Contents lists available at SciVerse ScienceDirect



Engineering Applications of Artificial Intelligence



journal homepage: www.elsevier.com/locate/engappai

A self-constructing cascade classifier with AdaBoost and SVM for pedestrian detection

Wen-Chang Cheng*, Ding-Mao Jhan

Department of Computer Science and Information Engineering, Chaoyang University of Technology, 168 Jifong E. Rd., Wufong District, Taichung 41349, Taiwan

ARTICLE INFO

ABSTRACT

Article history: Received 14 February 2012 Received in revised form 9 July 2012 Accepted 28 August 2012 Available online 19 September 2012

Keywords: Surveillance system Driver assistance system Combination classification Haar-like features HOG features Large-scale training sample problem In this paper, we propose a cascade classifier combining AdaBoost and support vector machine, and applied this to pedestrian detection. The pedestrian detection involved using a window of fixed size to extract the candidate region from left to right and top to bottom of the image, and performing feature extractions on the candidate region. Finally, our proposed cascade classifier completed the classification of the candidate region. The cascade-AdaBoost classifier has been successfully used in pedestrian detection. We have improved the initial setting method for the weights of the training samples in the AdaBoost classifier, so that the selected weak classifier would be able to focus on a higher detection rate other than accuracy. The proposed cascade classifier can automatically select the AdaBoost classifier or SVM to construct a cascade classifier according to the training samples, so as to effectively improve classification performance and reduce training time. In order to verify our proposed method, we have used our extracted database of pedestrian training samples, PETs database, INRIA database and MIT database. This completed the pedestrian detection experiment whose result was compared to those of the cascade-AdaBoost classifier and support vector machine. The result of the experiment showed that in a simple environment involving campus experimental image and PETs database, both our cascade classifier and other classifiers can attain good results, while in a complicated environment involving INRA and MIT database experiments, our cascade classifier had better results than those of other classifiers.

© 2012 Elsevier Ltd. All rights reserved.

1. Introduction

Pedestrian detection based on computer vision acts an important role in many applications, for example driver assistance system, surveillance system and intelligent robot (Gandhi and Trivedi, 2007; Ma et al., 2009; Enzweiler and Gavrila, 2009; Geronimo et al., 2010; Nam et al., 2011). 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 memory to store the videos. Secondly, route of pedestrians can be predicted and behavior of pedestrians can be analyzed by tracking pedestrians, so as 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 actually applied to many fields.

A classification-based pedestrian detection system generally includes three parts, namely, candidate region segmentation, feature extraction, and pedestrian 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 invisible light equipment is generally much higher than that of visible light camera. Multiple cameras require more equipment 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, candidate region segmentation can be classified roughly into two methods. The first is to use the background image, which helps us to get the foreground image by subtracting the background image from the real-time input image. 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

^{*} Corresponding author. Tel.: +886 4 23323000x5208; fax: +886 4 23742375. *E-mail address:* wccheng@cyut.edu.tw (W.-C. Cheng).

^{0952-1976/\$-}see front matter @ 2012 Elsevier Ltd. All rights reserved. http://dx.doi.org/10.1016/j.engappai.2012.08.013

а

adopt the background image method, which segments the input image into some candidate regions with the same window size (such as sliding windows). 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. After completing the segmentation of the candidate region, feature extraction follows. In this paper, we have applied three kinds of commonly-used feature extraction methods for pedestrian detection; namely, edge detection, Haar-like feature (Viola and Jones, 2005), and histograms of oriented gradients (HOG) (Dalal and Triggs, 2005). Finally, we have identified whether the candidate regions belonged to pedestrians based on the extracted features.

Classifier performance is a very important factor in classificationbased pedestrian detection and thus it is a widely discussed subject (Xu et al., 2011; Dollar et al., 2012). The cascade-AdaBoost classifier (Viola and Jones, 2005) is one of commonly used classifiers. However, 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 samples were used for training in later layers, and more weak learners were usually chosen to satisfy the goals in the later layers. In this paper, we proposed a new self-constructing cascade classifier combining AdaBoost classifier with SVM for pedestrian detection, called self-constructing cascade-AdaBoost-SVM classifier, and modified the training algorithm of cascade-AdaBoost classifier 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 laver of AdaBoost classifier: when AdaBoost classifier of each laver 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. On the other hand, the cascade classifiers of the first layers removed most of the training samples, and thus the problem of long training time consumed by SVM classifiers for large samples was solved. However, the proposed self-constructing cascade classifier improves the problems of the original cascade-AdaBoost classifier. It also improves the time consumption of SVM at that time when it is applied in large-scale training sample sets. Finally, we have modified the original weight setting method for the AdaBoost classifier's training samples in order to maintain the highest detection rate of each layer's cascade classifier. With this, the highest detection rate would still be maintained after each selection of the weak classifier.

