ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

Multiview Pedestrian Detection Based on Online Support Vector Machine Using Convex Hull

Revathi M K[1], Ramya K P[2], Sona G[3]

Information Technology, Anna University, Dr.Sivanthi Aditanar College of Engineering, Tuticorin-628215, Tamilnadu, India[1, 2, 3]

ABSTRACT— The support vector machine (SVM), an assuring new method for the classification, has been widely used in many areas efficiently. However, the online learning issue of SVM is still not addressed satisfactorily since when a new sample arrives to retrain the SVM to adjust the classifier. This may not be feasible for real-time applications due to the expensive computation cost for re-training the SVM. This paper propose an Online SVM classifier algorithm known as OSVM-CH, which is based on the convex hull vertices selection depends on geometrical features of SVM. From the theoretical point of view, the first d+1(d is the dimension of the input samples) selected samples are proved to be vertices of the convex hull. This guarantees that the selected samples in our method keep the greatest amount of information of the convex hull. From the pedestrian detection application point of view, the new algorithm can update the classifier without reducing its classification performance.

KEYWORDS-Kernel, online learning, machine learning, support vector machine, pedestrian detection, online classifier.

I. Introduction

Pedestrian detection is a challenging problem in real traffic, cluttered environments, as long as pedestrian detection should perform robustly under variable illumination conditions, variable rotated positions and pose, and even if some of the pedestrian parts or limbs are partially occluded. In order to ease the pedestrian recognition task, we propose the effective method based on the support vector machines.

Support Vector Machines (SVMs) are a democratic machine learning method for classification, regression, and other learning tasks. Geometrically, it is to find a hyper plane with a maximal margin between two classes. The main advantages of the SVM method are summarized as follows [1], [2]. First, because of the combination of the empirical risk minimization and the prevention of over fitting, SVM can achieve a good generalization performance. Second, the problem of computing the hyper plane with a maximal margin can be converted into a convex quadratic optimization problem, which can be solved by quadratic programming techniques and has a global optimal solution. Finally, the obtained classifier is found out only by support vectors, and it can also be applied to nonlinear cases by using the kernel trick. Due to the above advantages, SVM has been successfully used in many real world problems:

- Hand-written characters can be recognized using SVMs.
- Pattern Recognition
- Hand-writing digital character recognition
- Face Recognition
- Classification tasks of text

However, a main drawback of SVM is that it requires a long time for training and a lot of memory for dealing with large scale data sets. Online learning has significant applications in real-time pattern recognition systems, such as pedestrian detection and aircraft visual navigation systems. In order to address the online learning issue of SVM, several successful algorithms has [3], [4], [5] been proposed, which will be reviewed in the next section. There are few works that consider updating SVM online by preserving the skeleton samples based on the geometric characteristics of SVM. Geometrically, for linearly separable data sets, the computation of SVM is equivalent to the problem of computing the nearest points between the convex hulls formed by the positive and negative samples [6]. Therefore, we can permanently and safely delete samples within the convex hulls and only preserve the vertices of the convex hulls for online learning [7]. The classification result of the online classifier trained with only the vertices of the convex hulls and newly arrived samples is the same as that obtained by retraining with all samples. Based on the above motivation, in this paper, we propose an online SVM classification algorithm. A main advantage of our approach over the existing ones is that it can deal with large-scale data sets efficiently since only a small number of samples are selected for online learning. Based on the above motivation, in this paper, we propose an online SVM classification algorithm. A main advantage of our approach over the existing ones is that it can deal with large-scale data sets expeditiously since only a small number of samples are selected for online learning. There are two main steps in our algorithm, which are described as follows.

1) Offline Process: The skeleton vertices of convex hulls, which are formed by the current positive and negative training samples, are selected and stored for the following online process.

2) Online Process: The SVM classifier is updated based on samples selected in the off-line process and the newly arriving samples whose distances to the current classification hyper plane are within a given threshold.

We apply the offline process only in the case when the number of the current training samples, which consist of the selected samples and all newly arrived samples exceeds a given threshold. We execute sporadically the off-line step in order to reduce the number of the current training samples. This makes the online learning with newly arrived samples possible.

