Pedestrian Detection Using Background Subtraction Assisted Support Vector Machine

Yuan Xu, Lihong Xu, *Member, IEEE*, Dawei Li, Yang Wu
Dept.of Control Science and Engineering,
Tongji University,
Shanghai, China
xuyuan_new@yahoo.cn, ldwei1986@163.com

***Abstract*—This paper achieves fast and effective pedestrian detection using Histogram of Oriented Gradient (HOG) descriptor based Support Vector Machine (SVM). A novel approach taking advantage of CodeBook background subtraction(CBBS) is presented in this paper to produce pedestrian samples for SVM. HOG features of the samples are extracted to train Linear and RBF SVM classifiers offline. The classifier is adopted as pedestrian detector in online real-time video sequence detection. The influence of various ratios of positive and negative training sets on detector's performance is carefully investigated. We also compare Linear and RBF SVM in experiments as well. It is concluded that robust feature extraction, proper positive and negative training sample construction, and fine kernel function are crucial for good classification results. Experiments prove that our detector obtains a reliable detection result, which not only satisfies real-time requirement, and is robust against pedestrian appearance and pose variations, illumination changes, background changes, shadows and etc.**

Keywords-Codebook background subtraction, HOG descriptor, SVM, Machine learning, Pedestrian detection

I. INTRODUCTION

The objective of pedestrian detection is to make computer intelligent in identifying pedestrians from images or video data automatically. Human detection and recognition is an important research area in computer vision and pattern recognition, which can be applied broadly in video surveillance, robot vision, virtual reality technology, and etc. The present difficulties mainly lie in the polymorphism and non-rigid features of pedestrian, illumination variation and the complexity of background environment.

Currently there are two methods for pedestrian detection focusing on motion [1] and shape [2] respectively. The popular motion-based methods such as Gaussian Mixture Models (GMM) [3] and Kernel Density Estimation (KDE) [4] generally root in statistical methods. The shape-based approaches detect target by analyzing image intensity, contour, edge and other feature information. Common algorithms in this regard are Explicit human model, Template matching and Machine learning techniques. The Explicit human model is based on block detection, which includes model building and entire human body identification. Template matching examines the similarity between input window and templates established previously in order to find a suitable object candidate. Machine learning techniques aim at building an intelligent classifier. It requires target feature description from input space. Existing feature description methods are: symmetry and edge density characteristics [5], Viola's Haar-like features [6], SIFT descriptor proposed by Lowe [7], and Shapelet features by Sbazmeydani [8]. Most of all, Dalal and Triggs [9] proposed the Histogram of Gradient feature (HOG) descriptor, which proves to be effective in human detection.

This paper uses the 2-class SVM classifier and HOG features to build the pedestrian detector. The CBBS method is innovatively applied to enrich the training samples for SVM. The content of this paper is organized as follows: Section II gives an overview of algorithm framework; Section III describes CBBS-based [10] method in acquiring training set; Section IV and V introduce basic principles of HOG Feature and SVM respectively. Section VI specifies the influences on classifier by modifying the training stage. Classifier performance evaluation and real video stream detection results are also given; Section VII is the conclusion and future work.

II. OVERVIEW OF THE ALGORITHM

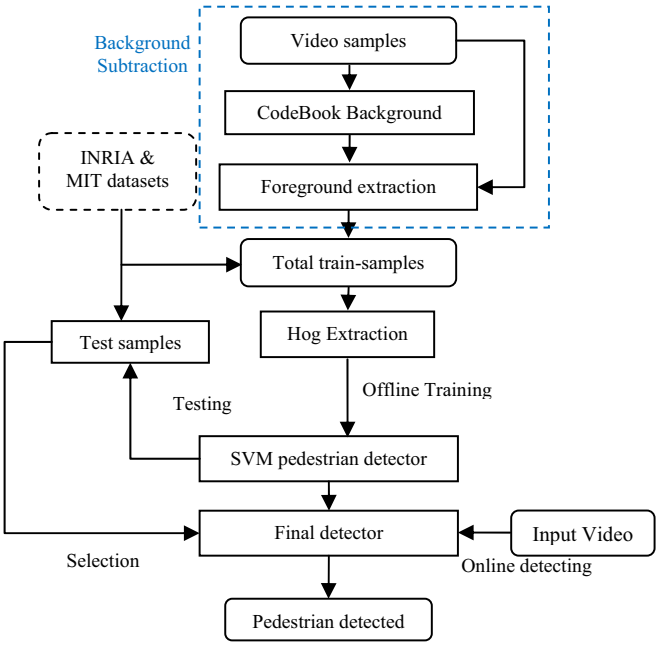


Figure 1. Algorithm framework

This section gives an overview of our method in Fig.1. It consists of offline SVM classifier training and online

