

Applications of Support Vector Machines for Pattern Recognition: A Survey

Hyeran Byun¹ and Seong-Whan Lee²

¹Department of Computer Science, Yonsei University
Shinchon-dong, Seodaemun-gu, Seoul 120-749, Korea
hrbyun@cs.yonsei.ac.kr

²Department of Computer Science and Engineering, Korea University
Anam-dong, Seongbuk-gu, Seoul 136-701, Korea
swlee@image.korea.ac.kr

Abstract. In this paper, we present a comprehensive survey on applications of Support Vector Machines (SVMs) for pattern recognition. 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 numerous applications.

1 Introduction

SVMs are a new type of pattern classifier based on a novel statistical learning technique that has been recently proposed by Vapnik and his co-workers [1-3]. Unlike traditional methods (e.g. 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 [4]. Since SVMs are known to generalize well even in high dimensional spaces under small training sample conditions [5] and have shown to be superior to traditional empirical risk minimization principle employed by most of neural networks [6], SVMs have been successfully applied to a number of applications ranging from face detection, verification, and recognition [5-11,26,50-56,76-80,83], object detection and recognition [12-15,24,47,57], handwritten character and digit recognition [16-18,45], text detection and categorization [19,58-61], speech and speaker verification, recognition [20-23], information and image retrieval [33-36,87], prediction [37-41] and etc.[22,27-32,41,42,53,62,64,65,74].

In this paper, we aim to give a comprehensive survey on applications of SVMs for pattern recognition. This paper is organized as follows. We give a brief explanation on SVMs in Section 2 and a detailed review of SVMs-related techniques in Section 3. Section 4 describes the limitations of SVMs. We conclude this paper in Section 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. It gives better generalization abilities (i.e. performances on unseen test data) and SRM is achieved through a minimization of the upper bound (i.e. sum of the training error rate and a term that depends on VC dimension) of the generalization error [1-3,43-45].

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 $\mathbf{x}_i \in R^d$ with a label $y_i \in \{-1, +1\}$, for all the training data $i = 1, \dots, l$, where l is the number of data, and d is the dimension of the problem. When the two classes are linearly separable in 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 represents the optimal separating hyperplane.

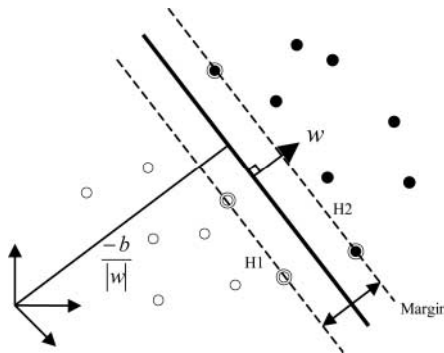


Fig. 1. Linear separating hyperplanes for the separable case. The support vectors are circled (taken from [44])

Let's 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 minimizing $\frac{1}{2} \|\mathbf{w}\|^2$

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

The optimal separating hyperplane can thus be found by minimizing equation (2) under the constraint (3) to correctly separate the training data.

$$y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1 \geq 0, \forall_i \tag{3}$$

This is a Quadratic Programming (QP) problem for which standard techniques (Lagrange Multipliers, Wolfe dual) can be used [43,51,69,70]. The detailed explanation on QP problems and alternative researches are described in Sub-section 2.4.

2.2 Linear Support Vector Machines for Non-separable 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 ξ_i in constraint (3), which then becomes

$$y_i(\mathbf{x}_i \cdot \mathbf{w} + b) \geq 1 - \xi_i, \forall_i \tag{4}$$

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

$$\min \{ \|\mathbf{w}\|^2 / 2 + C \sum_{i=1}^l \xi_i \} \tag{5}$$

where C is a parameter chosen by the user that controls the tradeoff between the margin and the misclassification errors. A larger C means that a higher penalty to misclassification errors is assigned. Minimizing equation (5) under constraint (4) gives the *Generalized Separating Hyperplane*. This still remains a QP problem. The nonseparable case is illustrated in Fig. 2.

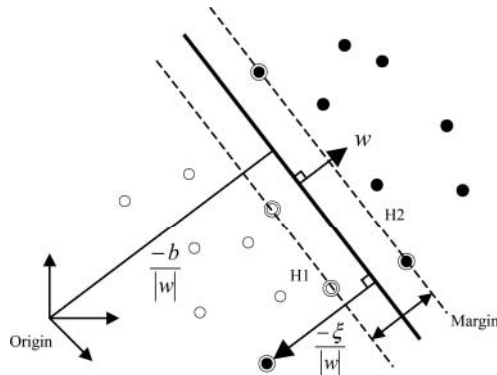


Fig. 2. Linear separating hyperplane for the non-separable case (taken from [44])

2.2.1 Nonlinear Support Vector Machines and Kernels

2.2.2 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 [41]. 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 feature space, the generalized optimal separating hyperplane shown in Fig 3 is constructed [43]. 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 [68]. This nonlinear transformation is performed in implicit way through so-called kernel functions.