II. Online SVM Based on Convex Hull

There are two main steps in our algorithm, which are described as follows. Offline Process: The skeleton vertices of convex hulls, which are formed by the current positive and negative training samples, are selected and stored in the following online process. Online Process: The SVM classifier is updated based on samples selected in the off-line process and the newly arriving samples whose distances to the current classification hyper plane are within a given threshold.

A. Feature Extraction

Transforming the input data into the set of features is called feature extraction. If the features extracted are carefully chosen it is expected that the features set will extract the relevant information from the input data in order to perform the desired task using this reduced representation instead of the full size input. A common feature extraction technique is Histogram of Oriented Gradients (HOG). HOG are feature descriptors used in computer vision and image processing for the purpose of object detection. The technique counts occurrences of gradient orientation in localized portions of an image. This method is similar to that of edge orientation histograms, scale-invariant feature transform descriptors, and shape contexts, but differs in that it is computed on a dense grid of uniformly spaced cells and uses overlapping local contrast normalization for improved accuracy. The aim of such method is to describe an image with a set of local histograms. These histograms count occurrences of gradient orientation in a local part of the image. In this work, in order to obtain a complete descriptor of an infrared image, we have computed such local histograms of gradient according to the following steps:

1. Compute gradients of the image,
2. Build histogram of orientation for each cell,
3. Normalize histograms within each block of cells.

The following paragraphs give more details on each of these steps.

1) Gradient computation: The gradient of an image has been simply obtained by filtering it with two one dimensional filters:

$$\text{Horizontal: } (-1 \;\; 0 \;\; 1)$$

$$\text{Vertical} \quad : (-1 \;\; 0 \;\; 1)^T$$

Gradient could be signed or unsigned. This last case is justified by the fact that the direction of the contrast has no importance. In other words, we would have the same results with a white object placed on a black background, compared with a black object placed on a white background. In our case, we have considered unsigned gradient which values going

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

from 0 to π. The next step is orientation binning, which is to say to compute the histogram of orientation. One histogram is computed for each cell according to the number of bins.

2) Cell and block descriptors: The particularity of this method is to split the image into different cells. A cell can be defined as a spatial region like a square with a predefined size in pixels. For each cell, we then compute the histogram of gradients by accumulating votes into bins for each orientation. Votes can be weighted by the magnitude of a gradient, so that histogram takes into account the importance of gradient at a given point. This can be justified by the fact that a gradient orientation around an edge should be more significant than the one of a point in a nearly uniform region.

When all histograms have been computed for each cell, we can build the descriptor vector of an image concatenating all histograms in a single vector. However, due to the variability in the images, it is necessary to normalize cells histograms. Cells histograms are locally normalized, according to the values of the neighbored cells histograms. The normalization is done among a group of cells, which is called a block. A normalization factor is then computed over the block and all histograms within this block are normalized according to this normalization factor. Once this normalization step has been performed, all the histograms can be concatenated in a single feature vector.

Different normalization schemes are possible for a vector V containing all histograms of a given block. The normalization factor nf could be obtained along these schemes:

- None: no normalization applied on the cells, $nf = 1$.
- L1-norm: $nf = \frac{V}{||V||_1+\xi}$ (1)
- L2-norm: $nf = \frac{V}{||V||_{\frac{2}{2}}+\xi^2}$ (2)

ξ is a small regularization constant. It is needed as we sometime evaluate empty gradients. The value of ξ has no influence on the results. Note that according to how each block has been built, a histogram from a given cell can be involved in several block normalization. In thus case, the final feature vector contains some redundant information which has been normalized in a different way. This is especially the case if blocks of cells have overlapping. The Extracted features are the input of the convex hull-vertices selection algorithm.