real-time video sequence detection. Offline training process includes the feed of training samples, HOG feature extraction, SVM training, as well as the online detection stage. To enrich the INRIA and MIT human database with more positive training samples, CBBS is used to extract pedestrians from various scenarios. Furthermore, after negative samples are given proportionally with respect to the quantity of positive samples, we extract HOG feature of all the samples by HOG descriptor to train SVM offline. Finally, online real-time video sequence pedestrian detection is carried out using the most appropriate detector according to the evaluation of classifier performance. We also analyze how the ratio of positive and negative training samples affects SVM classifier.

III. AUGMENT PEDESTRIAN SAMPLES BY BACKGROUND SUBSTRACTION

CBBS algorithm constructs a codebook background model akin to winner-take-all clustering technique through consistent observation of a video stream. A codebook is built for each pixel, consisting of one or more codewords. Samples at each pixel are clustered into the set of codewords based on a color distortion metric together with brightness bounds. Not all pixels have the same number of codewords. The clusters represented by codewords do not necessarily correspond to single Gaussian or other parametric distributions. Even if the distribution at a pixel were a single normal, there could be several codewords for that pixel. The background is encoded on a pixel-by-pixel basis. Detection involves testing the difference of the current image from the background model with respect to color and brightness differences.

Foreground detection is carried out subsequently according to background model. If the incoming pixel matches its codebook, the pixel belongs to background, meeting two conditions: the color distortion to some codeword is less than the detection threshold; and its brightness lies within the brightness range of that codeword [10]. Otherwise foreground. Moving pedestrian can be quickly discriminated as foreground target from video frames to enrich our training samples by CBBS.

Positive samples refer to pictures with uniform size which contain upright pedestrians, while negative samples do not contain any human body, and no size restrictions. Non-pedestrian backgrounds in positive samples and negative samples should be diversified to ensure a good sampling of non-pedestrian cases as the reason that the non-pedestrian regions are generally more complicated in color and texture appearance than their counterparts. In this paper, the basic training samples are directly obtained from well-known pedestrian datasets. We then extend this basic dataset by employing the CBBS on various practical video streams. After the pedestrian in foreground area is detected by CBBS, a smallest window which contains that pedestrian target is saved to positive data samples, while fixed-scaled windows are randomly selected from background region and are adopted as complementary negative samples. 924 pedestrian samples are from MIT, 2416 from INRIA. 639 pedestrian samples are extracted from some videos by CBBS Total up to 3979 positive sample images are available for the training. After gradation unification, the positive samples are scaled to size of 64×128 pixels, and histogram equalization is applied to enhance their contrast with background. We prepare mass negative samples selected from INRIA & MIT pedestrian datasets and some real scenes. Fig.2 shows some positive and negative samples used in training, wherein (a) are positive samples, (b) are negative ones.

Figure 2. Part of samples from INRIA, MIT and our CBBS-based results.

IV. HOG FEATURE EXTRACTION

A. Basic Algorithm Principle

HOG describes the distribution of image local gradients on different orientations, which perfectly characterizes appearance and shape of local object. Input image is divided to small connected spatial regions (cells) and all pixels inside the cell take part in the voting in histogram of gradient orientations with gradient magnitude as weights. The corresponding histogram reflects the pixel distribution with respect to gradient orientation. For better invariance to illumination and shadowing effect, we group neighbouring cells to form a block and normalize the local histogram within the block. The normalized descriptor blocks are called HOG descriptor.

French (INRIA) researcher Navneet, Dalal and Bill Triggs first proposed HOG method in [9].They suggested the cell of 8×8 pixels, block of 2×2 cell to be the best configuration. Each cell consists of 9 orientation bins covering $0^{\circ} \sim 180^{\circ}$. 4 cells are connected to form HOG feature vector with 4×9=36 dimensions of each block.

B. Implemetation

Step 1: Gamma Normalization

To reduce the influence of illumination variations in different images, we have the image gamma normalized in grayscale space.