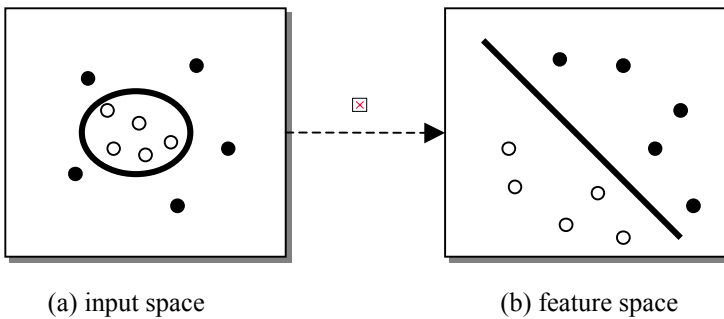


Fig. 3. Feature space is related to input space via a nonlinear map Φ , causing the decision surface to be nonlinear in the input space (taken from [33])

2.2.3 Inner-Product Kernels

In order to accomplish nonlinear decision function, an initial mapping Φ of the data into a (usually significantly higher dimensional) Euclidean space H is performed as

$\Phi: R^n \rightarrow H$, 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 number of training vectors $\Phi(\mathbf{x}_i)$ is very large, and since Φ is not known a priori, the Mercer’s theorem [44] 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 $K(\mathbf{x}_i, \mathbf{x}_j)$ and $\Phi(\mathbf{x}_i)$ does not need to be known since it is implicitly defined by the choice of kernel $K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$. 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 [75]. The use of implicit kernels allows reducing the dimension of the problem and overcoming the so-called “dimension curse” [3]. Variant learning machines are constructed according to the different kernel function $K(\mathbf{x}_i, \mathbf{x}_j)$ and thus construct different hyperplane in feature space. Table 1 shows three typical kernel functions.

Table 1. Summary of inner-product kernels [68]

Kernel function	Inner Product Kernel $K(\mathbf{x}_i, \mathbf{x}_j), i = 1, 2, \dots, N$
Polynomial kernel	$K(\mathbf{x}, \mathbf{x}_i) = (\mathbf{x}^T \mathbf{x}_i + 1)^d$
Gaussian (Radial-basis) kernel	$K(\mathbf{x}, \mathbf{x}_i) = \exp(-\ \mathbf{x} - \mathbf{x}_i\ ^2 / 2\sigma^2)$
Multi-layer perceptron (sigmoid)	$K(\mathbf{x}, \mathbf{x}_i) = \tanh(\beta_0 \mathbf{x}^T \mathbf{x}_i + \beta_1)$, β_0 and β_1 are decided by the user

2.3 Quadratic Programming Problem of SVMs

2.3.1 Dual Problem

In equation (2) and (3), the optimization goal $\Phi(\mathbf{x}_i)$ is quadratic and the constraints are linear, 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

$$(1) \sum_{i=1}^N \alpha_i d_i = 0$$

$$(2) \alpha_i \geq 0 \quad \text{for } i = 1, 2, \dots, N$$

We also may formulate the dual problem for non-separable pattern 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

$$Q(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j d_i d_j \mathbf{x}_i^T \mathbf{x}_j \quad (7)$$

subject to the constraints

$$(1) \sum_{i=1}^N \alpha_i d_i = 0$$

$$(2) 0 \leq \alpha_i \leq C \quad \text{for } i = 1, 2, \dots, N$$

where C is a user-chosen positive parameter. The objective function $Q(\alpha)$ to be maximized for the case of non-separable 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 non-separable case [68].

2.3.2 How to Solve the Quadratic Problem

A number of algorithms have been suggested for solving the dual problems. Traditional QP algorithms [71,72] are not suitable for large size problems because of the following reasons [70]:

- They require that the kernel matrix be computed and stored in memory and 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.* 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 sub-problems. As long as at least one example that violate the Karush-Kuhn-Tucker (KKT) conditions is added to the examples for the previous sub-problem, 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 sub-problems that always add at least one violator will be guaranteed to converge [51].

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 sub-problems. 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 [69]. Keerti *et al.* [73] 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.

2.4 SVMs Applied to Multi-Class Classification

The basic SVMs are for two-class problem. However it should be extended to multi-class to classify into more than two classes [45,46]. There are two basic strategies for solving q -class problems with SVMs.

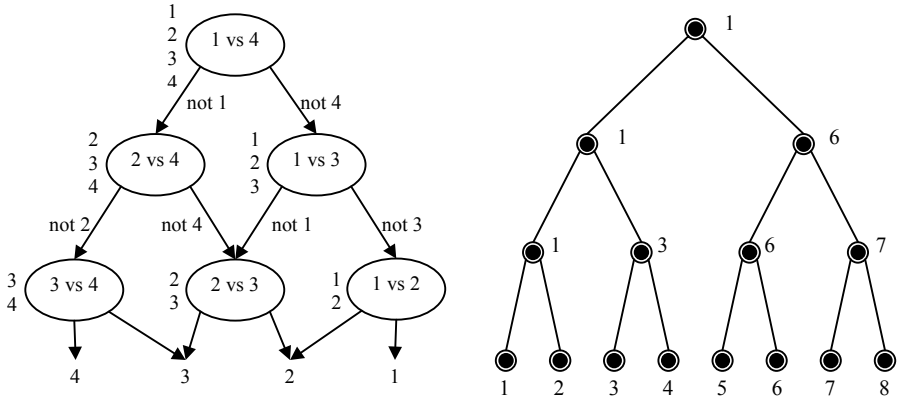
2.4.1 Multi-class SVMs: One to Others [45]

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. For the q -class problem ($q > 2$), q SVM classifiers are formed and denoted by SVM $_i$, $i=1,2,\dots, q$. As for the testing sample x , $d_i(x) = \mathbf{w}_i^* \cdot \mathbf{x} + b_i^*$ can be obtained by using SVM $_i$. The testing sample x belongs to j th class where

$$d_j(x) = \max_{i=1-q} d_i(x) \quad (8)$$

2.4.2 Multi-class SVMs: Pairwise SVMs

In the pairwise approach, q^2 machines are trained for q -class problem [47]. The pairwise classifiers are arranged in trees, where each tree node represents an SVM. A bottom-up tree, which is similar to the elimination tree used in tennis tournaments was originally proposed in [47] for recognition of 3D objects and was applied to face recognition in [9,48]. A top-down tree structure has been recently published in [49]. There is no theoretical analysis of the two strategies with respect to classification performance [10]. Regarding the training effort, the one-to-others approach is preferable since only q SVMs have to be trained compared to q^2 SVMs in the pairwise approach. However, at runtime both strategies require the evaluation of $q-1$ SVMs [10]. Recent experiments on people recognition show similar classification performances for the two strategies [24].