A. Convex Hull-Vertices Selection Algorithm

1) Convex Hull Distance

The convex hull of the set $P = \{x_i\}_{i=1}^{n} \subset \mathbb{R}^d$ is defined as:

(3)

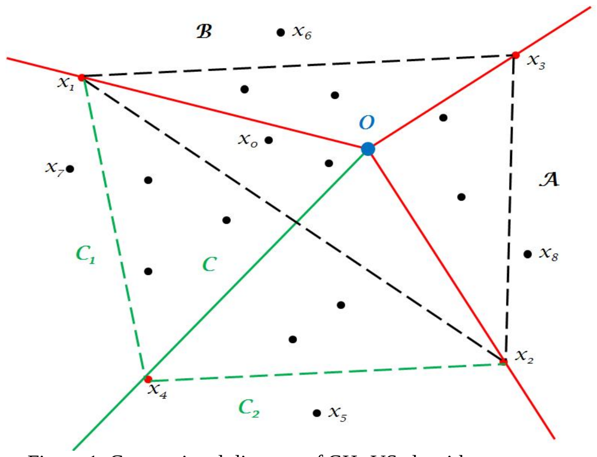

Figure 1. Conventional diagram of CH- VS algorithm

$$\mathrm{con}(P) = \left\{\sum_{i=1}^{n} \alpha_i\, x_i | x_i \in P, \sum_{i=1}^{n} \alpha_i = 1, \alpha_i \geq 0, i = 1, \ldots, n\right\}$$

Given a set $P = \{x_i\}_{i=1}^{n} \subset \mathbb{R}^d$ and a point $x \in \mathbb{R}^d$, the Euclidean distance between and x and $\mathrm{con}(P)$ can be computed by solving the following quadratic optimization:

$$\min_{\alpha} \frac{1}{2}\alpha^T Q\alpha - c^T \alpha \quad \text{s.t.}\, e^T \alpha = 1, \alpha \geq 0 \tag{4}$$

Where $e = [1,1,\ldots,1]^T$, $Q = X^T X$ and $c = X^T x$ with $X = [x_1,\ldots,x_n]$. suppose the optimal solution of (1) is α^*. then the convex hull distance between x and $\mathrm{con}(P)$ is given by

$$d_c(x,\mathrm{con}(P)) = \sqrt{x^T x - 2c^T\alpha^* + \alpha^{*T} Q\alpha^*} \tag{5}$$

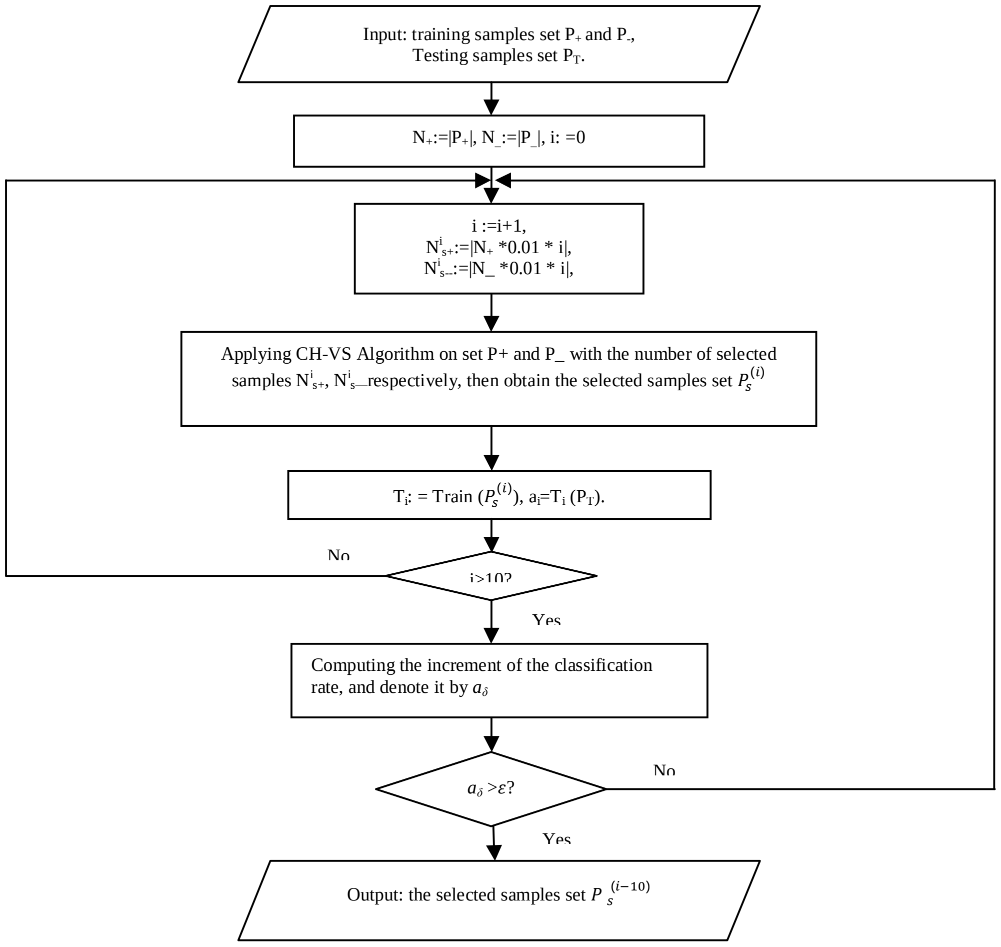
Input: training samples set P_+ and P_-, Testing samples set P_T.
$N_+ := |P_+|$, $N\_ := |P\_|$, i: =0
i :=i+1, $N^i_{s+} := |N_+ * 0.01 * i|$, $N^i_{s-} := |N\_ * 0.01 * i|$,
Applying CH-VS Algorithm on set P+ and P_ with the number of selected samples N^i_{s+}, N^i_{s-} respectively, then obtain the selected samples set $P_s^{(i)}$
T_i: = Train ($P_s^{(i)}$), $a_i = T_i$ (P_T).
No
i>10?
Yes
Computing the increment of the classification rate, and denote it by a_δ
$a_\delta > \varepsilon$?
No
Yes
Output: the selected samples set $P_s^{(i-10)}$