The content of this paper is summarized as follows: the second section is literature review; the third section introduces the feature extraction methods for the candidate regions; the fourth section introduces our proposed self-constructing cascade-AdaBoost-SVM classifier combining AdaBoost classifier and SVM; the next section shows the experimental results and the last section presents the conclusion

2. Literature reviews

So far, many sliding window classification-based pedestrian detection methods have already been proposed. The classifiers used in the literature can be categorized into two groups: single-classifier-based approaches and classifier ensemble-based approaches (Xu et al., 2011). Many single classifier approaches based on neural networks (NNs) (Gavrila, 2000; Sessler et al., 2001; Szarvas et al.,

2006) and support vector machines (SVMs) (Cheng et al., 2005; Tian et al., 2005; Xu et al., 2005; Chen et al., 2006; Schauland and Kummert, 2007; Liue and Fuentes, 2009) have been proposed. Gavrila (2000) trained a single classifier based on RBF-NNs, whose detection rate is 85% and false alarm rate is 10%. Szarvas et al. (2006) have proposed a pedestrian detection method using an artificial neural network, which combined radar sensor and neural network in their system. The experimental results showed that the accuracy rate reached more than 90%, but the system required radar sensors to capture the candidate region, thus increasing the system cost, Xu et al. (2005) proposed a single SVM classifier for an infrared camerabased pedestrian detection system, whose detection rate is 26%–94% and false alarm rate is 2.6%. Liue and Fuentes (2009) have proposed the use of SVM in pedestrian detection, which initially applied principal component analysis (PCA) to conduct feature extraction on the training samples, and used SVM for classification. The experimental result showed that the accuracy rate of this method reached about 90%. In addition, since SVM consumed a lot of time in the classified training of large-scale training sample sets and was difficult to converge, SVM was generally used in the classification of smallscale training sample sets. Based on the above information, the detection rate was still not high enough even if the pedestrian detection results obtained through the neural network or SVM could reach more than 90% of the accuracy rate.

In the application of surveillance system and driver assistance system, an extremely high detection rate and very low false alarm rate were expected in order to avoid dangerous situations caused by lapses in pedestrian detection and to avoid unnecessary occurrences due to many false alarm cases. Therefore, related studies on classification have been conducted in recent years; particularly face detection or pedestrian detection which analyzes the combination of multiple classifiers (Viola and Jones, 2004; Zhang et al., 2006). By combining similar or different kinds of classifiers (Zhang et al., 2006; Chen and Chen, 2004; Sahbi et al., Dec. 2006; Li et al., 2008; Wang et al., 2010), the performance of a single classifier can be improved in order to achieve a higher detection rate and lower false alarm rate. The combined architecture of these classifiers, also known as classifier ensemble-based approach, is differentiated into three types: parallel architecture (Viola and Jones, 2004; Grubb et al., 2004; Tsai et al., 2010; Cheng and Cheng, 2010; Hiromoto et al., 2009), cascade architecture (Viola and Jones, 2005; Xu et al., 2005; Ma and Ding, 2002; Kukenys and McCane, 2008; Ding et al., 2009) and mixture architecture (Xu et al., 2011; Wei et al., 2009), which are discussed and described below.

Fig. 1(a) shows the architectural chart of the general parallel classifier, e.g. AdaBoost classifier, multiple parallel SVM or linear



Fig. 1. Combination of multiple classifiers (a) classifier's parallel architecture diagram (b) classifier's cascade architecture diagram.