(a) example of top-down tree structure (b) example of bottom-up tree structure

Fig. 4. Tree structure for multi-class SVMs. (a) The decision Directed Acyclic Graph (DAG) for finding the best class out of four classes. The equivalent list state for each node is shown next to that node (taken from [49]), (b) The binary tree structure for 8 classes. For a coming test data, it is compared with each two pairs, and the winner will be tested in an upper level until the top of the tree is reached. The numbers 1-8 encode the classes (taken from [48,9])

3 Applications of SVMs for Pattern Recognition

In this Section, we survey applications of pattern recognition using SVMs. We classify existing applications into seven categories according to their aims. Some methods, which are not included in major categories, are classified into other methods and there can be more application areas which are not included in this section. Table 2 shows the summary of major SVMs-related applications

Table 2. Summary of major SVMs-related applications

Categories	Major differences	Summary of applications
Face Detection	Frontal face detection	<ul style="list-style-type: none"> - applied SVM to face detection first - suggested novel decomposition algorithm [51]
	To speed up face detection on skin segmented region	<ul style="list-style-type: none"> - face detection/eye detection [52] - ICA features as an input [83] - orthogonal Fourier-Mellin Moments as an input [11] - overcomplete wavelet decomposition as an input [76]

Face Detection	To speed up face detection	<ul style="list-style-type: none"> - eyes-in-whole and face templates as preprocessing [78] - calculated reduced support vectors [79]
	Multi-view face detection	<ul style="list-style-type: none"> - constructed separate SVMs for face detection on different views [26, 54, 80]
	Combination of multiple methods	<ul style="list-style-type: none"> - eigenface for a coarse face detection followed by an SVM for fine detection [55] - Majority voting on outputs of 5 different kernels of SVMs [77]
Face Vaerification	M2VTS database (EER=3.7%)	<ul style="list-style-type: none"> - reformulated Fisher's linear discriminant ratio to quadratic problem to apply SVM [8]
	M2VTS database (EER=1.0)	<ul style="list-style-type: none"> - showed that the performance of SVMs was relatively insensitive to the representation space(PCA, LDA) and pre-processing steps [5]
Object Recognition	ORL database (Recognition Rate 97%)	<ul style="list-style-type: none"> - bottom-up tree multi-class method - input feature for SVM was extracted by PCA [9,48]
	ORL database (Recognition Rate 98%)	<ul style="list-style-type: none"> - suggested the modified kernel to explore spatial relationships of the facial features [56]
	Own database (Recognition Rate 90%)	<ul style="list-style-type: none"> - top-down tree multi-class method - 3D range data for 3D shape features and 2D textures are projected onto PCA subspace and PC's are input to SVMs [50]
	Own database	<ul style="list-style-type: none"> - compared component-based features with global feature as an input of SVM - SVM gave better performance when component-based features were used [10]
Object Recognition	Own database (People Rec. Rate: 99.5% ; Pose Rec. Rate : 84.5%)	<ul style="list-style-type: none"> - people recognition(4 people) - pose recognition(4 poses) - compared bottom-up and top-down multi-class SVM and the results showed similar performance of two methods [24]

Object Recognition	COIL database (7200 images 72 views per each objects 100 objects)	<ul style="list-style-type: none"> - showed that SVMs gave a good performance for 3D object recognition from single view - tested on many synthesized images with noise, occlusion, and pixel shifting [47]
	Own database	<ul style="list-style-type: none"> - illustrated the potential of SVMs in terms of the number of training views per object(from 36 views to 2 views) for 3D object recognition - showed that the performance was decreased much when the number of training views were less than 18 views [15]
	Own database	<ul style="list-style-type: none"> - people detection - recognized trajectory of moving people [57]
	Own database	<ul style="list-style-type: none"> - detected moving vehicle - constructed the problem as 2-class problem by classifying moving vehicle from shadows [13]
Handwritten Character/ Digit Recognition	Own database, Character recognition	<ul style="list-style-type: none"> - radar target recognition [14] - pedestrian recognition [84]
	Own database, Character recognition	<ul style="list-style-type: none"> - used local view and global view for character recognition - local view model for input normalization - SVM, global view model for recognition [16]
Handwritten Character/ Digit Recognition	NIST database, Handwritten digit recognition (Rec. Rate:98.06%)	<ul style="list-style-type: none"> - combined structural and statistical features are input to single SVM classifier - constructed different SVM classifier for each feature and then combined 2 different SVMs by rule-based reasoning - single SVM gave better performance[17]