Fig 2. Flowchart of CH-OSVM Algorithm

2) Initial Vertices Selection: Before formally Introducing the sample selection method (offline process), to present a synthetic example in R^2 is shown in Fig.1 to explain the basic idea of the proposed method. As shown in Fig.1, first select three samples with the following steps. First, randomly select a sample, denoted by x_0. Second, the furthest sample to x_0 is selected and denoted by x1. Thirdly, the furthest sample to x1 is selected and denoted by x2. Finally, the sample that is the

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

furthest to the straight line passing through $x1$ and $x2$ is selected and denoted by $x3$. To select the $d + 1$ samples that can construct an as large d-simplex as possible. Suppose O is the mean of the samples $\{x1, x2, x3\}$. Then radials $Ox1, Ox2$ and $Ox3$ divide the samples outside of $con(\{x_1, x_2, x_3\})$ into three parts, denoted by A, B, and C severely.

3) Data Partition: From Fig.1, it can be seen that the number of samples in part C is still la e, so need to further divide this part into several smaller parts. In part C, the sample which is furthest to the line x_1x_2 is selected and denoted by x_4. Then the radial Ox_4 further divides the samples in C but outside of $con(\{x_1, x_2, x_3, x_4\})$ into two parts, denoted by C_1 and C_2. So far, divided the samples outside of $con(\{x_1, x_2, x_3, x_4\})$ into four parts and the numbers of samples in the four parts are all small. The edges $x_2x_3, x_1x_3, x_1x_4, x_2x_4$ are called the corresponding edges of parts A, B, C_1, and C_2, respectively.

4) Final Vertices Selection: From Fig.1, it can be seen that the distance to $con(\{x_1, x_2, x_3, x_4\})$ of each sample in each part is equal to its distance to the representing edge of that part. For example, the distance to $con(\{x_1, x_2, x_3, x_4\})$ of each sample in part A is equal to the distance of the sample to $con(\{x_2, x_3\})$ (the corresponding edge of part A). To call the convex hull of the samples in one part as a sub-convex hull which is initialized to be representing the edge of that part, that is, the sub-convex hulls of parts A, B, C_1, and C_2 are initialized to be the edges $x_2x_3, x_1x_3, x_1x_4, x_2x_4$ respectively and individually compute the distances between the samples and the sub-convex hull in each part. Then the furthest sample is selected, which is denoted as x_5. The sub-convex hull of part C_2 which x_5 belongs to is then spanned by adding x_5 to the vertices of the sub-convex hull. The distances between the samples in part C_2 and the spanned sub-convex hull are recomputed, while the distances between samples in other parts and their corresponding sub-convex hulls will not be changed. Then the next furthest sample will be selected. In such a procedure, samples x_6, x_7, x_8 are selected in turn.