SVM, in which the parallel combination of *T* sets of classifiers is included; **x** is the input vector and β is the weight value of each classifier. Therefore, the output can be expressed as the sum of the product between individual output of T classifiers and its weight value. Since this parallel classifier shows the example of two categories, the final output result is either True or False. Grubb et al. (2004) trained two SVM classifiers to detect pedestrians from front/rear and side poses separately, and the final decision is made by fusing the two results. Tsai et al. (2010) have proposed the use of many SVMs' parallel connection in pedestrian detection. They used the HOG (Dalal and Triggs, 2005) feature extraction method to extract the features of the candidate region and to carry out the classification of the SVM parallel combination. Each SVM had the same weight value and the experimental result could reach more than 92% of the detection rate and about 4% of the false alarm rate. Viola and Jones (2004) have completed the pedestrian detection using the AdaBoost classifier which is a kind of parallel classifier originally proposed by Freund and Schapire (1996,1997). The AdaBoost classifier is a linear combination of many classifiers in which each classifier only focuses on one dimension's classification of the input feature vector. Thus, each classifier is known as a weak classifier. In the latest published surveyed paper of Dollar et al. (2012) 16 types of state-of-the-art pedestrian detection methods (published from 2004 to 2010) are selected to conduct efficiency evaluation. Nearly all modern detectors employ some forms of HOG feature. In addition, detectors can utilize gradients directly, Haar-like, color, texture, selfsimilarity and motion feature. Most of the classifiers of these methods adopt AdaBoost and linear SVM and a few of them adopt latent SVM and Hik-SVM. The experiments include 6 types of pedestrian databases, which is divided into near, middle, and remote groups for experiments. Since each method uses different features, it is actually more like an evaluation of the combination of different input features. For the case without considering the motion feature, the best overall performing detector is CHNFTRS (Dollar et al., 2009) and FPDW (Dollar et al., 2010), and these two methods both use HOG, gradients, grayscale (Haar-like feature) and AdaBoost classifier. The worst one (Viola and Jones, 2004) is the method using Haar-like feature and AdaBoost classifier. During the construction of the AdaBoost classifier wherein a new weak classifier was added, the minimum error was used to calculate the weight of this weak classifier. At the same time, each training sample's weight was readjusted before it was passed to the next newly added weak classifier. Based on the newly added weak classifier, the effect of the overall parallel classifier was gradually improved. The experimental result showed that a continuous increase in the number of weak classifiers would effectively improve the overall detection rate of the parallel classifier, but there was still a high false alarm rate. Another problem was that each classifier used all samples (i.e. large number of training samples) for training, which was time-consuming during the entire training process. Therefore, some scholars have proposed to use the cascade combination of many classifiers to address the above problems.

Fig. 1(b) shows the architectural chart of the cascade classifier which includes classifiers with a total of *L* layers; **x** is an input vector. Assuming that it is an example of two kinds of classifications, the output of each layer's classifier is either *True* or *False*. So when the output of each layer of classifier is *True*, then the classifier of the next layer would continue the classification. If the output of the classifiers of the last layer is still *True*, it means that the classifiers is *True*. If the result of any layer's classifier is *False*, it would determine in advance that the classification result of this input vector in the cascade classifiers is *False*. The main purpose of the cascade classifier is to reduce the false alarm rate by using the

cascade combination of many classifiers. Assume there are L layers in this cascade 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 classifier can be defined as $D = (d_i)^L$ and $F = (f_i)^L$ respectively. For example, there are 10 layers in this cascade classifier. If the detection rate d_i and the false alarm rate f_i are set as 0.99 and 0.3 respectively in all layers, then the whole detection rate D and the whole false alarm rate F will be $(0.99)^{10} > 0.9$ and $(0.3)^{10} < 1e-5$, respectively. It can be observed that the cascade classifier can reduce the overall false alarm rate as well as the overall detection rate, therefore classifiers in each laver must maintain the highest detection rate while reducing the false alarm rate. Ma and Ding (2002) have also used the cascade combination of multi-layered SVM in face detection. They arranged a six-layer SVM and the first five layers used a linear kernel function while the last layer used a nonlinear kernel function. The experimental result showed that this method achieved a detection rate of 88.9% given a false alarm rate of below 7e-6. Xu et al. (2005) trained a three-layer SVM classifier for an infrared camera-based pedestrian detection system, and experimental results showed that the classifier ensemble achieved only a little lower false alarm rate and almost the same detection rate. Ding et al. (2009) have used the cascade combination of many SVMs in eye detection, which used 20×20 grayscale image samples with 400 dimensions as input feature vectors. The classification was conducted based on a cascade SVM. The experimental result showed that this method can achieve an accuracy rate of 88%. Kukenys and McCane (2008) have applied the cascade combination of a two-layer SVM in pedestrian detection which used pedestrian heads and shoulders as training samples in pedestrian detection, and conducted feature extraction based on HOG. The experimental result showed that this method achieved an accuracy rate of 90% given a false alarm rate of below 1e-4. In the training process of the cascade classifier, the number of training samples were reduced layer by layer, thus the classifier of each layer did not need all samples for training. Moreover, the consumed training time of the SVM cascade classifier was 30 times faster than that of a single nonlinear SVM classifier. Viola and Jones (2005) have also used a cascade classifier combining many AdaBoost classifiers in face detection and pedestrian detection; this was referred to as the cascade-AdaBoost classifier. They used Haar-like feature for feature extraction, and there were 4916 face samples and more than three million non-face samples. The cascade classifier had 38 cascade layers and could achieve a detection rate of about 93.7% given a false alarm rate of about 1e-7. The same method was also used in detecting pedestrians, applying Haar-like feature, as well as mobile and appearance features. The experimental result showed that it could achieve a detection rate of about 90% given a false alarm rate of 1e - 6.