Handwritten Digit Recognition	NIST database, Handwritten digit recognition	<ul style="list-style-type: none"> - compared the performance according to: effect of input dimension, effect of the kernel function(Linear, Polynomial, Gaussian), comparison of different classifier (ML, MLP, SOM+LVQ, RBF, SVM), comparison of 3 types of multi-class SVM(one-to-others, pairwise, decision tree)[45]
		<ul style="list-style-type: none"> - extracted biologically plausible features - showed that their extracted features were linearly separable features by using linear SVM classifier [18]
Speaker/ Speech Recognition	Utterance verification for Speech recognition	<ul style="list-style-type: none"> - SVMs are used to accept keyword or reject non-keyword for speech recognition [22]
	Speaker verification/ recognition	<ul style="list-style-type: none"> - PolyVar telephone database is used [21] - new method for normalizing polynomial kernel to use with SVMs, YOHO database, text independent, best EER=0.34% [23]
		<ul style="list-style-type: none"> - combined Gaussian Mixture Model in SVM outputs - text independent speaker verification - best EER = 1.56% [20]
Image Retrieval	Brodatz texture database	<ul style="list-style-type: none"> - boundaries between classes were obtained by SVM [33]
	Correl image database	<ul style="list-style-type: none"> - SVMs were used to separate two classes of relevant and irrelevant images [34, 36, 87]
Prediction	Financial time series prediction	<ul style="list-style-type: none"> - C-ascending SVMs were suggested based on the assumption that it was better to give more weights on recent data than distant data [41]
	Bankruptcy prediction	<ul style="list-style-type: none"> - suggested to select suitable input variables that tends to discriminate within the SVM kernel used [40]

Other Classifications	Gender classification	<ul style="list-style-type: none"> - FERET database : 3.4% error rate - compared SVM-based method to: linear, quadratic, FLD, RBF, ensemble-RBF [27]
	Goal detection	<ul style="list-style-type: none"> - ghost goal detection [64]
	Fingerprint classification	<ul style="list-style-type: none"> - Types of fingerprints were classified into 5 classes [62]
	Data condensation	<ul style="list-style-type: none"> - extracted data points from huge databases and the accuracy of a classifier trained on this reduced sets were comparable to results from training with the entire data sets [42]
	Face pose classification	<ul style="list-style-type: none"> - on FERET database [31,18]
	Other Classifications	<ul style="list-style-type: none"> - bullet-hole classification for auto-scoring [32] - white blood cell classification [88] - spam categorization [89] - cloud classification [74] - hyperspectral data classification [28] - storm cell classification [29] - image classification [30]

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 use 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 [50]. 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, recognition and face expression recognition and compared their results with other methods. Each method used different input features, different databases, and different kernels to SVMs classifier.

Face Detection: The application of SVM in frontal face detection in image was first proposed by Osuna *et al.* [51]. The proposed algorithm scanned input images with a 19x19 window and a SVM with a 2nd-degree polynomial as kernel function is trained with a novel decomposition algorithm, which guarantees global optimality. To avoid exhaustive scanning for face detection, SVMs are used on different features of seg-

mented skin regions. Kumar and Poggio [52] 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 [83], SVMs classified the ICA features 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 image data. In Terrillon *et al.* [11], they applied SVM to invariant Orthogonal Fourier-Mellin Moments as features for binary face/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). Also to speed up the face detection, in [78], two templates : eyes-in-whole and face are used for filtering out face candidates for SVMs to classify face and non-face classes. Another method to improve the speed of the SVM algorithm, [79] found a set of reduced support vectors (RVs) which are calculated from support vectors. RVs are used to speed up the calculation sequentially.

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 [26,54,80]. Also combined methods are tried to improve the performance for face detection. In [55], they 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 followed by a fine SVM phase and could achieve an improved performance by speeding up the computation and keeping the accuracy. Buciu *et al.* [77] attempted to improve the performance of face detection by majority voting on the outputs of 5 different kernels of SVM. Papageorgio *et al.* [76] applied SVM to overcomplete wavelet representation as input data to detect faces and people and Richman *et al.* [82] applied SVM to find nose cross-section for face detection.

Face Recognition and Authentication: The recognition of 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 [8]:

- 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.* [9,48] proposed multi-class SVM with a binary tree recognition strategy for face recognition. Normalized feature extracted by PCA was the input of the SVM classifier. For face recognition, the papers used different inputs to an SVM classifier. Heisele *et al.* [10] developed a component-based method and global method for face recognition. In the component-based system they extracted facial components and combined them into a single feature vector, which is classified by SVM. The global system used SVM to recognize faces by classifying a single feature vector consisting of the gray values of the whole face image. Their results showed that component-

based method outperformed the global method. Kim *et al.* [56] modified SVM kernel to explore spatial relationships among potential eye, nose, and mouth object and compared their kernel with existing kernels. Wang *et al.* [50] proposed a face recognition algorithm based on both of 3D range and 2D gray-level facial images. 2D texture and 3D shape features are projected onto PCA subspace and then integrated 2D and 3D features are an input to SVM to recognize faces. For face authentication and recognition, Jonsson *et al.* [5] presented that SVMs extracted the relevant discriminative information from the training data and the performance of SVMs was relatively insensitive to the representation space and preprocessing steps. Tefas *et al.* [8] 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 for frontal face authentication. SVMs, which find the optimal separating hyperplane are constructed to solve the reformulated quadratic optimization problem for face authentication.

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.* [24] developed people recognition and pose estimation as a multi-class classification problem. This paper used bottom-up and top-down multi-class SVMs and the two types of SVM classifiers showed very similar performance.

3D object recognition was developed in [15] and [47]. Both of them used COIL object database, which contained 7200 images of 100 objects with 72 different views per each object. Roobaert *et al.* [15] 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. M. Pontil and A. Verri [47] 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. Pittore *et al.* [57] proposed a system that was able to detect the presence of moving people, represented the event by using an SVM for regression, and recognized trajectory of visual dynamic events from an image sequence by SVM classifier. Gao *et al.* [13] proposed a shadow and headlights elimination algorithm by considering this problem as a 2-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 [14] and pedestrian recognition [84].

