Abstract—In this paper, we improve cascade-Adaboost classifier and propose a cascade-Adaboost-SVM classifier combined with Adaboost and SVM and a real-time pedestrian detection system with a single camera. In this paper, we capture the pedestrian candidate areas with a window of fixed size, conduct feature extraction to candidate areas and mobile images with Haar-like rectangle feature calculation and then, complete pedestrian classification by using the proposed cascade-Adaboost-SVM classifier; as this cascade-Adaboost-SVM classifier can adjust numbers of cascade classifiers adaptively, it can construct cascade classifiers effectively based on training set; finally, we complete the pedestrian detection experiment with the database of captured samples and PETS database; the experimental result shows that the cascade classifier proposed by us can get better performance than cascade-Adaboost classifier and its accuracy can reach 99.5% and the false alarm rate is less than 1e-5.

Keywords—Background subtraction; Human detection; Object recognition; Haar-like feature; Ensemble classifier

I. INTRODUCTION

Pedestrian detection based on computer vision act an important role in many applications, for example driver assistance system, surveillance system and intelligent robot [1-3]. The driver assistance system can avoid the collision through pedestrian and route detection to guarantee the safety of drivers and pedestrians effectively; for surveillance system, the false operation caused by shadow changes and swaying of trees can be avoided by using pedestrian detection to activate camera to monitor and record, which can save labor cost and reduce the space to store the videos; secondly, route of pedestrians can be predicted and behavior of pedestrians can be analyzed by tracking pedestrians to prevent accidents actively; if accidents are unavoidable, the prediction and video of pedestrians’ route can help the police know information of suspects quickly to increase the possibility of case solving; for the robot applications, pedestrian detection can be used for interactivity of human-computer interface and can provide information about location of surrounding pedestrians and offer real time service when necessary. It can be known from the description above that the pedestrian detection is being applied to many fields actually.

The real-time pedestrian detection system is currently a hot research subject. Many methods have already been proposed [1-12]. A pedestrian detection system generally includes three parts, namely, object segmentation, feature extraction, and classification. According to the use of camera, they can be divided into visible light camera and invisible light camera; according to camera architecture, they can be divided into single camera and multiple cameras; the cost of visible light equipment is generally much higher than that of visible light camera and multiple camera requires more equipments and the cost is relatively high, therefore we mainly discuss the pedestrian detection method using single visible light camera in this paper. Based on whether or not to adopt the background image method, object segmentation can be classified roughly into two methods. The first method is to use the background image [1-3, 11], which helps us to get the foreground image by subtracting the background image from the real-time input image. Then, it segments the object according to the connected parts on the foreground image. Finally, it performs feature extraction and classification identification. This method needs background reconstruction and technology update. This is unsuitable for mobile platforms, such as driving safety assistance systems and intelligent robots. The second is not to adopt the background image method [1-9, 12], which segments the input image into some image blocks with the same window size. Then, it makes feature extraction to all image blocks. Finally, it identifies whether the blocks belong to a pedestrian according to the extracted feature. The advantage of this method is that it does not need extra time for reconstruction and update of the background image. This is suitable for mobile platforms.