Step 2: Gradient Computation

We use 1 dimensional (1-D) discrete differential template [-1, 0, 1] to compute horizontal gradient $G_x(x, y)$ and vertical gradient $G_y(x, y)$ of every pixel $I(x, y)$, which can capture contour information of human body meanwhile weaken the illumination uncertainty.

$$G_x(x, y) = [-1, 0, 1] * I(x, y) \quad (1)$$

$$G_y(x, y) = [-1, 0, 1]^T * I(x, y) \quad (2)$$

Compute the norm and orientation of each pixel.

$$G(x, y) = \sqrt{G_x(x, y)^2 + G_y(x, y)^2} \quad (3)$$

$$\theta(x,y) = \arctan(\frac{G_y(x,y)}{G_x(x,y)}) \quad (4)$$

Then we divide $0^\circ \sim 180^\circ$ into 9 equal intervals. If some pixel's gradient orientation is within some bin above, then the pixel's gradient magnitude is its weight in this interval, and its weight in other intervals is 0. Feature vector of this pixel is presented as follows:

$$H_k(x,y) = \begin{cases} G(x,y), & \theta(x,y) \in bin(k) \\ 0, & \theta(x,y) \notin bin(k) \end{cases} \quad 1 \le k \le 9 \quad (5)$$

Step 3: cell-based HOG unit

Feature vector of each cell consists of the addition of histograms of all pixels inside the cell.

Step 4: Combine HOG of each cell into block HOG

HOG of each block is the concatenation of HOG of all cells belonging to the block, with the dimension of $4 \times 9 = 36$ in usual configuration. Adjacent blocks are overlapped because of the sharing of some cells. This means every cell plays a role in multiple blocks. There are three parameters characterizing one block: cells number, pixel number and bin's number of each cell. The block is applied with a spatial Gaussian window to reduce weights of pixels near the edge.

Step 5: HOG normalization

$$v = \frac{v}{\sqrt{\| v \|_2^2 + \varepsilon}} \quad (6)$$

Equation (6) gives a normalization process, where v represents feature vector of each block, $\| v \|_2$ is the L2-norm, ε is a small constant introduced to avoid numerical problem.

Step 6: Concatenate HOG feature vectors of all blocks in the sample image to form our final HOG descriptor.

V. SUPPORT VECTOR MACHINE (SVM)

Support Vector Machine is developed by Cortes and Vapnik [11] in 1995. It maps input vector nonlinearly to a very high-dimensional feature space and constructs a linear decision plane in this feature space. Special properties of this decision plane ensure high generalization ability of the classifier tackling real-life problems. SVM shows many special advantages in resolving small samples classification and nonlinear high-dimensional pattern recognition. SVM takes structural risk minimization as its learning strategies.

A. Basic SVM training

The set is said to be linearly separable if there exists a vector w and a scalar b for decision functions as below:

$$g(x) = w \otimes x + b \quad (7)$$

$$\mathrm{sgn}(g(x)) = \begin{cases} 1 & x \in Class\ 1 \\ -1 & x \in Class\ 2 \end{cases} \quad (8)$$

where x is the feature vector to be classified, $\otimes$ represents Hilbert space dot-product, the general form is $u \otimes v = K(u,v)$, the kernel function $K(u,v)$ accepts two input vectors with same dimension and gives their dot-product in high-dimensional feature space. According to Hilbert–Schmidt principle, functions meeting Mercer condition can be chosen as kernel function [11]. Different kernel functions result in different function decision sets. Some common kernel functions are: Linear, Polynomial, and RBF function. Linear and RBF kernel function can be expressed as follows:

$$K_{Linear}(\mu,\nu) = \mu \cdot \nu$$

$$K_{RBF}(\mu,\nu) = e^{(-\gamma\|\mu-\nu\|^2)}$$

$w \otimes x + b = 0$ is the separation plane, which takes on a form of a dot in 1-D space, a straight line in 2-D space, and a plane in 3-D space, and a hyperplane in high-dimensional space. Define the distance d_i of sample x_i to some hyperplane:

$$d_i = y_i(w \otimes x_i + b) = | w \otimes x_i + b | = | g(x_i) | \quad (9)$$

In (9) y_i is the label assigned to x_i, $y_i = 1$ for class 1, -1 for class 2. Normalize w and b by L2-norm $\| w \|$, then we have the geometric margin δ_i, given as follows,