3.3 Handwritten Character/Digit Recognition

Among the SVM-based applications, on the handwritten digit recognition problem, SVMs have shown to largely outperform all other learning algorithms, if one excludes the influence of domain-knowledge [15]. A major problem in handwriting recognition is the huge variability and distortions of patterns. Elastic models based on local observations and dynamic programming such as HMM are efficient to absorb this variabil-

ity. But their vision is local [16]. To combine the power of local and global characteristics, Choisy *et al.* [16] used NSPH-HMM for local view and normalization. SVM for global view is used for character recognition after normalization of NSPH-HMM. For handwritten digit recognition, SVMs are used in [17], [18] and [45]. Gorgevik *et al.* [17] used two different feature families (structural features and statistical features) for handwritten digit recognition using SVM classifier. They tested single SVM classifier applied on the both feature families as one set. Also two feature sets are forwarded to 2 different SVM classifiers and obtained results are combined by rule-based reasoning. The paper showed that single SVM classifier was better than rule-based reasoning applied to 2 individual classifiers. Teow *et al.* [18] had developed a vision-based handwritten digit recognition system, which extracts features that are biologically plausible, linearly separable and semantically clear. In their system, they showed that their extracted features were linearly separable features over a large set of training data in a highly non-linear domain by using linear SVM classifier. In [45], they showed the performance of handwritten digit recognition 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, decision tree).

3.4 Speaker/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, and etc. The well-known generative model classification approaches include Hidden Markov models (HMM) and Gaussian Mixture models (GMM) [20]. For training and testing data, there are text dependent and text independent data. Bengio *et al.* [21] and Wan *et al.* [23] used SVMs for speaker verification on different data sets. In [21], they experimented on text dependent and text independent data and replaced the classical thresholding rule with SVMs to decide accept or reject for speaker verification. Text independent tasks gave significant performance improvements. [23] proposed a new technique for normalizing the polynomial kernel to use with SVMs and tested on YOHO database. Dong *et al.* [20] reported on the development of a natural way of achieving combination of discriminative classifier and generative model classifiers by embedding GMM in SVM outputs, 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.* [22] have trained and tested SVMs classifier to the confidence measurement problem in speech recognition.

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 [33]. Guo *et al.* [33] proposed a new metric, distance-from-boundary to retrieve the texture image. The boundaries between classes are obtained by SVM. To retrieve more images relevant to the query image,

SVM classifier was used to separate two classes of relevant images and irrelevant images in [36,34,87]. Drucker *et al.* [36], Tian *et al.* [34] and Zhang *et al.* [87] proposed that SVMs automatically generated preference weights for relevant images. The weights were determined by the distance of the hyperplane, which was trained by SVMs using positive examples (+1) and negative examples (-1).

3.6 Prediction

The aim of many nonlinear forecasting methods[37,39,40,41] is to predict next points of time series. Tay and Cao [41] 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 was based on the assumption that it was 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 *et al.* [40] 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.

3.7 Other Applications

There are many more applications of SVMs for pattern recognition problems. Yang *et al.* [27] have investigated SVMs for visual gender classification with low-resolution “thumbnail” faces (21-by-12 pixels) processed from 1,755 images from the FERET face database. Then they trained and tested each classifier with the face images using five fold cross validation. The performance of SVM (3.4% error) was shown to be superior to traditional pattern classifiers (linear, quadratic, FLD, RBF, ensemble-RBF). Gutta *et al.* [31] have applied SVMs to face pose classification on FERET database and their results yielded 100% accuracy. Also Huang *et al.* [81] applied SVMs to classify into 3 different kinds of face poses. Yao *et al.* [62] proposed to classify fingerprint types into 5 different fingerprint classes. SVMs were trained on combining flat and structured representation and showed good performance and promising approach for fingerprint classification.

In addition, SVMs had been applied to many other applications such as data condensation [42], goal detection [64], and bullet-hole image classification [32]. Data condensation [42] 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 SVMs. But the problem of large memory requirements for training SVMs 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 An-

cona *et al.* [64]. Xie *et al.* [32] 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. White blood cells classification[88], spam categorization[89], text detection and categorization [85,86] and more others [63, 65] are applied SVMs. .

4 Limitations of SVM

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 [44]. Smola *et al.* [66] 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 [4]. Amari and Wu [4] 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 SVMs both in training and testing. In terms of running time, SVMs are slower than other neural networks for a similar generalization performance [68]. Training for very large datasets with millions of SVs is an unsolved problem [44]. Recently, even though Platt [69] and Keerthi *et al.* [70] 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 hyper-plane because the errors are needed for penalizing nonseparability [68].

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 [44].

5 Conclusion

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 and 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 did not compare the performance results for same application.

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[81].

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 2 class classification problems. But SVMs had been used for multi-class classification problems since one to others and pairwise bottom-up, top-down multi-class classification methods were developed.

Most of applications using SVMs showed SVMs-based problem solving outperformed to 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.

Acknowledgements

The authors would like to thank Mr. Byungchul Ko for many useful suggestions that helped to improve the presentation of the paper. This research was supported by the Brain Neuroinformatics Research Program and the Creative Research Initiative Program of the Ministry of Science and Technology, Korea.

