Robust Support Vector Machine With Bullet Hole Image Classification

Qing Song, *Member, IEEE*, Wenjie Hu, and Wenfang Xie

***Abstract*—This paper proposes a robust support vector machine for pattern classification, which aims at solving the over-fitting problem when outliers exist in the training data set. During the robust training phase, the distance between each data point and the center of class is used to calculate the adaptive margin. The incorporation of the average techniques to the standard support vector machine (SVM) training makes the decision function less detoured by outliers, and controls the amount of regularization automatically. Experiments for the bullet hole classification problem show that the number of the support vectors is reduced, and the generalization performance is improved significantly compared to that of the standard SVM training.**

***Index Terms*—Bullet hole classification, robust support vector machine.**

I. Introduction

SUPPORT VECTOR MACHINE (SVM) was first introduced to solve the pattern classification and regression estimation problem by Vapnik and his colleagues [1]–[3]. It can be seen as a new training algorithm for the traditional polynomial, radial basis function, and multilayer perceptron classifier by defining relevant kernel functions. In this paper, we have named it the *standard SVM training algorithm.* The main idea of SVM is to derive a hyperplane by maximizing the margin between two classes. The interesting property of SVM is that it is an approximate implementation to the structure risk minimization (SRM) principal in statistical learning theory, rather than the empirical risk minimization method, in which the classification function is derived by minimizing the mean square error (MSE) over the data set. In recent years, it has been found in a significant amount of literature that SVM leads to remarkable improvements in handwritten digit recognition [1], image classification, and face detection [4]–[6], object detection [7], text categorization, and nonlinear time-series prediction [8].

It should be noted that one of main assumptions of SVM is that all samples in the training set are independent and identically distributed (i.i.d), however, in many practical engineering applications, the obtained training data is often contaminated by noise. Furthermore, some samples in the training data set are misplaced on the wrong side by accident. These are known as outliers. In this case, the standard SVM training algorithm will make the decision boundary deviate severly from the optimal hyperplane, such that, the SVM is very sensitive to noise, and especially those outliers that are close to the decision boundary. This makes the standard SVM no longer sparse, that is, the number of support vectors increases significantly due to outliers. Some techniques have been found [2], [9]–[11] to tackle the outlier problem. In [10], a central SVM method is proposed to use the class centers in building the SVM. In [11], an adaptive margin (AM-) SVM is developed based on the utilization of adaptive margins for each training pattern. However, there is no general way to use class center in margin of each training pattern to make the machine less sensitive to noise and outliers.

In this paper, we present a general method that follows the main ideas of SVM using the adaptive margin for each data point to formulate the minimization problem, which can be used for Mercer's kernel functions (see [1]). In [2], it is noted that the classification functions obtained by minimizing MSE are not sensitive to outliers in the training set. The reason that classical MSE is immune to outliers is that it is an "average" algorithm. A particular sample in the training set only contributes little to the final result. The effect of outliers can be eliminated by taking average on the samples. That is why the "average" technique is a simple yet effective tool to tackle outliers [10].

Motivated by the two considerations about the adaptive margin and "average" algorithm, we use the distance between the center of each class of the training data, and the sample point to form an adaptive margin. A new slack variable is introduced in the optimal function, which is the product of a preselected parameter and the square distance between each data point to the center of the respective class. Although we do not directly solve the optimization problem by minimizing MSE here, we do use the center of class representing the averaged information of the noisy training set to the margin. Therefore, the adaptive margin will make the SVM less sensitive to some specific samples, such as outliers.

The proposed method is applied to the classification of bullet hole images. We test both the SVM algorithms using different kernel functions and regularization parameters. The experiment results show that the number of the support vectors and the test error in robust SVM are reduced significantly when outliers exist, compared to that of the standard SVM. The obtained decision boundary detours less and is, therefore, more sparse compared to that of the standard SVM.

This paper is organized in the following manner. In Section II, we present the algorithm for the robust SVM. In Section III, we briefly introduce the auto-scoring system and the bullet hole image classification problem. The detailed experiment results are presented in Section IV, and finally, the summary is given in Section V.

Manuscript received May 23, 2001; revised December 2, 2001. This paper was recommended by Associate Editor P. Bhattacharya.

Q. Song and W. Hu are with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798 (e-mail: eqsong@ntu.edu.sg).

W. Xie was with CORETEC Incorporated, Waterloo, ON N2J 4R7, Canada.

Digital Object Identifier 10.1109/TSMCC.2002.807277

II. Robust Support Vector Machine

The robust SVM aims at solving over-fitting problem with outliers that make the two classes nonseparable. We develop the algorithm by following the procedure of derivation of the standard SVM for the linearly nonseparable case [12].

Consider the training samples

$$(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_l, y_l), \quad \mathbf{x}_i \in \mathbf{X}, \quad y_i \in \{-1, 1\}, \quad i = 1, \ldots, l. \tag{1}$$

These two classes cannot be separated without error by a hyperplane since there are misclassified patterns or measurement noises in the training samples. In this case, there is no decision function, such that, the inequalities

$$y_i f(\mathbf{x}_i) \geq +1, \quad i = 1, 2, \ldots l \tag{2}$$

hold true.

In the standard SVM training for nonseparable data set, represented by the following optimization problem

$$\begin{aligned} &\text{Minimize } \Phi(\mathbf{w}) = \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{l}\xi_i \\ &\text{subject to } y_i f(\mathbf{x}_i) \geq 1 - \xi_i \end{aligned} \tag{3}$$

where $\mathbf{w}$ is the weight of the kernel function and C is a constant for the slack variable $\{\xi_i\}_{i=1}^{l}$. One must admit some training errors to find the best tradeoff between training error and margin by choosing the appropriate value of C.