Xu et al. (2011) and Wei et al. (2009) proposed a tree classifier for pedestrian detection. It can be regarded as the mixed ensemble classifier. The author compared the approach to others based on two widely used classifiers in the same benchmark environment; these are AdaBoost (Viola and Jones, 2005; Cao et al., 2008) and SVM (Cheng et al., 2005; Tian et al., 2005; Xu et al., 2005; Schauland and Kummert, 2007; Grubb et al., 2004). The experimental results showed that the tree classifier had good overall performance. Its detection rate was 92.34% on average and its false alarm rate was about 0.72%. The cascade-AdaBoost classifier is somehow similar to a skewed tree classifier. 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 samples 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, Chen and Chen (2008) proposed a novel cascade classifier that could exploit both the stage-wise and the crossstage information. In their approach, some meta-stage classifiers were added to the cascaded classifier to utilize inter-stage 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 of the stage-wise and the cross-stage information. Cao et al. (2008) 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.

AdaBoost classifier or cascade-AdaBoost classifier, as well as parallel-SVM or cascade-SVM, are the most widely used classifiers among those pedestrian detection methods that have been proposed by now. The main reason is that each has its advantages and disadvantages. AdaBoost classifier has high detection rate, but its false alarm rate is relatively high; while cascade-AdaBoost classifier has low false alarm rate, but cascade reduces its detection rate at the same time. In order to maintain high detection rates, more weak classifiers shall be used, which results in that the latter layers are more difficult to be classified; parallel-SVM and cascade-SVM also face the same situation. On the other hand, sliding window combined with various features leads to a great number of features. SVM requires using all features while AdaBoost classifier does not, which causes that the speed of SVM is slower than that of the AdaBoost classifier. Although the use of nonlinear kernel SVM presents better performance than that of the AdaBoost classifier of linear classifier, the nonlinear kernel SVM consumes more time than that of linear kernel SVM and linear classifier. Over all of the above discussion, it can be found that there is some space for improvement. If the cascade-AdaBoost classifier is used, the problems that the detection rate decreases with cascade and the latter layers are more difficult to be classified must be improved. If SVM is used, the problems of a bulky number of input features and much consumed time of dimensions much be improved. Therefore, this paper, based on cascade-AdaBoost classifier, proposes a cascade classifier combining AdaBoost classifier and SVM for pedestrian detection. The following chapters will introduce the proposed methods and experiments respectively.

3. Features extraction

In this paper, we have used a sliding window of fixed size to segment the image into candidate regions. After the candidate regions were extracted, we used three common types of feature extraction methods in pedestrian detection; namely, edge detection, Haar-like features and HOG features. Edge detection applies the Sobel method (Engel, 2006) which is commonly used and is therefore not applied here. The other two methods are introduced in the following sections. After the feature extraction was completed, the completed cascade classifier in the previous training was used to determine whether it was a pedestrian or not. The following section shall introduce the feature extraction methods of the Haar-like matrix features and the HOC.