$$\delta_i = \frac{1}{\| w \|} | g(x_i) | \quad (10)$$

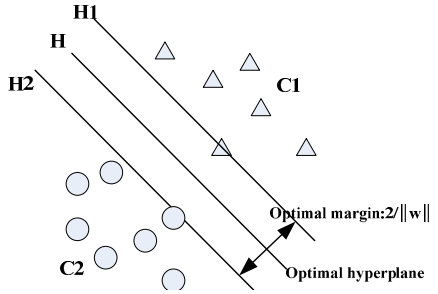


Figure 3. geometric interval schematic

Fig.3 shows an example in 2-D space. H is a separation line represented by $g(x) = wx + b = 0$, paralleling to H1($g(x) = wx + b = 1$) and H2 ($g(x) = wx + b = -1$). And samples which are closest to H lie on H1 and H2 respectively. Geometric margin is defined as the distance between H1 and H, H2 and H. We try to find the unique optimal hyperplane which can separate training data with a maximal margin $2/\| w \|$, that is:

$$\min\left\{\frac{1}{2}\| w \|^2\right\} \quad (11)$$

Subject to:

$$y_i[(w \otimes x_i) + b] - 1 \ge 0 \quad (12)$$

Where $i = 1,...,l$ and l is the number of data point. We define Lagrange function:

$$L(w,b,\alpha) = \frac{1}{2}\| w \|^2 - \sum_{j=1}^{l} \alpha_j (y_j(w \otimes x_j + b) - 1)) \quad (13)$$

By setting $\frac{\partial}{\partial w} L(w,b,\alpha) = 0$ we get:

$$w = \sum_{j=1}^{l} \alpha_j y_j x_j \tag{14}$$

Where α_j is the Lagrange Multiplier.

B. *Variants of SVM*

(1) Introduction of slack variable

Consider the case (showed in Fig.4) where the training data can not be separated without any error. In this case, we want to minimize the error by introducing some non-negative slack variables $\varsigma_i \geq 0$, $i = 1,...,l$. Then constraints change to:

$$y_i[(w \otimes x_i) + b] \geq 1 - \varsigma_i \tag{15}$$

And objective function becomes:

$$\min\left\{\frac{1}{2}\| w\|^2 + C\sum_{i=1}^{l}\varsigma_i\right\} \quad \varsigma_i \geq 0 \tag{16}$$

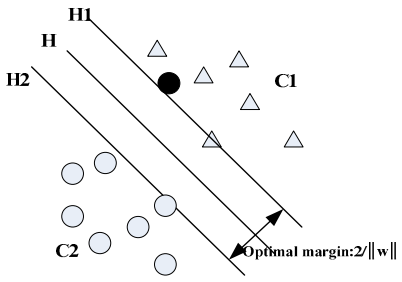


Figure 4. Separation with some error

(2) Introduction of Asymmetric Risks

Two types of risks exist in pedestrian detection, one is False Negative, the other is False Positive. Table I is a contingency table for a single test:

TABLE I. CONTINGENCY TABLE FOR A SINGLE TEST

Contingency Table		Test Sample	
		Non-Pedestrian	Pedestrian
Detection Result	Non-Pedestrian	True Negative (TN)	False Negative (FN)
	Pedestrian	False Positive (FP)	True Positive (TP)

Relatively, consequence of FN risk outweighs FP risk, hence the penalty factors for these two slack variables should not be identical. This paper adopts the design approach of risk sensitive SVM classifier based on two types of asymmetry risks [12].

$$C_{FN}\sum_{i|y_i=1}\varsigma_i + C_{FP}\sum_{i|y_i=-1}\varsigma_i \tag{17}$$

Where C_{FN} is the FN risk, C_{FP} is the FP risk, $C_{FN} > C_{FP}$.

The objective function becomes:

$$\min\left\{\frac{1}{2}\| w\|^2 + C_{FN}\sum_{i|y_i=1}\varsigma_i + C_{FP}\sum_{i|y_i=-1}\varsigma_i\right\} \tag{18}$$

Constraints still be (15). Let z be a test sample, we have the decision function:

$$g(z) = \sum_{i=1}^{l}\alpha_i y_i K(x_i, z) + b \tag{19}$$

when $g(z) \geq 0$, z belongs to Class 1, else z belongs to Class 2.