We formulate the prime problem of the robust algorithm by only minimizing the margin of the weigh $\mathbf{w}$ instead of minimizing the sum of the margin and misclassification error in the standard SVM training. The classification accuracy is sacrified to obtain the smooth decision boundary. In order to obtain a formal setting of nonseparable training data points, we introduce a new slack variable $\lambda D^2(\mathbf{x}_i, \mathbf{x}^*_{y_i})$ instead of $\{\xi_i\}_{i=1}^{l}$ in the standard SVM training

$$\begin{aligned} &\text{Minimize } \Phi(\mathbf{w}) = \frac{1}{2}\mathbf{w}^T\mathbf{w} \\ &\text{subject to } y_i f(\mathbf{x}_i) \geq 1 - \lambda D^2\left(\mathbf{x}_i, \mathbf{x}^*_{y_i}\right) \end{aligned} \tag{4}$$

where $\lambda \geq 0$ is a preselected parameter measuring the influence of averaged information, and $D^2(\mathbf{x}_i, \mathbf{x}^*_{y_i})$ represents the normalized distance between each data point and the center of the respective class in the kernel space, which is calculated by

$$\begin{aligned} D^2\left(\mathbf{x}_i, \mathbf{x}^*_{y_i}\right) &= \left|\phi(\mathbf{x}_i) - \phi\left(\mathbf{x}^*_{y_i}\right)\right|^2 / D^2_{\max} \\ &= \left[(\phi(\mathbf{x}_i)\cdot\phi(\mathbf{x}_i) - 2\phi(\mathbf{x}_i)\cdot\phi\left(\mathbf{x}^*_{y_i}\right)\right. \\ &\quad \left. + \phi\left(\mathbf{x}^*_{y_i}\right)\cdot\phi\left(\mathbf{x}^*_{y_i}\right)\right] / D^2_{\max} \\ &= \left[k(\mathbf{x}_i, \mathbf{x}_i) - 2k\left(\mathbf{x}_i, \mathbf{x}^*_{y_i}\right)\right. \\ &\quad \left. + k\left(\mathbf{x}^*_{y_i}, \mathbf{x}^*_{y_i}\right)\right] / D^2_{\max} \end{aligned} \tag{5}$$

where $\{\phi(\mathbf{x}_i)\}_{i=1}^{h}$ $(h \leq l)$ denotes a set of nonlinear transformations from the input space to the feature space, $k(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)\cdot\phi(\mathbf{x}_j)$ represents the inner-product kernel function; $D_{\max} = \max(D(\mathbf{x}_i, \mathbf{x}^*_{y_i}))$ is the maximum distance between the center and training data points of the respective class in the kernel space; and $k(\mathbf{x}_i, \mathbf{x}^*) = \phi(\mathbf{x}_i)\cdot\phi(\mathbf{x}^*)$ is the kernel function formed by the sample data and the center of the respective class in the feature space. For samples in class $+1, \phi(\mathbf{x}^*_{y_i}) = \phi(\mathbf{x}^*_{+1}) = (1/n^+)\sum_{y_j=+1}\phi(\mathbf{x}_j)$, n^+ is the number of data points in class $+1$, for samples in class $-1, \phi(\mathbf{x}^*_{y_i}) = \phi(\mathbf{x}^*_{-1}) = (1/n^-)\sum_{y_j=-1}\phi(\mathbf{x}_j)$, n^- is the number of data points in class -1.

Suppose the normalized data lives on the surface of a hypersphere in feature space F. The distance $D^2(\mathbf{x}_i, \mathbf{x}^*_{y_i})$ satisfies the following inequalities:

$$0 \leq D^2(\mathbf{x}_i, \mathbf{x}^*_{y_i}) \leq 1. \tag{6}$$

The support vectors are those particular points that satisfy the following equation:

$$y_i f(\mathbf{x}_i) = 1 - \lambda D^2(\mathbf{x}_i, \mathbf{x}^*_{y_i}). \tag{7}$$

The decision function

$$f(\mathbf{x}_i) = \mathbf{w}\cdot\phi(\mathbf{x}_i) + b \tag{8}$$

is constructed on the base of the solution of the above optimization problem.

We construct the Lagrangian function

$$\begin{aligned} J(\mathbf{w}, b, \alpha) = \frac{1}{2}\mathbf{w}^T\mathbf{w} - \sum_{i=1}^{l}\alpha_i &\left(y_i(\mathbf{w}\cdot\phi(\mathbf{x}_i) + b)\right. \\ &\left. - 1 + \lambda D^2\left(\mathbf{x}_i, \mathbf{x}^*_{y_i}\right)\right) \end{aligned} \tag{9}$$

where $\alpha = (\alpha_1, \ldots, \alpha_l)^T$ is the Lagrangian multipliers. The parameters that minimize the Lagrangian must satisfy the conditions

$$\frac{\partial J(\mathbf{w}, b, \xi, \alpha)}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{l}\alpha_i y_i \phi(\mathbf{x}_i) = \mathbf{0} \tag{10}$$

$$\frac{\partial J(\mathbf{w}, b, \xi, \alpha)}{\partial b} = -\sum_{i=1}^{l}\alpha_i y_i = 0. \tag{11}$$

From the condition, we derive

$$\mathbf{w} = \sum_{i=1}^{l}\alpha_i y_i \phi(\mathbf{x}_i) \tag{12}$$

$$\sum_{i=1}^{l}\alpha_i y_i = 0. \tag{13}$$

Substituting (12) into (9), we obtain

$$\begin{aligned} W(\alpha) = \sum_{i=1}^{l}\alpha_i &\left(1 - \lambda D^2\left(\mathbf{x}_i, \mathbf{x}^*_{y_i}\right)\right) \\ &-\frac{1}{2}\sum_{i,j=1}^{l}\alpha_i\alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j). \end{aligned} \tag{14}$$

We now state the dual problem for nonseparable patterns as

$$\begin{aligned} \text{Maximize } W(\alpha) = \sum_{i=1}^{l}\alpha_i &\left(1 - \lambda D^2\left(\mathbf{x}_i, \mathbf{x}^*_{y_i}\right)\right) \\ &-\frac{1}{2}\sum_{i,j=1}^{l}\alpha_i\alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) \end{aligned} \tag{15}$$

subject to the constraints

$$\sum_{i=1}^{l} y_i \alpha_i = 0 \tag{16}$$

$$\alpha_i \geq 0. \tag{17}$$

Comparing with the dual problem in the standard SVM, we may find that the only difference lies in the additional part $-\lambda D^2(\mathbf{x}_i, \mathbf{x}^*_{y_i})$ in the maximization functional $W(\alpha)$. For the large-scale problem, we can easily modify the available program that is designed for the standard SVM to tackle it.

We can justify the slack variable $\lambda D^2(\mathbf{x}_i, \mathbf{x}^*_{y_i})$ by taking it into account as part of the margin. For each data point, the separation margin can be thought as $1-\lambda D^2(\mathbf{x}_i, \mathbf{x}^*_{y_i})$ which is adaptive compared with the margin $1-\xi_i$ in standard SVM training, which is equally controlled by the single parameter C for every data point. Suppose a data point is an outlier that is located on the wrong side and far away from the separable hyperplane. The distance between this point and the center of the class is longer than that of the other normal point in the same class. The augmented term $\lambda D^2(\mathbf{x}_i, \mathbf{x}^*_{y_i})$ is relatively large. Therefore, the inequality in (4) is satisfied and the coefficients α_i associated with the data point should be toward zero. This data point, therefore, may not become a support vector. Thus, the number of support vectors in the proposed algorithm will be reduced and the decision boundary will be less detoured.