Viola and Jones [12-14] used Haar-like method for the local feature of the pedestrian shown in the image blocks. Moreover, they took an Adaboost classifier [15-16], which has many weak classifiers forming a powerful one, to identify the pedestrians. Each weak classifier makes the classification for only one dimension within the feature vector. For a few weak classifiers, this method can effectively reduce the complexity and time of the calculation, but the accuracy obtained is low. For many weaker classifiers, it provides high accuracy but requires a long calculation time and still has high false alarm rates. To solve the problem, Viola and Jones further proposed cascade Adaboost classifier which is composed by the cascading of many Adaboost classifiers; assume there are $L$ layers in this classifier and the detection rate and false alarm rate of each layer are $d_i$ and $f_i$ respectively, and then the detection rate and the false alarm rate of the whole cascade-Adaboost classifier can be defined as $D=\prod_{i=1}^{L} d_i$ and $F=\prod_{i=1}^{L} f_i$ respectively. For example, if the detection rate $d_i$ and the false alarm rate $f_i$ are set as 99.9% and 30.0% respectively.
in all layers, then the whole detection rate $D$ and the whole false alarm rate $F$ will be $(0.999)^{10} > 0.99$ and $(0.3)^{10} < 1e-5$, respectively, for training set. However, when we made the cascade-AdaBoost classifier, we found that AdaBoost classifiers in front layers could reach preset targets with less weak classifiers but those in rear layers need more weak classifiers because the training set would remove some negative samples when passing through each layer of AdaBoost classifier; with the increase of layers, the samples of the remaining training set became less and similar, so more difficult negative examples were used for training in later layers, and more weak learners were usually chosen to satisfy the goals in the later layers. To solve this problem, Yu-Ting Chen etc [17] proposed a novel cascade approach that could exploit both the stage-wise and the cross-stage information. In their approach, some meta-stage classifiers were added to the cascaded classifier to utilize interstage information and learn new classification boundaries to enhance the detection performance. This method could reduce numbers of weak classifiers used by AdaBoost classifier effectively, but it required more complicated calculations. Xian-Bin Cao etc [18] proposed substituting AdaBoost classifier of the last layer of cascade-AdaBoost classifier with SVM, but the structure remained unchanged; in other words, the structure of cascade classifier could not adjust numbers of SVMs adaptively, so in this paper, we proposed a cascade classifier combining Adaboost with SVM, called cascade-AdaBoost-SVM classifier, and modified the training algorithm of cascade-AdaBoost classifier proposed by Viola and Jones to make it suitable for constructing a cascade-AdaBoost-SVM classifier; at the very start, the algorithm set the lowest detection rate, the highest false alarm rate and the maximum number of weak classifiers of each layer of Adaboost classifier; when Adaboost classifier of each layer could not achieve the preset performance under the predetermined maximum number of weak classifiers, substitute this Adaboost classifier with SVM and perform SVM training based on the feature dimensions selected by Adaboost classifier without calculating all dimensions; in this way, the SVM training could be completed more effectively and quickly.

Finally, apply the cascade-AdaBoost-SVM classifier proposed by us to the real time pedestrian detection system under the architecture of the single camera and compare the result with that of cascade-Adaboost classifier; it could be found that higher accuracy rate and lower false alarm rate could still be obtained with less cascade layers.

II. BACKGROUND

As mentioned in the paragraphs above, we proposed the cascade classifier combined with Adaboost and SVM and modified the training algorithm of cascade-AdaBoost classifier of Viola and Jones to make it suitable for constructing a cascade-AdaBoost-SVM classifier, so we will first introduce briefly Adaboost algorithm, cascade-AdaBoost algorithm and SVM in sequence in this section.

A. AdaBoost Algorithm

Adaboost classifier initially proposed by Freund et al [15-16] is a ensemble classifier composed of many weak classifiers(such as linear classifier); each classifier performs classification according to only one dimensionality of input vector, so it is also called weak classifier; the result of ensemble classifier can be expressed as:

$$H(x) = \text{sign}\left[\sum_{t=1}^{T} \beta_t h_t(x)\right]$$

(1)

where $x$ represents input vector; $h_t(x)$, $t = 1, ..., T$ means the number of classifiers is $T$; $\beta_t$, $t = 1, ..., T$ refers to weight of each weak classifier; Fig. 1 shows Adaboost algorithm; suppose a training sample set $\{x_i, y_i\}$, $i = 1, ..., m$, where $x_i \in R^r$, $y_i \in \{-1, 1\}$ is given and first initialize the initial distribution values of all training samples; to make the classification results maintain a higher detection rate, let the distribution value of a positive sample be equivalent to that of all negative values; if the training sample set includes $p$ positive samples and $q$ negative samples, that is, $m = p + q$, we set the distribution value of the positive sample and that of the negative sample as $1/(p+1)$ and $1/q(p+1)$ respectively and then perform the selection loop to $T$ weak classifiers; each time the algorithm performs the loop, it first searches for weak classifiers $h_t(x)$, $j = 1, ..., n$, with minimum error according to each dimension and then finds out the weak classifier $h_t(x)$ with minimum error from these weak classifiers; after the selection of weak classifiers at $t$th loop, readjust the probability value of the training sample so that samples of classification error in this loop can have the priority when the next loop is performed; then calculate the corresponding weight $\beta_t$ of the weak classifier $h_t(x)$; finally, the strong classifier $H(x)$ can be obtained by calculating sum of products of weight of $T$ weak classifiers and corresponding $\beta$ values; determine the classification result of the input vector with result of ensemble classifier.