3.1. Haar-like feature

The Haar-like feature is commonly used in pedestrian detection (Viola and Jones, 2005,2004). It is an extraction method used



Fig. 2. Haar-like feature mask.

in a local block and its calculation is simple. In this paper, we have used four kinds of Haar-like feature masks, as shown in Fig. 2. The two left masks included two blocks, and the third and fourth masks included three blocks and four blocks. These four masks covered the candidate regions sequentially. A feature value was obtained by calculating the difference of the sums of the image gray values at the corresponding location of the black and white blocks. The features were extracted through a full-scanning method of the candidate region. After adopting four different masks to complete the full-scanning of a candidate region, a feature vector was obtained. Since it couldn't be determined whether the mask size was helpful for pedestrian classification, this paper has adopted the Haar-like masks of various scales from the four basic masks so as to achieve better features.

3.2. Histograms of oriented gradients

The feature extraction method of the histograms of oriented gradients (HOG) is another commonly used method in pedestrian detection (Dalal and Triggs, 2005; Tsai et al., 2010). First, the 3×3 pixel block in the candidate region is regarded as a cell and the candidate region is segmented into $m \times n$ blocks. Next, the magnitude and direction of the gradient of the grayscale pixel value in each cell is calculated and the statistics histogram of each block can be obtained based on the magnitude and direction of the gradient in all cells of the block. Finally, the histograms of all blocks are combined into a vector as a pedestrian detection feature. In this paper, the gradient calculation has used the Sobel method (Engel, 2006) to obtain the gradient horizontal component G_x and vertical component G_y of the grayscale pixel value in each cell. The gradient magnitude of the grayscale pixel value in one cell is calculated and expressed as follows:

$$\left|\nabla f\right| = \sqrt{G_x^2 + G_y^2},\tag{1}$$

with the gradient direction θ , the calculation is as follows:

$$\theta = \operatorname{atan}\left(\frac{G_y}{G_x}\right). \tag{2}$$

Gathering statistics on the gradient magnitude of the cell in each block is required based on the directional features. The gradient direction is divided by 45 degree as a unit; therefore, eight units, i.e. U_k , k=1, ..., 8 is obtained. When θ belongs to unit U_k , then the accumulated magnitude $|\nabla f|$ on the direction of U_k is calculated using the following equation:

$$\varphi_k \leftarrow \varphi_k + |\nabla f|, \ \theta \in U_k, \ k = 1, \dots, 8, \tag{3}$$

wherein, ϕ_k is the accumulated value of 8 directions, k=1, ..., 8, which indicates eight directions.

4. The proposed self-constructing cascade classifier

In this section, we shall introduce our proposed selfconstructing cascade-AdaBoost-SVM classifier which combines AdaBoost classifier (Viola and Jones, 2004,2005) and SVM (Nello and John, 2000; Chen et al., 2006; Wang et al., 2010). First, this section introduces the AdaBoost classifier and cascade-AdaBoost classifier, and then proceeds with our proposed self-constructing cascade-AdaBoost-SVM classifier and training algorithm.

4.1. The cascade-AdaBoost classifier

The AdaBoost classifier is a parallel classifier combined with many linear weak classifiers. Each weak classifier only focuses on the classification of one dimension in the input feature vector. During the entire training process after the goal is given to the classifier, the algorithm is able to self-adaptively increase the number of weak classifiers so as to improve the overall accuracy rate of the classification and focus on key features. After a weak classifier is added, the algorithm uses the minimum error to calculate the weight value of this weak classifier and re-adjust the weight value of every training sample, and then pass the value to the next newly added weak classifier. Based on the newly added weak classifier, the effect of the overall parallel classifier is improved. Fig. 3 shows the algorithm of the AdaBoost classifier.

Since the AdaBoost classifier is a strong classifier composed of many weak classifiers, the selection of effective weak classifiers is important. Fig. 4 shows the case of two classifications for one dimension. Assuming that both positive and negative samples have Gaussian probability distribution features, the blue solid line covered by the Gaussian function is the probability distribution of positive samples and the red dotted line covered by the Gaussian function is the probability distribution of negative samples. Both Gaussian functions are partially overlapped. The weak classifiers obtained by the minimum mean square error method can get the best accuracy rate, as shown in the classifier location of the critical value marked by the 'Th2' line in Fig. 4. But with the high detection rate requirement, this classification leads to a wrong classification of some positive samples. In case of a high detection rate requirement, the weak classifiers must be able to correctly classify all positive samples and maintain the error classification result of the negative samples only, as shown in the classifier location of the critical value marked by the 'Th3' line in Fig. 4. In this case, two types of classification samples can have a totally correct detection rate and high false alarm rate. In contrast, if the zero false alarm rate is set as the target, then the classifier would use the location of the critical value marked by the 'Th1' line in Fig. 4. Since we have focused on high detection rate in this paper, the simplest method to achieve a high detection rate is to change the initial weight value setting method of each training sample assuming that the AdaBoost algorithm would not be changed. If the weight value ω_p of a positive sample is equal to the weight