In contrast, for the standard SVM training, the ξ_i in this case should become large to make the following inequality equitation satisfied

$$y_i f(\mathbf{x}_i) \geq 1 - \xi_i. \tag{18}$$

However, since we try to minimize the misclassification error $\sum_{i=1}^{l} \xi_i$, this kind of data point may become the support vector. Here, we should study the effect of the regularization parameter λ.

- If $\lambda = 0$ no adaptation of the margin is performed. The robust SVM becomes the standard SVM.
- If $0 < \lambda \leq 1$, then $0 < \lambda D^2(\mathbf{x}_i, \mathbf{x}^*_{y_i}) \leq 1$. The algorithm is robust against the outliers located inside the decision region but on the right side of the decision surface. The contribution of the center of class is very small. The classification accuracy is kept almost the same as the standard SVM.
- If $\lambda > 1$, then $\lambda D^2(\mathbf{x}_i, \mathbf{x}^*_{y_i}) \geq 1$. The algorithm is robust against the outliers falling on the wrong side of the data set. The support vectors should be set by the data points that are relatively closer to the center of the class. The larger the parameter λ is, the nearer the support vectors will be to those data points toward the center. The algorithm becomes "more robust" against outliers and thus, results in smoother decision surface. However, the classification error may be increased correspondingly.

The characteristic of this algorithm is to gain robustness against outlier at cost of the classification accuracy. A good selection of parameter λ is the one that leads to compromise between the robustness and classification accuracy.

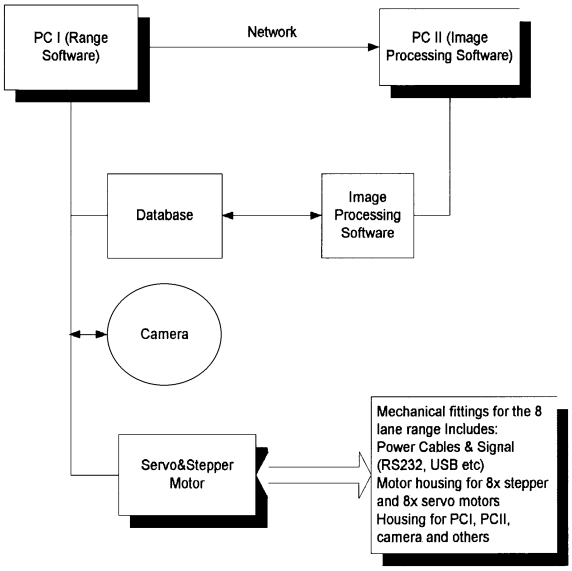

Fig. 1. Structure of auto-scoring system.

TABLE I
THE STATISTICS OF BULLET HOLES

Holes	1	2	3	4	5	6
Number	1496	108	30	4	4	0
Percentage (%)	91.9	6.6	0.9	0.2	0.2	0

III. BULLET HOLE IMAGE CLASSIFICATION

In order to test the proposed algorithm, we apply it to the classification of bullet hole images in the auto-scoring system. The target carriage in the auto-scoring system is pulled and controlled by the control processor (computer), as shown in Fig. 1. The system has a target image printed on a paper with specified scoring areas. A digital or video camera is installed at the fixed home position. After a complete shooting session, the target carriage is pulled back to the home position and the image of the target paper is captured by the camera. The obtained target images are transferred subsequently to the image processor for auto-scoring procedure.

A classical region growth algorithm is used to obtain each independent bullet hole image. Normally, certain bullet hole images contain two or more bullet holes with different degrees of overlapping. Therefore, it is critical to classify bullet hole images into multi-class sets for the auto-scoring system. The statistical information of the bullet hole images based on 35 representative samples of target papers is shown in Table I.

The main task of the auto-scoring system is to classify the bullet hole images into proper classes. To simplify the illustration, only one-hole and two-hole images are considered in this paper. In order to obtain the classifier by using SVM method, two steps should be carried out:

Step 1) design a feature extractor to extract important features;

Step 2) design a classifier for one-hole and two-hole images with good generalization ability.

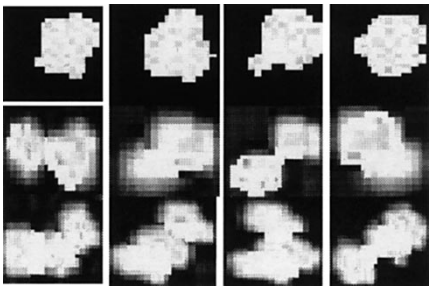

Fig. 2. Some repesentative examples of bullet hole images used in the experiments. The first row: 1-bullet holes; the second row; 2-bullet holes; the third row; 3-bullet holes.

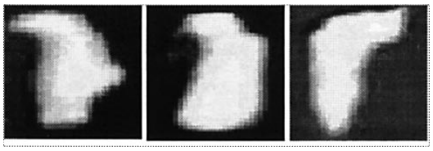

Fig. 3. One-hole bullet images behaving like a two-hole image due to the gun with malfunction.

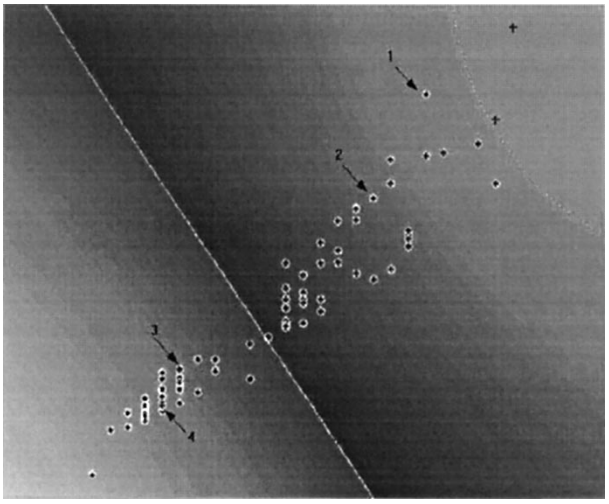

Fig. 4. Standard SVM training with Gaussian RBF kernel function (width $= 2$), $C = 0.1$ with two outliers on each wrong side divided by the boundary (dashed) line, which are the same for figures.