VI. CLASSIFIER EVALUATION AND TESTING

A. *Classifier performance evaluation*

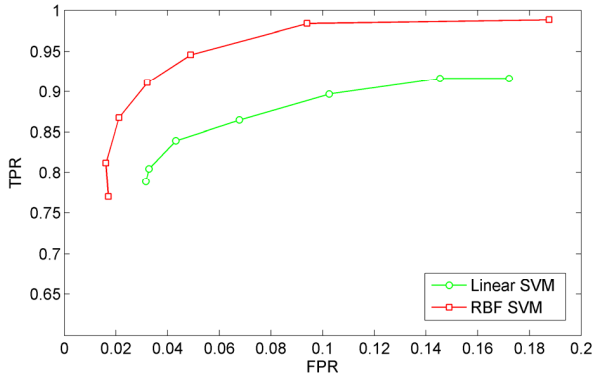
In pedestrian detection, each sample is represented by a pair $D_i = (x_i, y_i)$ comprised a vector x_i denoting HOG features and a class label y_i, $y_i \in \{-1, 1\}$. 1 represents a positive sample, -1 stands for negative samples.

Classifier training process is to obtain an optimal hyperplane as decision function exploiting all training samples. Pedestrian detection is realized according to decision function and HOG features of input images. All the training samples are rescaled to 64×128 pixel, which is also the size of detection window. Each is then divided into 7×15 blocks, with each block containing four cells including 8×8 pixels. Then each sample corresponds to a 3780-dimensional HOG descriptor. In the testing phase, the detection window moves on the image by a step length.

Table I gives four cases of test results for a single image: TP, TN, FP, FN. TN and TP are desirable in our algorithm testing. In the test analysis stage, we introduce ROC curve (Receiver Operator Characteristic curve) to quantify classifier's performance, False Positive Rate (FPR) is the horizontal axis and True Positive Rate (TPR) is the vertical axis. Both are defined below.

$$FPR = \frac{\text{Number of } FP}{\text{Number of total negative samples tested}}$$

$$TPR = \frac{\text{Number of } TP}{\text{Number of total positive samples tested}}$$



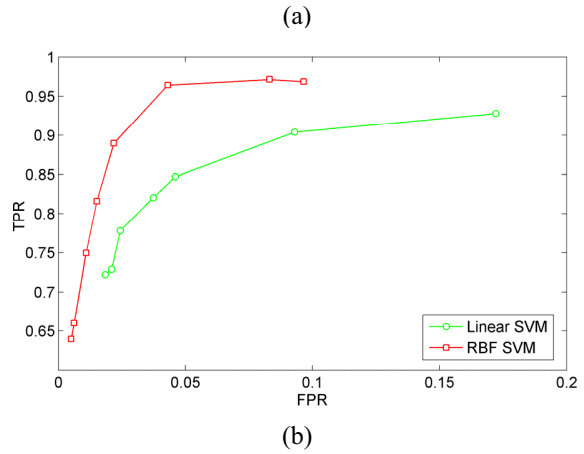


Figure 5. ROC curves of Linear and RBF SVM classifiers trained by different sample configuration.

The closer ROC curve approximates the upper-left corner, the better the classifier is. Fig.5 is the ROC curves based on the complete INRIA testing dataset. Fig. 5(a) is trained by 2400 positive training samples; (b) is trained by 3979 positive training samples. Green and red lines are results of Linear and RBF SVM respectively with the ratio of positive and negative samples ranging from 3:1 to 1:3. Table II lists some detailed data in Fig. 5(a).

TABLE II. SOME DATA CORRESPONDING TO FIG.5(A)

Pos: Neg samples	FPR: Linear SVM	FPR: RBF SVM	TPR: Linear SVM	TPR: RBF SVM
2400:800	0.14533417	0.18757881	0.91651865	0.98845470
2400:1000	0.10687263	0.10655737	0.89253996	0.97158081
2400:1200	0.17213114	0.09394703	0.91651865	0.98401420
2400:1599	0.07093316	0.04413619	0.8579040	0.90674955
2400:1800	0.13272383	0.04886506	0.90674955	0.94582593
2400:2400	0.10245901	0.03215000	0.89698046	0.91119005
2400:2700	0.04319041	0.02427490	0.83925399	0.82948490
2400:3000	0.0825977	0.02112232	0.87122557	0.86767317
2400:3250	0.03436317	0.02080706	0.79573712	0.78411829
2400:3500	0.03278688	0.01733921	0.80461811	0.78952042
2400:3750	0.03152585	0.01702395	0.78863232	0.76998223
2400:3979	0.06778058	0.01607818	0.86500888	0.81172291