- 1. Given example images $(\mathbf{x}_1, y_1), ..., (\mathbf{x}_m, y_m)$, where $\mathbf{x}_i \in \mathbb{R}^n, y_i \in \{1, -1\}$, for positive and negative example respectively.
- 2. Initialization distribution $\omega_t(1)=1/2p$ and 1/2q, for y = 1 and -1, respectively where p and q are the number of positives and negatives respectively.
- 3. For t = 1, ..., T

• Find classifier $h_i: \mathbf{x} \rightarrow \{1, -1\}, h_t = \operatorname{argmin} \varepsilon_i$, where

• Weight classifier: $(z_j = \sum_{i=1...m} \omega_i(t) [v_i \neq h_j(x_{i,j})], j=1...,n$

 $\beta_i = 0.5 \ln[(1 - \varepsilon_i)/\varepsilon_i].$

Update distribution:

 $\omega_i(t+1) = \omega_i(t) \exp[-\beta_i y_i h_i(\mathbf{x}_i)]/Z_i,$ Z_i is for normalization.

4. Output final classifier: $H(\mathbf{x}) = \operatorname{sign}(\Sigma_{t=1...T}\beta_t h_t(\mathbf{x}))$.

Fig. 3. AdaBoost algorithm.



Fig. 4. Diagram of the weak classification performance evaluation.

value ω_n of all negative samples given that no positive sample can be falsely classified, the relationship can be expressed in the following equation:

$$\begin{cases} \omega_p = q\omega_n, \\ p\omega_p + q\omega_n = 1, \end{cases}$$
(4)

By solving this simultaneous equation, we can get the weight values $\omega_p = 1/(p+1)$ and $\omega_n = 1/q(p+1)$; the result can be used to replace the initial setting of the weight value of training samples in the AdaBoost algorithm, shown in Fig. 3. The experiment has shown that this equation is capable of maintaining a high detection rate for the classification result. However, the AdaBoost classifier still has a high false alarm rate. To improve this condition, Viola and Jones (2005,2004) have further proposed a cascade-AdaBoost classifier to reduce the false alarm rate.

4.2. The self-constructing cascade-AdaBoost-SVM classifier

In previous section, 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 over-fitting and time-consuming; so in this paper, we propose a cascade classifier combined AdaBoost classifier and SVM, which we call cascade-AdaBoost-SVM classifier; because SVM can solve the nonlinearly classification 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 classifier and SVM. The cascade classifier proposed by us can increase AdaBoost classifier or SVM adaptively.

Fig. 5 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 n_{th} and the target F_{target} of the overall false alarm rate of classifiers. Suppose assume the sets of positive examples and negative examples are called P and N respectively; the construction algorithm of the cascade classifier is mainly composed of two loops; the internal loop mainly uses the above mentioned AdaBoost algorithm to train AdaBoost classifier; each time a weak classifier is added, the present AdaBoost classifier will be reappraised to see if it has satisfied the condition; if it has, continue adding weak classifiers until conditions of the inner loop are not satisfied; otherwise, it determines whether the number n_i

Step 1. User selects values for f , the maximum acceptable false alarm rate per layer and d , the minimum acceptable detection rate per layer.
Step 2. User selects target overall false alarm rate, F_{target} .
Step 3. User selects values for n_{th} , the maximum acceptable number of weak
classifiers.
Step 4. P and N are the set of positive and negative examples.
Step 5. $F_0 = 1; D_0 = 1; i = 0;$
Step 6. While $(F_i > F_{target})$
• $i = i + 1; n_i = 0; F_i = F_{i-1};$
• while $(F_i > f \times F_{i-1})$ and $(n_i < n_{th})$
• $n_i = n_i + 1;$
• 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 .
• $if(n_i \ge n_{th})$
• 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}
• <i>N</i> is NULL.
Step 7. 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.