A. Feature Extraction

Feature extraction is the first step for pattern recognition. Fig. 2 shows some representative examples of the bullet hole imagines. The size and shape of a bullet hole are crucial features for pattern recognition. The object size is reflected in measurements of area, length, width, and perimeter. Object shape can be reflected in measurements of rectangle fit, circularity, and invariant moments. It can also be encoded in chain code, polar boundary, complex boundary function, and medial axis transform. Altogether, we compute 20 geometry-related features for each bullet hole image including some intuitive geometric features of the bullet holes and some invariant moments within the system. A detailed description of the feature extraction can be found in [13].

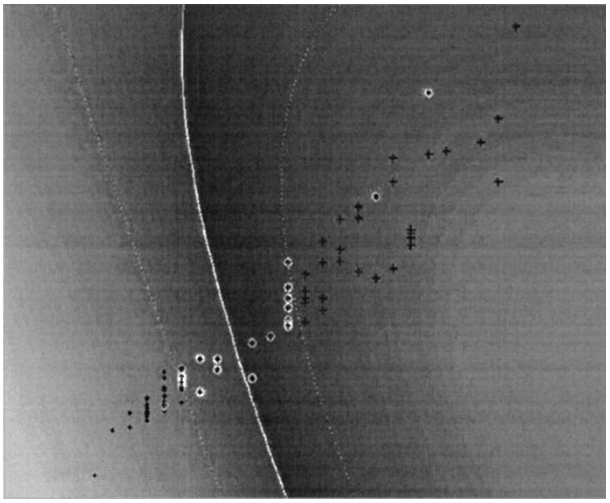

Fig. 5. Standard SVM training with Gaussian RBF kernel function (width $= 2$) $C = 10$.

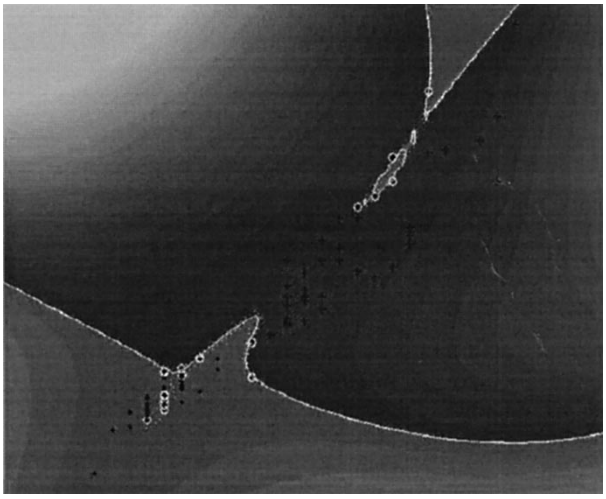

Fig. 6. Standard SVM training with Gaussian RBF kernel function (width $= 2$) $C = 1000$.

B. Design a Classifier With Robust SVM

The next step is to train the SVM to obtain the classifier based on the extracted features. It should be noted that in practical application the extracted features might contain error or noise. These are mainly caused by the significant calculation of features.

Another main factor which affects the performance of SVM is the man-made mistake during the preparation procedure of the training data set. To do this, we should not only calculate 20 features of each bullet hole image, but also provide the correct classification information (teaching signal). The information is normally provided by the experienced training supervisor in the shoot range. However, there are a lot of factors affecting the accuracy of the correct classification information. For example, the malfunction of the gun will cause the one-hole bullet image to behave like a two-hole image, as shown in Fig. 3. Sometimes, different training supervisor may classify the same bullet hole image as different classes due to individual perspective. Although these mistakes may be reduced by cross validation,

TABLE II
STANDARD SVM TRAINING WITH DIFFERENT PARAMETER C (GAUSSIAN RBF KERNEL FUNCTION WITH WIDTH $= 2$)

C	Nsv	No.M	Test Error
0.1	80	32	3.6%
1	40	23	2.6%
10	25	20	2.2%
20	20	23	2.6%
50	17	38	4.2%
100	17	43	4.8%
500	18	61	6.8%
1000	17	67	7.4%

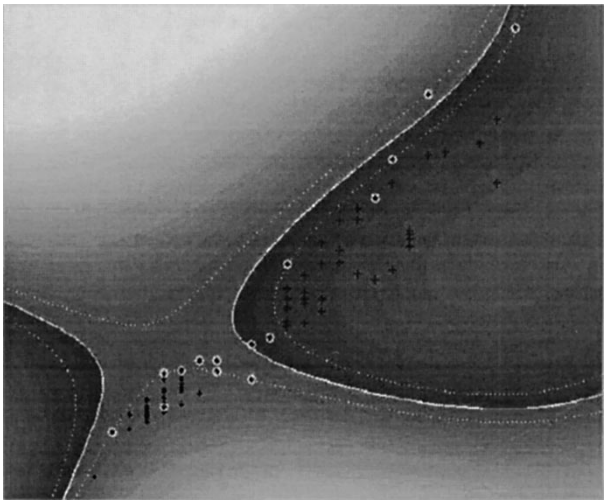

Fig. 7. Robust SVM training with Gaussian RBF kernel function (width $= 2$), $\lambda = 0.1$.

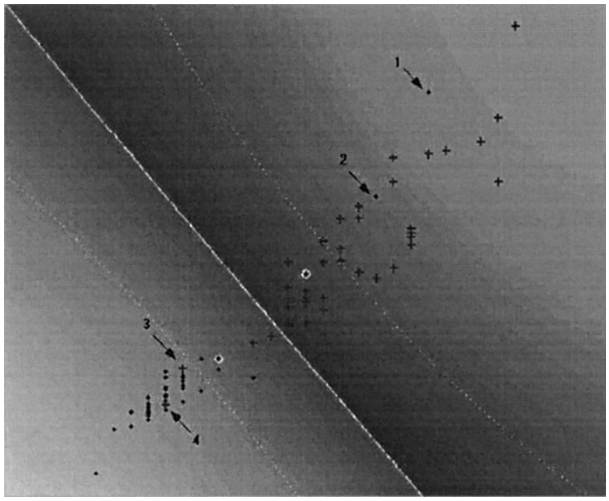

Fig. 8. Robust SVM training with Gaussian RBF kernel function (width $= 2$), $\lambda = 10$. (The four outliers are marked by arrow).

the misclassification will remain, particularly, when the training data points are in a situation of over-supply. These misclassified data points are outliers, which may cause over-fitting problem in the SVM training. The next section will show how this problem is solved by the proposed algorithm.