C. Online SVM Based on Convex Hull

Fig.2 shows Flowchart of CH-OSVM Algorithm. N^i_{s+} and N^i_{s-} represent the numbers of samples that need to select from the positive samples set P+ and the negative samples set P−, respectively, by using the CH-VS algorithm (see the fourth box) in the ith iteration. Train $(P^{(i)}_s)$ represents the fact that the sample set $(P^{(i)}_s)$, which is selected in the ith iteration is used to train the SVM classifier. The classifier is denoted by r_i, and a_i is the classification rate of r_i on the testing samples. $a_\delta = [\alpha^j_\delta]_{1\times10}$ is an increment vector of the classification rate, where $\alpha^j_\delta = a_{i-10+j} - a_{i-10}.a\delta \leq \varepsilon$ represents that all elements of a_δ are not larger than ε, where ε is a small threshold. This means that the set $P_s^{(i-10)}$ of selected samples already has the greatest amount of information of the training samples. In our experiments, always set $\varepsilon = 1/|PT|$, where $|PT|$ the number of the testing samples is. As shown in Fig.2, in each iteration, to add one percent of the training samples on the basis of the selected samples set in the last iteration. That is, in the ith -iteration of CH-VS algorithm, select $[N+ * i * 0.01]$ positive samples and $[N- * i * 0.01]$ negative samples, respectively, by applying the CH-VS algorithm to each class. To use these selected samples to train the SVM classifier.

III. EXPERIMENTAL RESULTS

In the samples selection process, the average classification rates measured on testing samples, as a function of the number of selected samples with the proposed samples selection method (red line) and the random selection method (green line), are shown in Fig. 3. It can be seen that the samples selected by our method is more effective than the samples selected by the RS method. Fig.4.shows pedestrian detection results of our proposed algorithm (CH-OSVM).

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

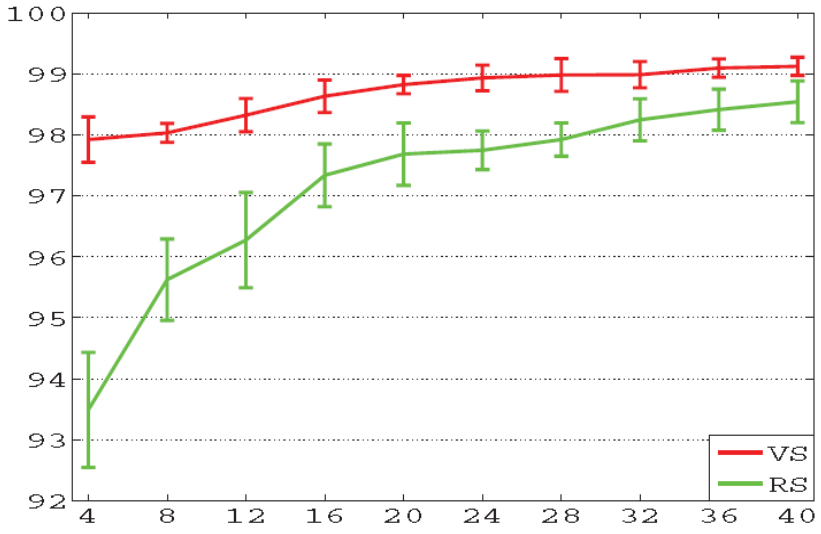

Fig.3. Classification rate of SVM trained with samples selected from training samples by CH-VS (red line) and the classification rate of SVM trained with randomly selected samples (green line).