References

1. B. Boser, I. Guyon, and V. Vapnik, A training algorithm for optimal margin classifiers, *In Proceedings of Fifth Annual Workshop on Computational Learning Theory*, New York, (1992).
2. C. Cortes and V. Vapnik, Support vector networks, *In Proceedings of Machine Learning*, vol. 20, pp. 273-297, (1995).
3. V. Vapnik, *The nature of statistical learning theory*, Springer, (1995).
4. S. Amari and S. Wu, Improving support vector machine classifiers by modifying kernel functions, *In Proceedings of International Conference on Neural Networks*, 12, pp. 783-789, (1999).
5. K. Jonsson, J. Kittler, and Y.P. Matas, Support vector machines for face authentication, *Journal of Image and Vision Computing*, vol. 20. pp. 369-375, (2002).
6. Juwei Lu, K.N. Plataniotis, and A.N. Ventesanopoulos, Face recognition using feature optimization and v-support vector machine, *IEEE Neural Networks for Signal Processing XI*, pp. 373-382, (2001).
7. F. Seraldi and J. Bigun, Retinal vision applied to facial features detection and face authentication, *Pattern Recognition Letters*, vol. 23, pp. 463-475, (2002).

8. A. Tefas, C. Kotropoulos, and I. Pitas, Using support vector machines to enhance the performance of elastic graph matching for frontal face authentication, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23. No. 7, pp. 735-746, (2001).
9. G. Guo, S. Z. Li, and K. L.Chan, Support vector machines for face recognition, *Journal of Image and Vision Computing*, vol. 19, pp. 631-638, (2001).
10. B. Heisele, P. Ho, and T. Poggio, Face Recognition with support vector machines: global versus component-based approach, *In Proceedings of Eighth IEEE Int. Conference on Computer Vision*, vol. 2, pp. 688-694, (2001).
11. T. J. Terrillon, M.N. Shirazi, M. Sadek, H. Fukamachi, and S. Akamatsu, Invariant face detection with support vector machines, *In Proceedings of 15th Int. Conference on Pattern Recognition*, vol. 4, 20, pp. 210-217, (2000).
12. E. M. Santos and H.M. Gomes, Appearance-based object recognition using support vector machines, *In Proceedings of XIV Brazilian Symposium on Computer Graphics and Image Processing*, pp. 399, (2001).
13. D. Gao, J. Zhou, and Leping Xin, SVM-based detection of moving vehicles for automatic traffic monitoring, *IEEE Intelligent Transportation System*, pp. 745-749, (2001).
14. Z. Li, Z. Weida, and J. Licheng, Radar target recognition based on support vector machine, *In Proceedings of 5th Int. Conference on Signal processing*, vol. 3, pp. 1453-1456, (2000).
15. D. Roobaert and M.M. Van Hulle, View-based 3D object recognition with support vector machines, *In Proceedings of IX IEEE Workshop on Neural Networks for Signal Processing*, pp. 77-84, (1999).
16. C. Choisy and A. Belaid, Handwriting recognition using local methods for normalization and global methods for recognition, *In Proceedings of Sixth Int. Conference On Document Analysis and Recognition*, pp. 23-27, (2001).
17. D. Gorgevik, D. Cakmakov, and V. Radevski, Handwritten digit recognition by combining support vector machines using rule-based reasoning, *In Proceedings of 23rd Int. Conference on Information Technology Interfaces*, pp. 139-144, (2001).
18. L.N. Teow and K.F. Loe, Robust vision-based features and classification schemes for off-line handwritten digit recognition, *Pattern Recognition*, January, (2002).
19. C.S. Shin, K.I. Kim, M.H. Park, and H.J. Kim, Support vector machine-based text detection in digital video, *In Proceedings of IEEE Workshop on Neural Networks for Signal Processing*, vol. 2, pp. 634-641, (2000).
K. I. Kim, K. Jung, S. H. Park, and H. J. Kim, Support vector machine-based text detection in digital video, *Pattern Recognition*, vol 34, pp. 527-529, (2001).
20. X. Dong and W. Zhaohui, Speaker recognition using continuous density support vector machines, *Electronics Letters*, vol. 37, pp. 1099-1101, (2001).
21. S. Bengio and J. Mariethoz, Learning the decision function for speaker verification, *In Proceedings of IEEE Int. Conference on Acoustics, Speech, and Signal Processing*, vol. 1, pp. 425-428, (2001).
22. C. Ma, M.A. Randolph, and J. Drish, A support vector machines-based rejection technique for speech recognition, *In Proceedings of IEEE Int. Conference on Acoustics, Speech, and Signal Processing* vol. 1, pp. 381-384, (2001).