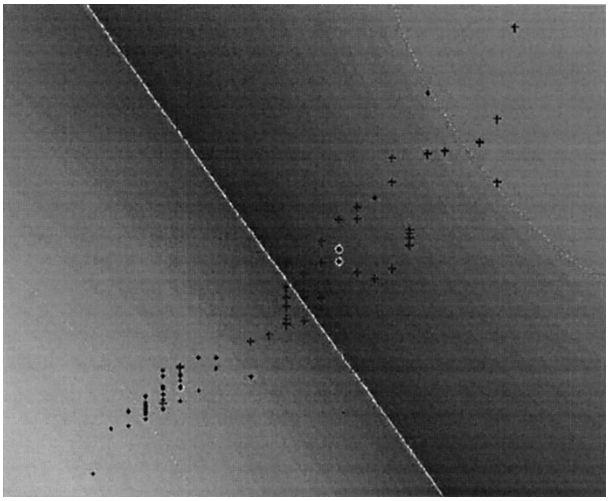

Fig. 9. Robust SVM training with Gaussian RBF kernel function (width $= 2$), $\lambda = 100$.

TABLE III
ROBUST SVM TRAINING WITH DIFFERENT PARAMETER λ (GAUSSIAN RBF KERNEL FUNCTION WITH WIDTH $= 2$)

λ	Nsv	No.M	Test Error
0.1	16	29	3.2%
1	15	27	3.0%
4	8	20	2.2%
5	4	20	2.2%
7	2	19	2.1%
10	2	19	2.1%
50	3	22	2.4%
100	4	38	4.2%

IV. EXPERIMENT RESULTS

To show effectiveness of the algorithm, 2-D features of the bullet hole image, i.e., the area and perimeter, are selected in this experiment for illustration purpose. Altogether, 41 one-hole bullet samples and 41 two-hole bullet samples are chosen as training samples. In order to show the robustness of the algorithm, we deliberately classify two one-hole images as two-hole images, and two two-hole images as one-hole images (The four outliers are marked by arrows in Fig. 4 and Fig. 8 for illustration purpose, all the following figures also have the same outliers). These four samples may be considered as outliers to the training samples. The robust SVM training program is modified based on the Matlab program written by Gunn [14]. Two kernel functions (Gaussian RBF and polynomial kernel functions) and different selections of the regularization parameters (C and λ) have been used for the training to compare performance of the robust SVM with that of the standard SVM. In addition to the representative samples in Table I, extra samples are selected from other target images, which consist of a total of 900 samples including 600 one-hole and 300 two-hole images for testing purpose. Figs. 4–6 show the results of standard Gaussian RBF SVM with penalty parameter $C = 0.1, 10, 1000$ from which 82 training samples are contaminated by four outliers. More detailed results are shown in Table II.

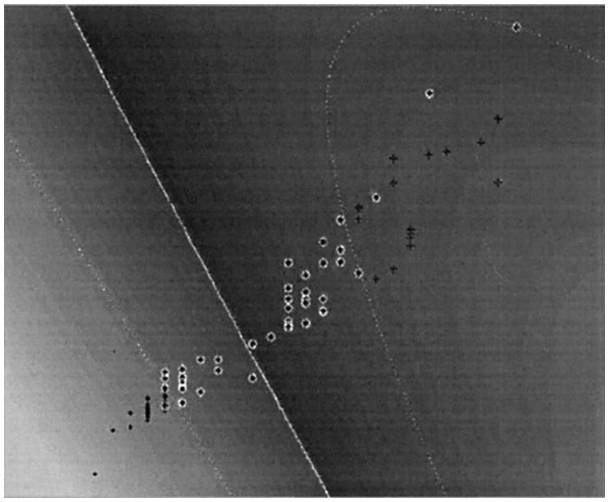

Fig. 10. Standard SVM training with polynomial kernel function (degree $= 2$) $C = 0.1$.

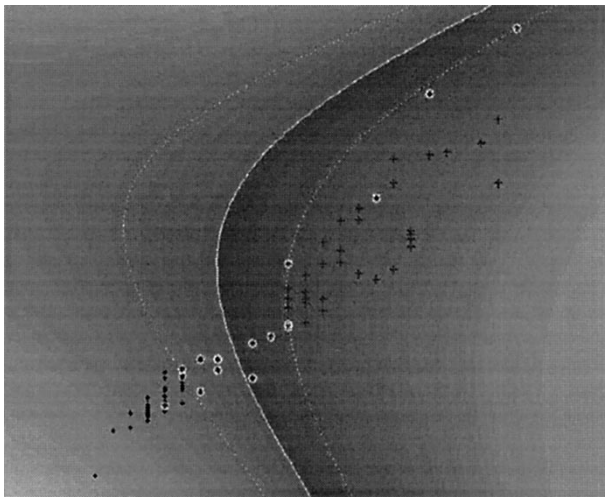

Fig. 11. Standard SVM training with polynomial kernel function (degree $= 2$) $C = 10$.

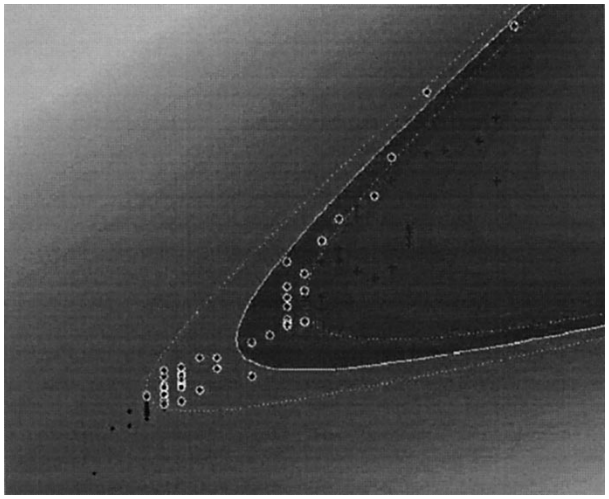

Fig. 12. Standard SVM training with polynomial kernel function (degree $= 2$) $C = 1000$.

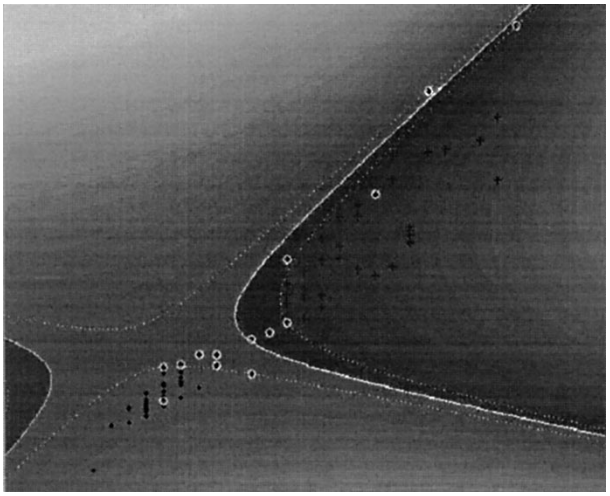

Fig. 13. Robust SVM training with polynomial kernel function (degree $= 2$), $\lambda = 0.1$.