From Fig. 5 and Table II, we can conclude that: (1) With the increase of negative training samples, FPR and TPR both decline, which means less false alarms and less correct pedestrian detections. This phenomenon somehow reflects the sample-preference property of SVM. (2) The ROC curve of RBF-SVM is steeper than its linear counterpart, which means the RBF-SVM is more sensitive to training data build-up. (3) The closer ROC curve approximates the upper-left corner, the better the classifier is. Therefore here RBF-SVM outcompetes Linear-SVM. (4) With the same TPR, Linear-SVM owns a higher FPR compared to RBF, demonstrating that Linear SVM is more likely to cause false detection. (5) Compare the curves in Fig. 5(a) and (b), the ROC curve of RBF-SVM in (b) is slightly closer to the upper-left corner than (a), hence we can conclude that the RBF-SVM classifier achieves better performance as total training samples increase.

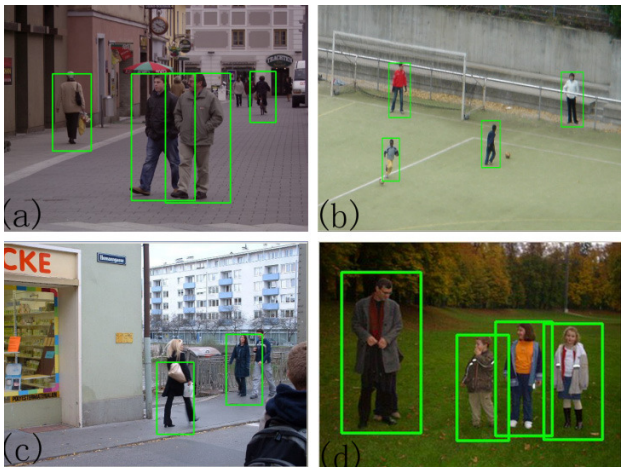


Figure 6. Detection results of part INRIA images

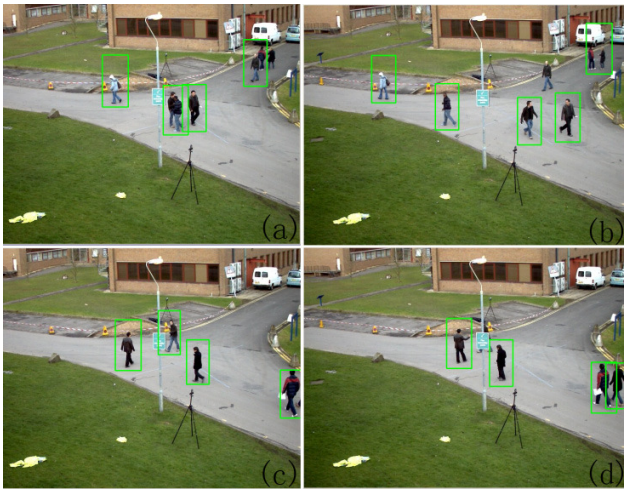


Figure 7. Detection results of PETS2009 video frames

B. *Real video stream detection results*

After the performances of multiple classifiers are evaluated, we adopt RBF-SVM trained with 3979 positive samples and 1989 negative samples as our final classifier. A part of INRIA images and PETS2009 video frames are tested upon this classifier.

Multi-scale, multi-window detection is adopted. We use initial 64×128 window to scan input image in a left to right and top to bottom manner. Window step length is set to 8 pixels. After the whole image is scanned, the input image shrinks with a ratio of 1.1 and we run the detection again until the input image shares the same size with detection window. The above steps will enable those objects bigger than training samples can also be identified. And the object will be considered as pedestrian only if it has been identified for three times at the same position under different scales. The multiple detection results of the same target are merged to avoid repetitive detection labeling. Finally we get the rectangular regions of pedestrian objects. Fig.6 is detection results of some INRIA images. Fig.7 lists some detection results of PETS 2009 dataset. In these images, the pedestrians are bounded with accurate green boxes by adjusting parameters properly.