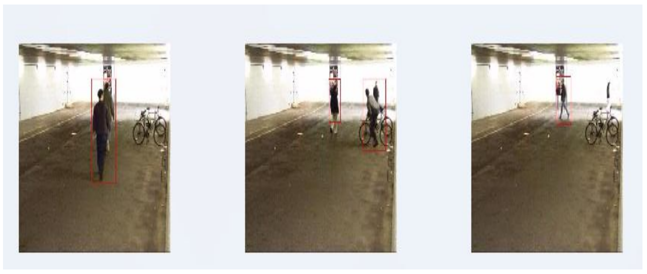

Fig.4. Pedestrian Detection Results of our proposed algorithm (CH-OSVM)

IV. CONCLUSION

This paper provides a new online SVM classifier based on the selection of vertices of the convex hull in each class. In the sample selection process, a small number of skeletal samples, which constitute the approximate convex hull of the current training samples, were selected for online learning. In the online classification process, the classifier was updated online based on the selected samples and newly arriving samples. It was proved theoretically that the d + 1 selected samples are all vertices of the convex hull. It guarantees that the samples selected by our approach retain the greatest amount of information about the current training samples. From the application point of view, the online adjustment of an SVM classifier based on the selected skeleton samples improves the classification rate very little but consumes much less time. Therefore, the proposed online SVM can be applied to various online classification tasks, such as visual tracking, Hand-writing digital character recognition. Experimental results on pedestrian data sets have validated the effectiveness of the proposed method.

REFERENCES

[1] Anguita.D, Ghio.A, Oneto.L, and Ridella.S, "In-sample and out of- sample model selection and error estimation for support vector machines," IEEE Trans. Neural Netw. Learn. Syst., vol. 23, no. 9, pp. 1390–1406,September,2012.

[2] Agarwal.S, Saradhib.V, and Karnick.H, "Kernel-based online machine learning and support vector reduction," Neurocomputing, vol. 71, nos. 7–9, pp. 1230–1237,2008.

[3] Barber.C.B and Dobkin.D.P, "The Quickhull Algorithm for Convex Hulls," ACM Transactions on Mathematical Software, Vol. 22, No. 4, Pages 469–483,December,1996.

[4] Bottou.L, "Online Learning and Stochastic Approximations", Cambridge, U.K.: Cambridge Univ. Press, pp. 9–42,1998.

[5] Bordes.A, Ertekin.S, Weston.J, and Bottou.L, "Fast kernel classifiers with online and active learning," J. Mach. Learn. Res., vol. 6, pp. 1579–1619, September,2005.

[6] Bordes.A and Bottou.L, "The Huller: A simple and efficient online SVM," in Proc. 16th Eur. Conf. Mach. Learn., Oct. 2005, pp. 505–512,October,2005.

[7] Cheng.S and Shih.F, "An improved incremental training algorithm for support vector machines using active query," Pattern Recognit., vol. 40, no. 3, pp. 964–971,2007.

[8] Geebelen.D, Suykens.J.A.K, and Vandewalle.J,"Reducing the number of support vectors of SVM classifiers using the smoothed separable case approximation," IEEE Trans. Neural Netw. Learn. Syst, vol. 23, no. 4, pp. 682–688, April,2012.

[9] Menon.A,"Large-Scale Support Vector Machines: Algorithms and Theory"[Online]. Available: http://cseweb.ucsd.edu/~akmenon/ResearchExam.pdf,2009.

[10] Shilton.A, Palaniswami.M, Ralph.D, and Tsoi.A. "Incremental training of support vector machines," IEEE Trans. Neural Netw., vol. 16, no. 1, pp. 114–131, January, 2005.

[11] Shilton.A, Palaniswami.M, Ralph.D, and Tsoi.A, "Incremental training of support vector machines," IEEE Trans. Neural Netw., vol. 16, no. 1, pp. 114–131, January, 2005.

[12] Singh.R, Vatsa.M, Ross.A, and Noore.A, "Biometric classifier update using online learning: A case study in near infrared face verification," Image Vis. Comput., vol. 28, no. 7, pp. 1098–1105 ,July,2010.

[13] Vapnik. V, "The Nature of Statistical Learning Theory". New York: Springer-Verlag, 1995.

[14] Vapnik. V,)."An overview of statistical learning theory," IEEE Trans. Neural Netw., vol. 10, no.5, pp. 988–999,September,1999.

[15] Lau.K and Wu.Q, "Online training of support vector classifier," Pattern Recognit., vol. 36, no. 8, pp. 1913–1920,2003.