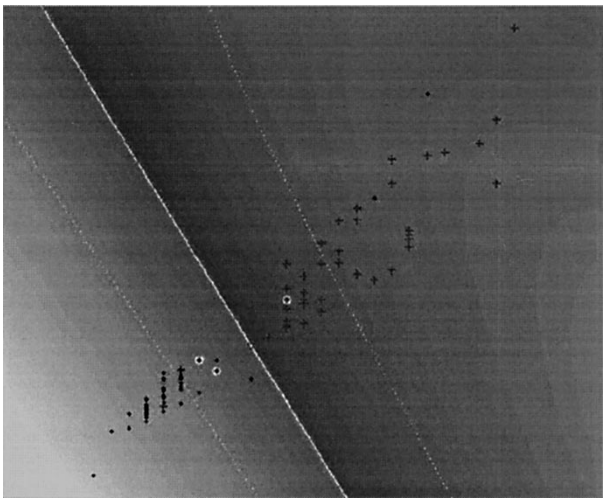

Fig. 14. Robust SVM training with polynomial kernel function (degree $= 2$), $\lambda = 10$.

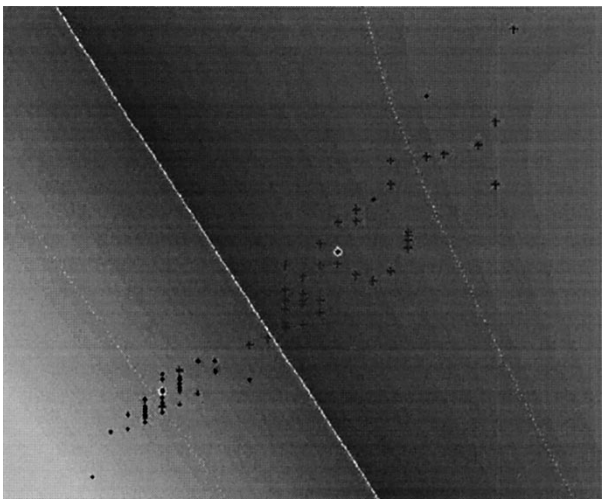

Fig. 15. Robust SVM training with polynomial kernel function (degree $= 2$), $\lambda = 100$.

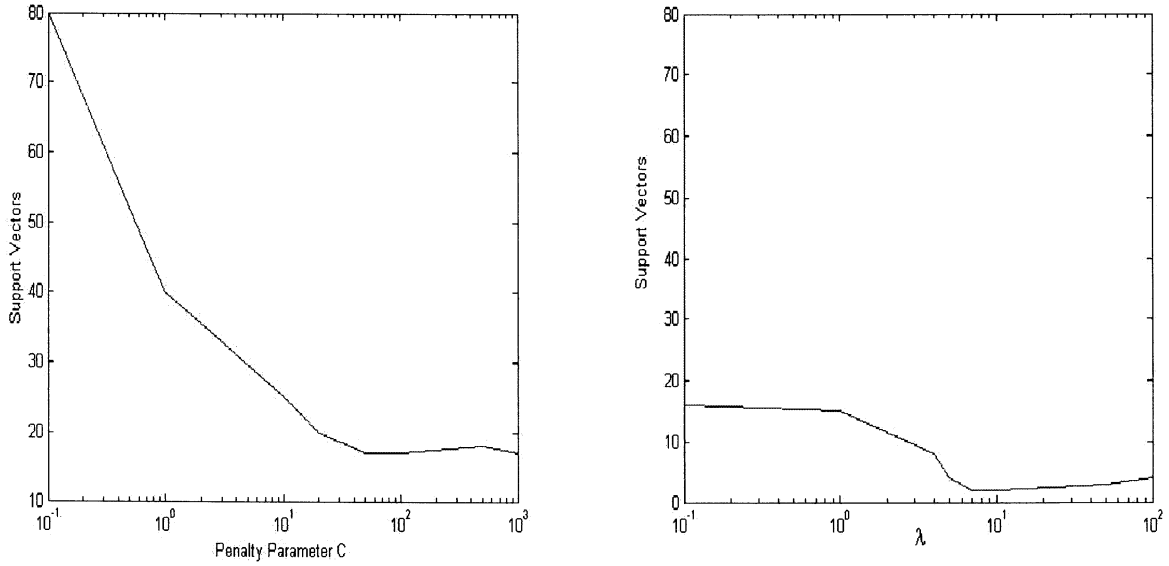

Fig. 16. Number of support vectors against penalty parameter C and λ. The left one is from standard Gaussian RBF SVM and the right one is from robust Gaussian RBF SVM.

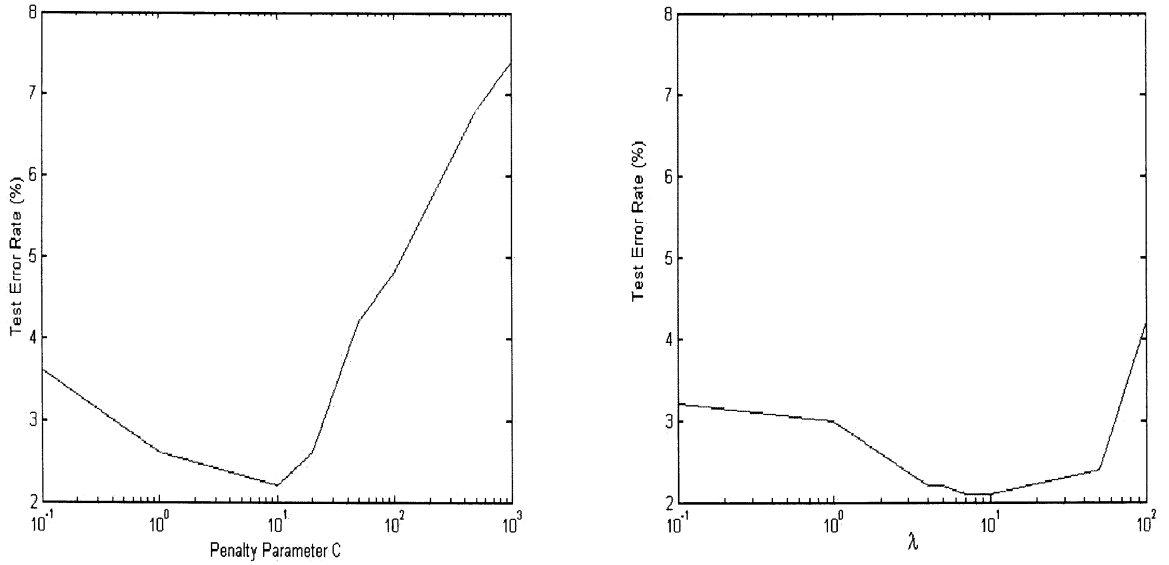

Fig. 17. Test error against penalty parameter C and λ. The left one is from standard Gaussian RBF SVM and the right one is from robust Gaussian RBF SVM.