VII. Conclusion and Future Work

This paper realizes pedestrian detection based on CBBS assisted SVM classifier, and discusses classifier performance with different kernel functions and training samples. The well-trained SVM classifier can basically satisfy real-time detection needs, the speed reaches 10 frames per second at resolution of 320×240 pixels,which can come up to the speed of current mainstream algorithms like GMM and KDE. Furthermore, our algorithm not only extracts the foreground regions in video streams, but also recognizes the pedestrians. Our online detection method provides a good foundation for higher-level computer vision technology such as target tracking, and object understanding in intelligent vision systems.

The background subtraction technique which is employed to enrich the sample data in our paper is not restricted to CBBS, other well-known foreground extraction methods are applicable as well. In the real implementation, the background subtraction process runs separately with the detection process done by the trained classifier. Detection often carries problems of classification error, which is unavoidable. It is natural to validate the SVM classification result with foreground region extracted from the independent background subtraction process. Furthermore, the background subtraction can provide fresh positive and negative training samples online to update the training dataset.

In future work, we will focus on designing an online training scheme for classifier. The streaming sample data is generated from background subtraction process running simultaneously. In this way the pedestrian classifier should be applicable in various scenarios which differs in camera displacement, lighting condition, and can adapt to gradual changes of environment. The fusion of SVM and boosting technique [13] is also a promising direction for our future research. The former is robust against light variation and shows high accuracy, while the latter achieves fast computation speed.

Acknowledgment

This work was supported in part by the U.S. National S cience Foundation under Grant #DBI0939454,in part by N ational Nature Science Foundation OfChina underGrant #6 0674070, and in part by Chinese National Key Technology R&D Program Grant #2008BADA6B01.

References

[1] R. Cutler, L. Davis, "Robust Real-Time Periodic Motion Detection, Analysis and Applications," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 8, pp. 781-796, 2000.

[2] Bertozzi, M. Broggi, A. Cellario, M. Fascioli, A. Lombardi, P. Porta, "Artificial vision in road vehicles," Proceedings of the IEEE, vol. 90, no. 7, pp.1258 -1271, 2002.

[3] C. Stauffer, W. E. L. Grimson, "Learning Patterns of Activity Using Real-time Tracking," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, pp. 747-757, 2000.

[4] A. Elgammal, R. Duraiswami, D. Harwood, L. Davis, "Background and Foreground Modeling Using Nonparametric Kernel Density Estimation for Visual Surveillance," Proceedings of the IEEE, vol. 90, pp.1151-1163, 2002.

[5] Schauland, S.Kummert, A.Su-Birm Park Iurgel, U.Yan Zhang, "Vision-Based Pedestrian Detection-Improvement and Verification of Feature Extraction Methods and SVM-Based Classification," Intelligent Transportation Systems Conference, 2006. ITSC '06, Toronto. IEEE, pp.97-102, 2006.

[6] P.Viola, M.Jones, D.Snow, "Detecting pedestrians using patterns of motion and appearance," Proceedings of International Conference on Computer Vision, Washington DC,USA, pp.734-741, 2003.

[7] D. G. Lowe, "Distinctive image features from scale invariant keypoints," in International Journal of Computer Vision (IJCV), pp.91-110, 2004.

[8] P. Sabzmeydani, G. Mori, "Detecting pedestrians by learning shapelet features," IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Minneapolis, pp.1-8, 2007.

[9] N. Dalal, B.Triggs, "Histograms of oriented graients for human detection," Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, San Diego, CA, USA, vol. 1, pp.886-893, 2005.

[10] K. Kim, T. H. Chalidabhongse, D. Harwood, L. Davis, "Real-time foreground-background segmentation using codebook model," Real-Time Imaging, vol.11, pp. 172-185, 2005.

[11] C.Cortes, V.Vapnik, "Support Vector Network," Machines Learning, vol.20, no. 3, pp.273-297, 1995.

[12] Y. Ma, X. Ding, "Face detection based on cost-sensitive support vector machines," Proceedings of First International workshop on Pattern Recognition with Support Vector Machines. NiagaraFalls, Canada, pp.260-267, 2002.

[13] P.Viola, M. Jones, "Rapid object detection using a boosted cascade of simple features," Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. Marriott, Hawaii, IEEE, pp.511-518, 2001.