Fig. 5. Training algorithm for building a cascade AdaBoost-SVM classifier.

of weak classifiers is greater than the maximum value n_{th} ; 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; otherwise, training of AdaBoost classifier at this layer will be completed. 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 all positive examples; otherwise, terminate the training of cascade-AdaBoost-SVM classifier.

In addition, the SVM training of the cascade-AdaBoost-SVM is different from the previous training which focuses on a higher accuracy rate. Our proposed cascade classifier is based on the premise of a high detection rate to further reduce the false alarm rate. By considering the accuracy rate, a lower false alarm rate could be obtained but might lead to a wrong classification of some positive training samples, causing a detection rate reduction of the SVM classification result; this further reduces the detection rate of the overall cascade classifier. In order to maintain a high detection rate of the overall cascade classifier, the related parameters selected in the SVM training would be capable of maintaining a higher detection rate rather than a higher accuracy rate. Therefore, the cascade-AdaBoost-SVM classifier can use classifiers with less layers but achieve lower overall false alarm rate than that of the cascade-AdaBoost classifier.

5. Experimental results

In this section, four experiments have been performed to demonstrate the proposed method. The first experiment used a captured video image on campus and the existing pedestrian sample database was used from the second to the fourth experiments. The PETs (VS-PETS' 2003), INRIA (INRIA Person Dataset) and MIT (MIT Pedestrian Dataset) pedestrian sample databases were also included. In addition, we have defined the accuracy rate (*AR*), detection rate (*DR*) and false alarm rate (*FAR*) of those three parameters in order to assess the results of our proposed methods in every database; the calculation is as follows:

$$AR = \frac{TP + TN}{p + q} \times 100\%,\tag{5}$$

$$DR = \frac{TP}{p} \times 100\%,\tag{6}$$

$$FAR = \frac{FP}{q} \times 100\%,$$
(7)

where *p* and *q* represent the number of pedestrian sample collections and non-pedestrian samples. TP is true positive representing the number of pedestrian samples which are detected as pedestrians; FP is false positive representing the number of non-pedestrian samples which are detected as pedestrians. Therefore, AR is defined as the ratio of all detected correct samples divided by the overall sample number, while DR is defined as the ratio of positive classifications divided by the positive sample number. The higher the DR is, the higher the ratio of positive samples being detected. FAR is defined as the ratio of the number of positive samples from the negative samples divided by the negative samples. The higher the FAR, the higher the false alarm rate and vice versa. In the following experiments, we have used random sampling and the 3-fold cross-validation method for all sample sets of the database. Each experimental result was the average of three repeated experiments which were compared with the results obtained using a single SVM and the cascade-AdaBoost classifier. With feature extraction, the size of each sample was normalized to an image sample of 15 pixels wide and 36 pixels high, and the standard deviation of the grayscale value was normalized to 1. The Haar-like feature and HOG feature calculation as well as edge detection were conducted on the sample. All three features were combined into an input feature vector. The training and test conducted on the pedestrian detection were carried out using our proposed cascade classifier, single SVM and cascade-AdaBoost classifier. The next step was to describe the experiment and result of each database.

5.1. Experiment on captured images in campus

In this experiment, pedestrian and non-pedestrian databases were taken from video images of various scenes around campus, and we divided them into set 1 and set 2. Each set contained 10,560 sheets of samples, composed of 480 sheets of pedestrian samples and 10,080 sheets of non-pedestrian samples, in which the pedestrian samples contained the front, back, and different sides of pedestrians, while the non-pedestrian samples included roads, trees, traffic signs, banners, cars, motorcycles, bicycles, telephone poles, flagpoles, incomplete pedestrian images, background and other non-pedestrian samples. The samples in set 1 were selected from the same scene, and the contrast between the pedestrian and the background in the pedestrian samples was significant and simple. Fig. 8 shows the experimental scene in set 1 and Fig. 6 shows some pedestrian and non-pedestrian samples in set 1. The samples in set 2 were selected from different scenes in various campuses and the background in the pedestrian samples was relatively complex, in which there was overlapping among pedestrians. Fig. 9 shows the experimental scenes in set 2 and Fig. 7 shows some pedestrian and non-pedestrian samples in set 2.