To compare the robust SVM with the standard SVM, we train the robust SVM with different regularization parameter λ by using same contaminated training samples. Figs. 7–9 show the results of robust Gaussian RBF SVM with regularization parameter $\lambda = 0.1, 10, 100$ and Table III summarizes the detailed results. According to Table II and Table III, it is clear that the standard SVM with parameter $C = 10$ and the robust SVM with parameter $\lambda = 10$ have the best generalization performance (lowest test error rate). However, the robust SVM is less sensitive to the selection of penalty parameters compared to that of the standard SVM, particularly, when the penalty parameter is relatively larger to take care of the outliers. From Figs. 5 and 8, one may also notice that the decision boundary in Fig. 5 is deviated from the optimal hyperplane severely and the number of support vectors is increased significantly, which lead poor generalization performance, i.e., the test error is large. In robust SVM training method, the number of support vectors and the test error are reduced significantly compared with the standard SVM training method, particularly when a large penalty parameter is needed to take care of the outliers.

In order to show good gerenalization performance of the proposed algorithm, we have compared results between the robust SVM and standard SVM training with other kernel functions, say, Ppolynomial kernel function. Figs. 10–12 shows the result of standard polynomial SVM with penalty parameter $C = 0.1, 10, 1000$, and Figs. 13–15 show the result of robust polynomial SVM with regularization parameter $\lambda = 0.1, 10, 100$. Figs. 16 and 17 show the graphic illustration of Tables II and III, respectively. It is clear that training of robust SVM leads to very sparse suppor vectors and good generalization performance. More detailed results are shown in Tables IV and V. Figs. 18 and 19 show the graphic illustration of Tables IV and V, respectively.

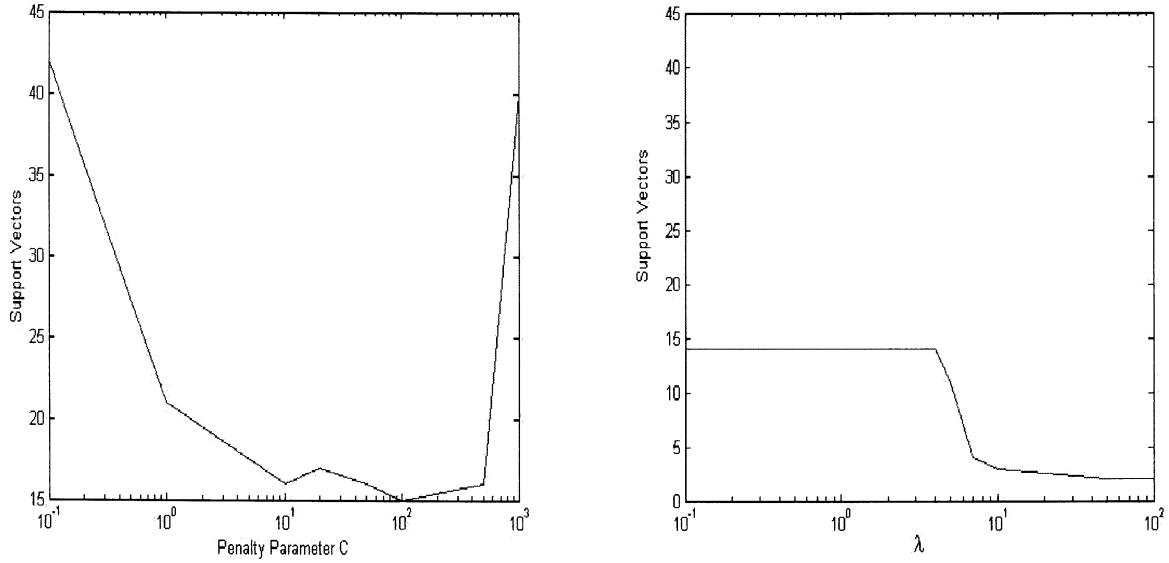

Fig. 18. Number of support vectors against penalty parameter C and λ. The left one is from standard polynomial SVM and the right one is from robust polynomial SVM.

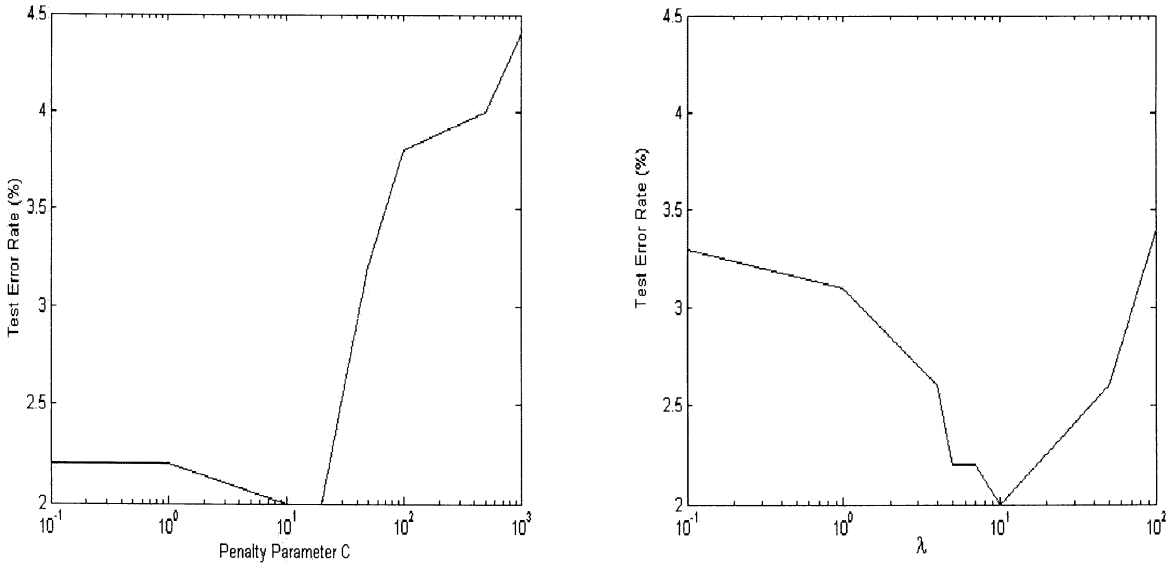

Fig. 19. Test error against penalty parameter C and λ. The left one is from standard polynomial SVM and the right one is from robust polynomial SVM.