- Given example images $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$, where $x_i \in R^r$, $y_i \in \{-1, 1\}$, for negative and positive example respectively.
- Initialization distribution $\omega_1(i) = \sqrt{1/(p+1)}$ for $y_i = -1$, $1$, respectively where $p$ and $q$ are the number of negatives and positives respectively.
- For $t = 1, ..., T$
  - Find classifier $h_t : x \rightarrow \{-1, 1\}$, $h_t = \text{argmin}_{\epsilon_j} \epsilon_j$, where $\epsilon_j = \sum_{i=1}^{n} \omega_i(t)[y_i \neq h_t(x_i)]$.
  - Weight classifier: $\beta_t = 0.5 \ln \left(\frac{1-\epsilon_t}{\epsilon_t}\right)$.
  - Update distribution: $\omega_t(t+1) = \frac{\omega_t(t)\exp[-\beta_t y_t h_t(x_t)]}{Z_t}$, $Z_t$ is for normalization.
- Output final classifier: $H(x) = \text{sign}\left[\sum_{t=1}^{T} \beta_t h_t(x)\right]$. 

Figure 1. AdaBoost Algorithm

B. Cascade AdaBoost Algorithm

To further reduce the false alarm rate, Viola and Jones [12-14] proposed a Cascade-AdaBoost classifier, the architecture of which is shown in Fig. 2, where each classifier is Adaboost classifier; $\text{Nega}$ represents that they are determined to be negative examples by the input vector; $P$ is
represents that they are determined to be positive examples by the input vector; \( x \) stands for input vector; if the input vector determines Adaboost classifier to be negative, the classifier will be removed from the training sample set and will not enter the next layer; so the number of samples decreases with the increase of layers; in this way, negative samples are removed quickly; if it is determined to be positive, the training sample will enter the next layer to continue the classification until the last layer.

\[ f(x) = w^T x + b, \]  

(2)

where \( w \) is the normal vector of this hyperplane, \( x \) is the input vector, \( -b/\|w\| \) is the distance from the origin perpendicular to the hyperplane. If \( w \) is a unit vector, this distance is \(-b\). Thus, \( w \) and \( b \) are the parameters that we search. SVM solution is based on maximum margin and minimum square error, so the objective function can be defined as follow:

\[ E(w,b) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{m} \alpha_i (y_i f(x_i))^2, \]  

(3)

where \( \alpha = [\alpha_1, ..., \alpha_m] \) is the coefficient of Lagrange and \( \alpha_i > 0, i = 1, 2, ..., m \). Maximizing the eqn. (3), the following equation can be obtained:

\[ w = \sum_{i=1}^{m} \alpha_i y_i x_i, \text{ and } \sum_{i=1}^{m} \alpha_i y_i = 0. \]  

(4)

Put the above equation into eqn. (3); the original objective function is converted into dual objective function, and the objective function can be redefined as

\[ Q(w) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j K(x_i, x_j), \]  

(5)

It also satisfies

\[ \sum_{i=1}^{m} \alpha_i y_i = 0 ; 0 \leq \alpha_i \leq C \forall i = 1, 2, ..., m, \]  

(6)

where \( \alpha = [\alpha_1, ..., \alpha_m] \) is the coefficient of Lagrange, \( K(x_i, x_j) \) is a kernel function, \( m \) is the number of training samples and \( C \) is an adjustable positive parameter. For separable linear problems, \( C \) value is infinite. On the contrary, for non-separable linear problems, \( C \) value is a positive integer. This equation is a quadratic equation, so it can be solved by using the optimization algorithm. By putting the obtained value of \( \alpha = [\alpha_1, ..., \alpha_m] \) into eqn. (4), the \( w \) vector is obtained. By putting \( w \) into eqn. (2), \( b \) value can be obtained. \( K(x_i, x_j) \) is a kernel function of positive number. In this paper, we use radial basis function (RBF) as the kernel function, which is defined as the following:

\[ K(x, x_i) = \exp\left(-\frac{\|x - x_i\|^2}{\sigma^2}\right), \]  

(7)

where \( \sigma \) acts as a variance parameter.