23. V. Wan and W.M. Campbell, Support vector machines for speaker verification and identification, *In Proceedings of IEEE Workshop on Neural Networks for Signal Processing X*, vol. 2, (2000).
24. C. Nakajima, M. Pontil, and T. Poggio, People recognition and pose estimation in image sequences, , *In Proceedings of IEEE Int. Joint Conference on Neural Networks*, vol. 4, pp. 189-194, (2000).
25. E. Ardizzone, A. Chella, and R. Pirrone, Pose classification using support vector machines, , *In Proceedings of IEEE Int. Joint Conference on Neural Networks*, vol. 6, pp. 317-322, (2000).
26. J. Ng and S. Gong, Composite support vector machines for detection of faces across views and pose estimation, *Image and Vision Computing*, vol. 20, Issue 5-6, pp. 359-368, (2002).
27. M. H. Yang and B. Moghaddam, Gender classification using support vector machines, , *In Proceedings of IEEE Int. Conference on Image Processing*, vol. 2, pp. 471-474, (2000).
28. J. Zhang, Y Zhang, and T. Zhou, Classification of hyperspectral data using support vector machine, *In Proceedings of Int. Conference on Image Processing*, vol. 1, pp. 882-885, (2001).
29. L. Ramirez, W. Pedrycz, and N. Pizzi, Severe storm cell classification using support vector machines and radial basis approaches, *In Proceedings of Canadian Conference on Electrical and Computer Engineering*, vol. 1, pp. 87-91, (2001).
30. Y. Zhang, R. Zhao, and Y. Leung, Image Classification by support vector machines, *In Proceedings of Int. Conference on Intelligent Multimedia, Video and Speech Processing*, pp. 360-363, (2001).
31. 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. on Neural Networks*, vol. 11, Issue.4, pp. 948-960, (2000).
32. W.F. Xie, D.J. Hou, and Q. Song, Bullet-hole image classification with support vector machines, *In Proceedings of IEEE Signal Processing Workshop on Neural Networks for Signal Processing*, vol.1, pp. 318-327, (2000).
33. G. Guo, H.J. Zhang, and S.Z. Li, Distance-from-boundary as a metric for texture image retrieval, *In Proceedings of IEEE Int. Conference on Acoustics, Speech, and Signal Processing*, vol. 3, pp. 1629-1632, (2001).
34. Q. Tian, P. Hong, and T.S. Huang, Update relevant image weights for content-based image retrieval using support vector machines, *In Proceedings of IEEE Int. Conference on Multimedia and Expo*, vol.2, pp. 1199-1202, (2000).
35. P. Hong, Q. Tian, and T,S. Huang, Incorporate support vector machines to content-based image retrieval with relevance feedback, *In Proceedings of Int. Conference on Image Processing*, vol. 3, pp. 750-753, (2000).
36. H. Druker, B. Shahrar, and D.C. Gibbon, Support vector machines: relevance feedback and information retrieval, *Information Processing & Management*, vol. 38, Issue 3, pp. 305-323, (2002).
37. T. Van 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. On Neural Networks*, vol. 12. Issue 4, pp. 809-821, (2001).

38. T. Frontzek, T. Navin Lal, and R. Eckmiller, Predicting the nonlinear dynamics of biological neurons using support vector machines with different kernels, *In Proceedings of Int. Joint Conference on Neural Networks*, vol. 2, pp. 1492-1497, (2001).
39. D. Mckay and C. Fyfe, Probability prediction using support vector machines, *In Proceedings of Int. Conference on Knowledge-Based Intelligent Engineering Systems and Allied Technologies*, vol. 1, pp. 189-192, (2000).
40. A. Fan and M. Palaniswami, Selecting bankruptcy predictors using a support vector machine approach, vol. 6, pp. 354-359, (2000).
41. F. Tay and L.J. Cao, Modified support vector machines in financial time series forecasting, *Neurocomputing*, Sept., (2001).
42. P. Mitra, C.A. Murthy, and S.K. Pal, Data condensation in large database by incremental learning with support vector machines, *In Proceedings of 15th Int. Conference on Pattern Recognition*, vol. 2, pp. 708-711, (2000).
43. B. Gutschoven and P. Verlinde, Multi-modal identity verification using support vector machines (SVM), *In Proceedings of The third Int. Conference on Information Fusion*, pp. 3-8, (2000).
44. C. C. Burges, A tutorial on support vector machines for pattern recognition, *In Proceedings of Int. Conference on Data Mining and Knowledge Discovery*, 2(2), pp. 121-167, (1998).
45. B. Zhao, Y. Liu, and S.W. Xia, Support vector machines and its application in handwritten numerical recognition, *In Proceedings of 15th Int. Conference on Pattern Recognition*, vol. 2, pp. 720-723, (2000).
46. S. Bernhard, C.C. Burges, and A.J. Smola, Pairwise classification and support vector machines, The MIT Press, C. Massachusetts, London England, pp. 255-268, Jan. (1999).
47. M. Pontil and A. Verri. Support vector machines for 3-D object recognition, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, pp. 637-646, (1998).
48. G. Guodong, S. Li, and C. Kapluk, Face recognition by support vector machines. *In Proceedings of IEEE Int. Conference on Automatic Face and Gesture Recognition*, pp. 196-201, (2000).
49. J. Platt, N. Christianini, and J. Shawe-Taylor, Large margin DAGs for multiclass classification, *Advances in Neural Information Processing Systems*, (2000).
50. Y. Wang, C.S. Chua, and Y.K. Ho. Facial feature detection and face recognition from 2D and 3D images, *Pattern Recognition Letters*, Feb., (2002).
51. E. Osuna, R. Freund, and F. Girosi, Training support machines: An application to face detection. *In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 130-136, (1997).
52. V. Kumar and T. Poggio, Learning-based approach to real time tracking and analysis of faces, *In Proceedings of IEEE Int. Conference on Automatic Face and Gesture Recognition*, (2000).
53. E. Hjelams and B. K. Low, Face Detection: A Survey, *Computer Vision and Image Understanding*, 83, pp. 236-274, (2001).

