schölkopf and C. Burges, *Advances in Kernel Methods: Support Vector Machines*, MIT Press, Cambridge, MA, Dec. 1998.
92. F. Seraldi and J. Bigun, "Retinal vision applied to facial features detection and face authentication," *Patt. Recogn. Lett.* **23**, 4 (2002) 463–475.

93. C. S. Shin, K. I. Kim, M. H. Park and H. J. Kim, "Support vector machine-based text detection in digital video," *Proc. IEEE Workshop on Neural Networks for Signal Processing*, Vol. 2, 2002, pp. 634–641.
94. A. J. Smola, B. Schölkopf and K. R. Muller, "The connection between regularization operators and support vector kernels," *Neural Networks* **11**, 4 (1998) 637–649.
95. A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *NeuroCOLT* Technical Report TR-1998-030, Royal Holloway College, London, UK, 1998.
96. F. Tay and L. J. Cao, "Modified support vector machines in financial time series forecasting," *Neurocomputing* **48**, 1–4 (2002) 847–861.
97. A. Tefas, C. Kotropoulos and I. Pitas, "Using support vector machines to enhance the performance of elastic graph matching for frontal face authentication," *IEEE Trans. Patt. Anal. Mach. Intell.* **23**, 7 (2001) 735–746.
98. L. N. Teow and K. F. Loe, "Robust vision-based features and classification schemes for off-line handwritten digit recognition," *Patt. Recogn.* **35**, 11 (2002) 2355–2364.
99. T. J. Terrillon, M. N. Shirazi, M. Sadek, H. Fukamachi and S. Akamatsu, "Invariant face detection with support vector machines," *Proc. 15th Int. Conf. Pattern Recognition*, Barcelona, Spain, Vol. 4, 2000, pp. 210–217.
100. Q. Tian, P. Hong and T. S. Huang, "Update relevant image weights for content-based image retrieval using support vector machines," *Proc. IEEE Int. Conf. Multimedia and Expo*, Vol. 2, 2000, pp. 1199–1202.
101. V. Vapnik, *The Nature of Statistical Learning Theory*, Springer-Verlag, 1995.
102. V. Wan and W. M. Campbell, "Support vector machines for speaker verification and identification," *Proc. IEEE Workshop on Neural Networks for Signal Processing X*, Vol. 2, 2000, pp. 775–784.
103. L. Walawalkar, M. Yeasin, A. M. Narasimharmurthy and R. Sharma, "Support vector learning for gender classification using audio and visual cues: a comparison," *Pattern Recognition with Support Vector Machines*, LNCS 2388, eds. S.-W. Lee and A. Verri, Springer, 2002, pp. 144–155.
104. Y. Wang, C. S. Chua and Y. K. Ho. "Facial feature detection and face recognition from 2D and 3D images," *Patt. Recogn. Lett.* **23**, 10 (2002) 1191–1202.
105. C. Wohler and U. Krebel, "Pedestrian recognition by classification of image sequences-global approaches versus local shape-temporal processing," *Proc. Int. Conf. Image Processing*, 2000, pp. 540–544.
106. D. Xi and S.-W. Lee, "Face detection and facial feature extraction using support vector machines," *Proc. 16th Int. Conf. Pattern Recognition*, Quebec City, Canada, 2002, pp. 209–212.
107. D. Xi, I. T. Podolak and S.-W. Lee, "Facial component extraction and face recognition with support vector machines," *Proc. 5th IEEE Int. Conf. Automatic Face and Gesture Recognition*, Washington DC, 2002, pp. 76–81.
108. D. Xi and S.-W. Lee, "Face detection based on support vector machines," *Pattern Recognition with Support Vector Machines*, LNCS 2388, eds. S.-W. Lee and A. Verri, Springer, 2002, pp. 370–387.
109. W. F. Xie, D. J. Hou and Q. Song, "Bullet-hole image classification with support vector machines," *Proc. IEEE Signal Processing Workshop on Neural Networks for Signal Processing*, Vol. 1, 2000, pp. 318–327.
110. M. H. Yang and B. Moghaddam, "Gender classification using support vector machines," *Proc. IEEE Int. Conf. Image Processing*, Vol. 2, 2000, pp. 471–474.
111. M. Yang, D. J. Kriegman and N. Ahuja, "Detecting faces in images: a survey," *IEEE Trans. Patt. Anal. Mach. Intell.* **24**, 1 (2002) 34–58.

112. Y. Yao, G. L. Marcialis, M. Pontil, P. Frasconi and F. Roli, "Combining flat and structured representations for fingerprint classification with recursive neural networks and support vector machines," *Patt. Recogn.* **36**, 2 (2002) 397–406.
113. J. Zhang, Y. Zhang and T. Zhou, "Classification of hyperspectral data using support vector machine," *Proc. IEEE Int. Conf. Image Processing*, Vol. 1, 2001, pp. 882–885.
114. Y. Zhang, R. Zhao and Y. Leung, "Image classification by support vector machines," *Proc. Int. Conf. Intelligent Multimedia, Video and Speech Processing*, 2001, pp. 360–363.
115. L. Zhang, F. Lin and B. Zhang, "Support vector machine learning for image retrieval," *Proc. IEEE Int. Conf. Image Processing*, 2001, pp. 721–724.
116. B. Zhao, Y. Liu and S. W. Xia, "Support vector machines and its application in handwritten numerical recognition," *Proc. 15th Int. Conf. Pattern Recognition*, Barcelona, Spain, Vol. 2, 2000, pp. 720–723.

**Hyeran Byun** received the B.S. and M.S. degrees in Mathematics from Yonsei University, Korea. She received her Ph.D. degree in computer science from Purdue University, West Lafayette, Indiana. She was an assistant professor in Hallym University, Chooncheon, Korea from 1994–1995. Since 1995, she has been an Associate Professor of Computer Science at Yonsei University, Korea.

Her research interests are multimedia, computer vision, image and video processing, artificial intelligence and pattern recognition.

**Seong-Whan Lee** received his B.S. degree in computer science and statistics from Seoul National University, Korea, in 1984; and M.S. and Ph.D. degrees in computer science from the Korea Advanced Institute of Science and Technology in 1986 and 1989, respectively.

From February 1989 to February 1995, he was an Assistant Professor in the Department of Computer Science at Chungbuk National University, Cheongju, Korea. In March 1995, he joined the faculty of the Department of Computer Science and Engineering at Korea University, Seoul, and now he is a full Professor. Prof. Lee is also the Director of National Creative Research Initiative Center for Artificial Vision Research (CAVR) supported by the Korean Ministry of Science and Technology and the visiting professor of the Artificial Intelligence Laboratory at MIT. Prof. Lee was the winner of the Annual Best Paper Award of the Korea Information Science Society in 1986. He obtained the Outstanding Young Researcher Paper Award at the 2nd International Conference on Document Analysis and Recognition in 1993, and the First Distinguished Research Professor Award from Chungbuk National University in 1994. He obtained the Outstanding Research Award from the Korea Information Science Society in 1996. He also received an Honorable Mention at the Annual Pattern Recognition Society Award for an outstanding contribution to the *Pattern Recognition Journal* in 1998. He is a fellow of International Association for Pattern Recognition, a senior member of the IEEE Computer Society and a life member of the Korea Information Science Society.

He has more than 200 publications on computer vision and pattern recognition in international journals and conference proceedings, and has authored 10 books.

His research interests include computer vision, pattern recognition and neural networks.