### III. The PROPOSED CASCADE CLASSIFIER

In Section I, we mentioned that because in cascade-Adaboost classifier, Adaboost classifier in front layers could reach preset targets with less weak classifiers; but with the increase of layers, the samples of the remaining training set became less and similar; Adaboost classifier in rear layers need linear combination of more weak classifiers to reach preset targets, which is easy to cause overfitting and time-consuming;
so in this paper, we propose a cascade classifier combined Adaboost and SVM, which we call cascade-Adaboost-SVM classifier; because SVM can solve the linearly inseparable problem, substitute Adaboost classifier behind cascade classifier with SVM; modify the cascade-Adaboost classifier training algorithm proposed by Viola and Jones to make it suitable for constructing a cascade-Adaboost-SVM classifier which is different from the cascade classifier combined Adaboost and SVM. The cascade classifier proposed by us can increase Adaboost classifier or SVM adaptively.

Fig. 4 shows the training algorithm of cascade-Adaboost-SVM classifier; during the training of cascade classifier, first set the detection rate \(d\), the false alarm rate \(f\), the maximum number of weak classifiers and the target \(F_{target}\) of the overall false alarm rate of classifiers; like the training algorithm used for constructing a cascade-Adaboost classifier, this training algorithm is composed of two loops; when the internal loop constructs a Adaboost classifier, it determines whether the number \(n_i\) of weak classifiers is greater than the maximum value \(n_{ak}\); if it is, substitute Adaboost classifier of this layer with SVM and train SVM with dimension of input vector selected by this Adaboost classifier without calculating all dimensions of input vector; in this way, the training of SVM can be completed more effectively and quickly; then, determine whether the overall false alarm rate of the present cascade-Adaboost-SVM classifier can satisfy the condition of external loop; if it can, train Adaboost classifier of the next layer with the rest negative examples and positive examples; otherwise, terminate the training of cascade-Adaboost-SVM classifier.

1. User selects values for \(f\), the maximum acceptable false alarm rate per layer and \(df\), the minimum acceptable detection rate per layer.
2. User selects target overall false alarm rate, \(F_{target}\).
3. User selects values for \(n_{ak}\), the maximum acceptable number of weak classifiers.
4. \(P\) is set of positive examples.
5. \(N\) is set of negative examples.
6. \(F_0 = 1; D_0 = 1; i = 0;\)
7. While \((F_i > F_{target})\)
   - \(i = i + 1; n_i = 0; F_i = F_{i-1};\)
   - \(n_i < n_{ak}\)
     - \(n_i = n_i + 1;\)
     - Use \(P\) and \(N\) to train a classifier with \(n_i\) features using Adaboost.
     - Evaluate current cascade classifier on validation set to determine \(F_i\).
   - if \((n_i >= n_{ak})\)
     - Use \(P\) and \(N\) to train a classifier with \(n_i\) features using SVM.
     - Evaluate current cascade classifier on validation set to determine \(F_i\).
8. \(N\) is NULL.
9. If \(F_i > F_{target}\) then evaluate the current cascaded classifier on the set of negative examples and put any false detections into the set \(N\).

**Figure 4.** The training algorithm for building a cascade classifier using the Adaboost and SVM

**IV. PEDESTRIAN DETECTION**

This section will introduce the process of pedestrian detection and classify the candidate areas of pedestrians with the cascade-Adaboost-SVM classifier proposed by us. The process includes candidate area segmentation, feature extraction and classification; they will be described separately below. After the image input, segment the candidate area first; the segmentation method used in this paper is shielding with windows of fixed size; besides, to use the motion feature, we move the candidate area one pixel up, down, left and right respectively and then calculate the difference between the former candidate area and the new one; Fig. 5 shows an example of candidate area; the leftmost one is an image of candidate area and from left to right, they are the images of difference between the former image and the images that have been moved one pixel up, down, left and right respectively. Next, perform feature extraction to image of difference between the former image and the images that have been moved.

**Figure 5.** Candidate region, images of difference between the former image and the images that have been moved up, down, left and right one pixel respectively (left to right)

The feature extraction method in this paper uses rectangular features of Haar-like; as shown in Fig. 6, it is an feature extraction method for image and object identification proposed by Viola and Jones [12-14]; it is an local block feature extraction method and is easy to extract; real time calculation can be finished by combining with the f Integral image calculation method proposed by them; Four Haar-like shielded rectangular features are used in this paper; shield the images of the candidate area above with the four shielding images of the candidate area from up to down and from left to right; feature information can be obtained through subtraction of sum of rectangle grey scale of black block and white block, and so forth. In addition, to detect pedestrians of different sizes, we used different scales based on rectangle feature of each Haar-like and perform pedestrian detection to the candidate area with the cascade-Adaboost-SVM classifier in this paper.