TABLE IV
STANDARD SVM TRAINING WITH DIFFERENT PARAMETER C
(POLYNOMIAL KERNEL FUNCTION WITH DEGREE $= 2$)

C	Nsv	No.M	Test Error
0.1	42	20	2.2%
1	21	20	2.2%
10	16	18	2.0%
20	17	18	2.0%
50	16	29	3.2%
100	15	34	3.8%
500	16	36	4.0%
1000	40	40	4.4%

TABLE V
ROBUST SVM TRAINING WITH DIFFERENT PARAMETER λ
(POLYNOMILA KERNEL FUNCTION WITH DEGREE $= 2$)

λ	Nsv	No.M	Test Error
0.1	14	30	3.3%
1	14	28	3.1%
4	14	23	2.6%
5	11	20	2.2%
7	4	20	2.2%
10	3	18	2.0%
50	2	23	2.6%
100	2	31	3.4%

From Figs. 4–15, we get to know that the number of support vectors and the test error are reduced significantly in robust SVM training when there exist outliers in the training data samples. The decision boundary become less detoured compared with that in standard SVM training. The smoother decision boundary is obtained and classification accuracy is improved. It is also clear that the decision boundary becomes smoother as the parameter λ is increased. The support vectors are

reduced to two data points that are drawn toward the nearest points to the centers of two classes. To show improvement of generalization performance by controlling the parameter λ of the robust SVM training, we plot parameter λ against test error in Figs. 17 and 18.

V. Conclusion

This paper proposes a general robust SVM algorithm against outliers. By adding the distance between each data point and the center of classes to form the margin of separating hyperplane, good robust performance is achieved. The simulation of robust SVM with different kernel functions and regularization parameters has been presented to show that the robust algorithm can be used for pattern classification problems with different difficulty levels. The experiment results show that the decision boundary become less detoured and the number of support vectors of the robust SVM are reduced significantly compared to that of the standard SVM. Therefore, generalization performance of the robust SVM is obtained as shown in the simulation.

Acknowledgment

The authors would like to acknowledgement the suggestions of many people in Intelligent Machines Research Lab, School of Electrical and Electronics Engineering, Nanyang Technology University, Singapore

References

[1] V. N. Vapnik, *Statistical Learning Theory*. New York: Wiley, 1998.
[2] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifier," in *Proc. 5th ACM Workshop Comput. Learning Theory*, Pittsburgh, PA, July 1992, pp. 144–152.
[3] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining Knowl. Discovery*, vol. 2, no. 2, pp. 121–167, 1998.
[4] E. E. Osuna, R. Freund, and F. Girosi, "Support vector machines: training and application," MIT A.I. Lab., A.I. Memo 1602, 1997.
[5] ——, "Training support vector machines: An application to face dectection," in *Proc. Computer Vision Pattern Recognit.*, 1997, pp. 130–136.
[6] A. K. Jain, *Fundamentals of Digital Image Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1989.
[7] M. Oren, C. Papageorgious, P. Sinha, E. Osuna, and T. Poggio, "Pedestrian detection using wavelet templates," in *Proc. Computer Vision and Pattern Recognition*, 1997, pp. 193–199.
[8] T. Joachims. (1997) Text categorization with support vector machines. Univ. Dortmund, Dortmund, Germany. [Online]. Available: ftp:/ftp-ai.informatik.uni-dortmund.de/pub/Reports/report23.ps.gz
[9] I. Guyon, N. Matic, and V. N. Vapnik *et al.*, "Discovering informative patterns and data cleaning," in *Advances in Knowledge Discovery and Data Mining*, U. M. Fayyad *et al.*, Eds. Cambridge, MA: MIT Press, 1996, pp. 181–203.
[10] X. G. Zhang, "Using class-center vectors to build support vector machines," presented at the IEEE Proc. Neural Net. Signal Processing IX, Aug. 1999.
[11] R. Herbrich and J. Weston, "Adaptive margin support vector machines for classification," in *Proc. 9th ICANN*, vol. 2, Sept. 1999, pp. 880–885.
[12] S. Haykin, "Neural networks—A comprehensive Foundation," in *Chapter 6-Support Vector Machine*. Englewood Cliffs, NJ: Prentice-Hall, 1999.
[13] W. F. Xie, D. J. Hou, and Q. Song, "Bullet-hole image classification with support vector machines," presented at the IEEE Workshop Neural Networks Signal Processing, Australia, Dec. 2000.
[14] S. Gunn, "Support vector machines for classification and regression," Univ. Southampton, Southampton, U.K., ISIS Tech. Rep., May 1998. Image Speech and Intelligent Systems Group.

Qing Song (S'88–M'91) received the B.S., M.S., and Ph.D. degrees in electrical and electronic engineering from Harbin Shipbuilding Engineering Institute, China P.R., 1982, Dalian Maritime University, China P.R., 1986, and the University of Strathclyde, Strathclyde, U.K., in 1992, respectively.

Currently, he is an Associate Professor and active industrial consultant in the School of Electrial and Electronic Engineering, Nanyang Technological University, NTU, Singapore. He was a Research Engineer at System and Control Pte Lte, University of Strathclyde U.K. before joining NTU. He has held a few neural-network-related patents and has published more than 30 referrred international journal papers in neural networks, control, image processing, and related areas.

Wenjie Hu is pursuing the Ph.D. degree in electrical and electronic engineering with the School of Electrial and Electronic Engineering, Nanyang Technological University, Singapore. He has published close to a dozen papers in international journals and conferences in the areas of image processing, pattern recognition, neural networks, and artificial intelligence.

Wenfang Xie received the Ph.D. degree in electrical and electronic engineering from the Chinese University of Hong Kong, Hong Kong in 1999.

She was a Research Fellow in the School of Electrical and Electronic Engineering, Nanyang Technological University (NTU), Singapore. She is currently a Senior Research Engineer at CORETEC Incorporated, Waterloo, ON, Canada.