54. J. Ng and S. Gong, Performing multi-view face detection and pose estimation using a composite support vector machine across the view sphere, *In Proceedings of IEEE Int. Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*, (1999).
55. Y. Li, S. Gong, J. Sherrah, and H. Liddell, Multi-view Face Detection Using Support Vector Machines and Eigenspace Modelling, *In Proceedings of Fourth Int. Conference on Knowledge-Based Intelligent Engineering Systems & Allied Technologies*, pp. 241-244, (2000).
56. K.I. Kim, J. Kim, and K Jung, Recognition of facial images using support vector machines, *In Proceedings of 11th IEEE Workshop on Statistical Signal Processing*, pp. 468-471, (2001).
57. M. Pittore, C. Basso, and A. Verri, Representing and recognizing visual dynamic events with support vector machines, *In Proceedings of Int. Conference on Image Analysis and Processing*, pp. 18-23, (1999).
58. A. K. Jain and B. Yu, Automatic text location in images and video frames, *Pattern Recognition*, vol. 31, No. 12, pp. 2055-2976, (1998).
59. 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).
60. I. Jang, B.C. Ko, and H. Byun, Automatic text extraction in news images using morphology, *In Proceedings of SPIE Visual Communication and Image Processing*, San Jose, Jan., (2002).
61. T. Joachims, Text categorization with support vector machines: learning with many relevant features, *In Proceedings of 10th European Conference on Machine learning*, (1999).
62. 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, *Pattern Recognition*, pp. 1-10, (2002).
63. A. Gretton, M. Davy, A. Doucet, and P. J.W. Rayner, Nonstationary signal classification using support vector machines, *In Proceedings of 11th IEEE Workshop on Statistical Signal Processing*, pp. 305-308, (2001).
64. N. Ancona, G. Cicirelli, A. Branca, and A. Distante, Goal detection in football by using support vector machines for classification, *In Proceedings of Int. Joint Conference on Neural Networks*, vol.1 pp. 611-616, (2001).
65. S. I. Hill, P. J. Wolfe, and P. J. W. Rayner, Nonlinear perceptual audio filtering using support vector machines, *In Proceedings of 11th IEEE Int. Workshop on Statistical Signal Processing*, pp. 305-308, (2001).
66. A. J. Smola, B. Scholkopf, and K. R. Müller, The connection between regularization operators and support vector kernels, *Neural Networks*, 11, pp. 637-649, (1998).
67. C. J. C. Burges, Simplified support vector decision rules, *In Proceedings of 13th Int. Conference on Machine Learning*, pp. 71-77, (1996).
68. S. Haykin, *Neural Networks*, Prentice Hall Inc. (1999).
69. J. Platt, Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines, *Microsoft Research Technical Report MSR-TR-98-14*, (1998).

70. 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).
71. B. Schölkopf and C. Burges, *Advances in Kernel Methods: Support Vector Machines*, MIT Press, Cambridge, MA, Dec. (1998).
72. A.J. Smola and B. Schölkopf, A tutorial on support vector regression, *NeuroCOLT Technical Report TR-1998-030*, Royal Holloway College, London, UK, (1998).
73. J. Kivinen, M. Warmuth, and P. Auer, The perceptron algorithm vs. winnow: Linear vs. Logarithmic mistake bounds when few input variables are relevant, *In Proceedings of Int. Conference on Computational Learning Theory*, (1995).
74. M.R. Azimi-Sadjadi and S.A. Zekavat, Cloud classification using support vector machines, *In Proceedings of IEEE Geoscience and Remote Sensing Symposium*, vol. 2, pp. 669-671, (2000).
75. C. Campbell. An introduction to kernel methods. In R.J. Howlett and L.C. Jain, editors, *Radial Basis Function Networks: Design and Applications*, page 31. Springer Verlag, Berlin, (2000).
76. C. P. Papageorgiou, M. Oren, and T. Poggio, A General framework for object detection, *In Proceedings of International Conference on Computer Vision*, pp. 555-562, (1998).
77. N. Bassiou, C. Kotropoulos, T. Kosmidis, and I. Pitas, Frontal face detection using support vector machines and back-propagation neural networks, *In Proceedings of Int. Conference on Image Processing*, pp. 1026-1029, (2001).
78. H. Ai, L. Liang, and G. Xu, Face detection based on template matching and support vector machines, *In Proceedings of Int. Conference on Image Processing*, pp. 1006-1009, (2001).
79. S. Romdhani, B. Schölkopf, and A. Blake, Computationally efficient face detection, *In Proceedings of Int. Conference Computer Vision*, pp. 695-700, (2001).
80. Y. Li, S. Gong, and H. Liddell, Support vector regression and classification based multi-view face detection and recognition, *In Proceedings of Face and Gesture Recognition*, pp. 300-305, (2000).
81. J. Huang, X. Shao, and H. Wechsler, Face pose discrimination using support vector machines(SVM), *In Proceedings of Int. Conference on Image Processing*, pp. 154-156, (1998).
82. M.S. Richman, T. W. Parks, and H. C. Lee, A novel support vector machine-based face detection method, *In Proceedings of Record of Thirty-Third Asilomar on Signals, Systems, and Computers*, pp. 740-744 (1999).
83. Y. Qi, D. Doermann, and D. DeMenthon, Hybrid independent component analysis and support vector machine, *In Proceedings of Int. Conference on Acoustics, Speech and Signal Processing*, vol. 3, pp. 1481-1484, (2001).
84. C. Wohler and U. Krebel, Pedestrian recognition by classification of image sequences-global approaches vs. local shape-temporal processing, *In Proceedings of Int. Conference on Image Processing*, pp. 540-544, (2000).
85. K. I. Kim, K. Jung, S. H. Park, and H. J. Kim, Supervised texture segmentation using support vector machines, *Electronics Letters*, Vol. 35, No. 22, pp. 1935-1937, (1999).

86. K. I. Kim, K. Jung, S. H. Park, and H.J Kim, Support vector machine-based text detection in digital video, *Pattern Recognition*, vol. 34, pp. 527-529 (2001).
87. L. Zhang, F. Lin, and B. Zhang, Support vector machine learning for image retrieval, *In Proceedings of Int. Conference on Image Processing*, pp. 721-724, (2001).
88. 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, *In Proceedings of Int. Joint Conference on Neural Networks*, pp. 2461-2466, (2001).
89. H. Drucker, D. Wu, and V. Vapnik, Support vector machines for spam categorization, *IEEE Transactions on Neural Networks*, vol. 10, No. 5, pp. 1048-1054 (1999).