**Figure 6.** Rectangle feature of Haar-like.

**V. EXPERIMENTAL RESULTS**

This section will introduce the sample sets used in the experiment and the related experimental results; these sample sets include self-built samples and samples of PETs database; first we will introduce these samples and then present the experimental numerical results.

**A. Datasets**

In samples built in this paper, there are a total of 3,000 samples used by us in the experimental sample set, including 500 samples of pedestrians and 2,500 samples of non-pedestrians in whole dataset. Fig. 7 shows part of the sample set; Fig. 7(a) shows samples of pedestrians, including the front, back and profile of pedestrians; Fig. 7(b) shows
samples of non-pedestrians, including trees, vehicles, traffic signs, incomplete pedestrians, flags, motorcycles, bicycles, telegraph poles, flagpoles, background and etc. The size of each sample is 15x36 pixels. Besides, we also used 3,000 samples of the PETs database [20]; the experimental samples were videos of the sports field, so the samples of non-pedestrians were uncomplicated; but the pedestrians were walking or running, so the postures were complicated; in this experiment, we used 500 samples of pedestrians and 2,500 samples of non-pedestrians; Fig. 8 shows some pedestrian samples from the PETs database; the size of each sample is also 15x36 pixels.

![Image](image_url)

**Figure 7.** Part pedestrian samples (a) pedestrian samples, (b) non-pedestrian samples

![Image](image_url)

**Figure 8.** Part training samples of PETs database (a) pedestrian samples, (b) non-pedestrian samples

### B. Numerical Results

To assess the detection result in the experiment, we defined the accuracy rate (AR), detection rate (DR) and false alarm rate (FAR) respectively as below for the assessment of result efficiency:

\[
AR = \frac{TP + TN}{P + N} \times 100 \%,
\]

\[
DR = \frac{TP}{P} \times 100\%,
\]

\[
FPR = \frac{FP}{N} \times 100\%,
\]

where \( P \) and \( N \) stands for the set number of pedestrians samples and non-pedestrians samples; \( TP \) stands for the number of pedestrians samples detected to be pedestrians; \( TN \) stands for the number of non-pedestrians samples detected to be non-pedestrians; \( FP \) stands for the number of non-pedestrians samples detected to be pedestrians; so it can be known from the definitions above that higher detection rate (DR) and false alarm rate (FAR) can get the best results rather than higher accuracy rate (AR). In the experiment, we verified the performance of the cascade classifier by means of the 5-fold cross-validation and compared it with that proposed by previous scholars; five experiments were conducted in total; RBF (Radial Basis Function) kernel function was used for SVM; set the parameter value \( C \) as 10 and set variance as 0.5; set the detection rate \( d \) and false alarm rate \( f \) of Adaboost classifier of each layer as 0.999 and 0.6 respectively.

For our database extraction experiment, cascade-Adaboost classifier used two layers of Adaboost classifier; the first layer used 1 weak classifier and the second layer used 6~8 weak classifiers, so we set the maximum number \( n_{ab} \) of weak classifiers as 5; the cascade-Adaboost-SVM classifier proposed by us had two layers of classifiers, too, the first layer being Adaboost classifier and the second layer being SVM; the experimental data result of the test is shown in table 1, from which we can find that the accuracy rate (99.9%) of the cascade-Adaboost-SVM classifier proposed by us is a little higher than that (99.63%) of the cascade-Adaboost classifier; but the difference is slight mainly because the identification of the pedestrian samples and non-pedestrian samples is high, so the results are both good; besides, we conducted the same experiment to PETs database; during the experiment, cascade-Adaboost classifier used 7 layer of Adaboost classifiers, the average number of weak classifiers being 1, 5.8, 7.6, 11, 12.2, 15.8 and 28.6 for each layer; so we set the maximum number \( n_{ab} \) of weak classifiers as 10; the cascade-Adaboost-SVM classifier proposed by us used 3~4 layers of classifiers and only the last classifier in the whole cascade-Adaboost-SVM classifier was replaced by SVM; the test result of the experimental data is also shown in table 1, from which we can find that the accuracy rate (99.58%) of the cascade-Adaboost-SVM classifier proposed by us is a little higher than that (96.9%) of the cascade-Adaboost classifier; compared with the database samples captured by us, our method has higher accuracy rate among the PETs database samples, which is mainly because samples of PETs database was captured during the football match, in which the incomplete pedestrian samples are in the majority of the non-pedestrian samples except for the greenward; so the identification of non-pedestrian and pedestrian samples is not so good; it is not easy to separate pedestrian samples from non-pedestrian samples; that's why the cascade-Adaboost classifier needs 7 layers of Adaboost classifier to complete the classification and the last layer needs nearly 29 weak classifiers. The above analysis shows that the cascade-Adaboost-SVM classifier proposed by us could get good results for both databases.

<table>
<thead>
<tr>
<th>Classifiers</th>
<th>Datasets</th>
<th>Our</th>
<th>PETs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cascade-Adaboost classifier</td>
<td>AR(%)</td>
<td>99.63</td>
<td>96.93</td>
</tr>
<tr>
<td></td>
<td>DR(%)</td>
<td>99.00</td>
<td>95.80</td>
</tr>
<tr>
<td></td>
<td>FPR(%)</td>
<td>0.48</td>
<td>2.84</td>
</tr>
<tr>
<td>Cascade-Adaboost-SVM classifier</td>
<td>AR(%)</td>
<td><strong>99.90</strong></td>
<td><strong>99.53</strong></td>
</tr>
<tr>
<td></td>
<td>DR(%)</td>
<td>99.40</td>
<td>97.20</td>
</tr>
<tr>
<td></td>
<td>FPR(%)</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

On the other hand, if substituting the pedestrian detection method of cascade-Adaboost-SVM classifier with the single SVM, the use of SVM has a problem, namely, the solution-searching process of SVMs requires solving a quadratic optimization. Therefore, though the use of SVM to solve the problems of small samples with non-linear and high dimension mode classification may show some special advantages, for large training samples, SVM will take much more time for calculations. So the single SVM is unsuitable for
the classification of large training samples [21-23]. In the cascade classifier proposed by us, more than 80% training samples were removed from classifiers, so the training samples used by SVM remained about 20% of the original training samples; therefore, the cascade classifier proposed by us can also solve the problem of the single SVM using large sample set.

Table 2 shows the comparison of training time and number of supporting vectors between cascade classifier proposed by us and the single SVM using two different databases mentioned above; based on our experiment of database extraction, SVM in the cascade classifier proposed by us needs 423 supporting vectors on average, while all 2,400 training samples are supporting vectors for the single SVM, which directly affects the time taken by the test; for the training time, if the time for construction of Adaboost classifier in cascade classifier is not considered, the training time of the cascade classifier proposed by us is about 62.98 sec, far below 1,639.45 sec needed by the single SVM. Results are similar for PETs database.

Table 2. Comparison of training time and number of supporting vectors between cascade classifier proposed by us and SVM.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Our cascade classifier</th>
<th>SVM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Number of SVs</td>
<td>Time/sec</td>
</tr>
<tr>
<td>Our</td>
<td>423</td>
<td>62.98</td>
</tr>
<tr>
<td>PETs</td>
<td>646</td>
<td>100.10</td>
</tr>
</tbody>
</table>

VI. CONCLUSIONS

In this paper, we proposed a cascade classifier combining Adaboost with SVM, called cascade-Adaboost-SVM classifier, and proposed a training algorithm; because the calculation of Adaboost classifier is simple and SVM has better effects on classifying linearly non-separable problems, the cascade-Adaboost classifier can be improved by combining the advantages of the two classifiers; for the experiment, we used two samples of databases for pedestrian detection and training; it can be known from the experimental result that the cascade-Adaboost-SVM classifier proposed by us is proven fine for both data bases; in addition, the cascade-Adaboost-SVM classifier proposed by us can also solve the time-consuming problem of SVM when applying to large training sample set.

ACKNOWLEDGMENT

For the results of this paper, we show our gratitude to the support from the Program Institute of National Science Council, the number is NSC-99-2632- E-324 -001-MY3.

REFERENCES