First, the detection rate d of each layer in the cascade classifiers was set to 0.99 and the false alarm rate f was set to 0.5. The false alarm rate F_{target} of the overall cascade classifier was set to 1e-5. In the set 1 experiment, the cascade-AdaBoost classifier used two layers of AdaBoost classifier, in which the first layer only used one weak classifier and 3-5 weak classifiers were used in the second layer. Assuming that the maximum allowable number of weak classifiers n_{th} is 2, then our proposed cascade-AdaBoost-SVM classifier had two layers, in which the first layer was the AdaBoost classifier and the second layer was the SVM with RBF kernel function, parameter $\sigma = 0.2$ and parameter *C* set to 10. In the set 2 experiment, the cascade-AdaBoost classifier used 6 layers of the AdaBoost classifiers in total, in which the average numbers of weak classifiers were 4, 5, 7, 9, 9, and 12. The maximum allowable number of weak classifier n_{th} was set to 10. The cascade-AdaBoost-SVM classifier used 5 to 6 layers of classifiers, in which the fifth or sixth layer's classifier of the



Fig. 6. Some training samples in set 1 (a) pedestrian samples, and (b) non-pedestrian samples.



Fig. 7. Some training samples in set 2 (a) pedestrian samples, and (b) non-pedestrian samples.

cascade classifier used SVM with RBF kernel function, parameter σ =5 and parameter *C* set to 10. Since the training samples in set 2 were a little more complex than those in set 1, the layer number required by the cascade classifier was more than that in set 1 and

the number of weak classifier in each layer was much greater. In addition, when the AdaBoost classifier was changed to SVM given the uncomplicated training samples in the experiment, the overall obtained false alarm rate was lower than the preset F_{target}

because the accuracy rate of the SVM training result could reach 100%. It can be seen from the architecture that the SVM classifier was the last layer of the cascade classifier. Since the previous layers' AdaBoost classifiers have filtered over 90% of the training samples, the training samples for SVM decreased sharply, allowing the SVM to easily complete the training. Another advantage of this architecture is that it solved the problems of SVM's unsuitability for large training samples.

Table 1 shows the experimental results. The single SVM classifier had the highest accuracy rate at 99.91% in set 1 and 98.85% in set 2, because the single SVM classifier did not focus on the detection rate but on the accuracy rate instead. While the accuracy rate of our proposed cascade classifier was 99.71% in set 1 and 98.25% in set 2, which were slightly better than those of the cascade-AdaBoost classifier's 99.60% in set 1 and 98.14% in set 2, our proposed cascade classifier had the highest detection rate of

99.45% in set 1 and 95.97% in set 2, which were better than those of the single classifier and the cascade-AdaBoost classifier. The results have met our expectation. In some applications which prioritize security like driver assistance system or surveillance system, the detection rate is considered more important than the accuracy rate. Our proposed cascade classifier had better performance. In comparing the training time, the consumed training time of the cascade classifier was significantly lower than that of the single SVM due to the cascade classifier's reduction of the number of training samples layer by layer. In addition, set 1 was far lower than set 2. As for the larger training samples required in the future, the simple SVM might not be able to fully perform the training. In short, the experiments showed that the results of the three types of classifiers had little difference in set 1, because of the significant identification degree of pedestrian and non-pedestrian samples in set 1. Therefore, good results were obtained. The results

Table 1

Comparison of experimental results and training time in campus video images.

Dataset	Classifier	Testing perfo	rmances (%)		Training time (s)
		AR	DR	FAR	
Set 1	Signal SVM (RBF Kernel, C=100, σ =5)	99.91	99.31	0.00	12752.1
	Cascade-AdaBoost	99.60	99.17	0.34	380.9
	Cascade-AdaBoost-SVM (RBF Kernel, $C=10$, $\sigma=0.2$)	99.71	99.45	0.26	371.4
Set 2	Signal SVM (RBF Kernel, $C=100, \sigma=5$)	98.85	94.10	0.48	9648.5
	Cascade-AdaBoost	98.14	95.91	1.54	2027.1
	Cascade-AdaBoost-SVM (RBF Kernel, C=10, σ =5)	98.25	95.97	1.43	1947.8



Fig. 8. Pedestrian detection results in several video images in set 1.

in set 2 were significantly different from those in set 1, because the pedestrian samples in set 2 were complicated.

Fig. 8 shows the experimental scenes in set 1 and some pedestrian image detection results, in which the final detection of pedestrian is combined with many similar pedestrian detection results. It can be observed from the experimental results that our proposed pedestrian detection method can effectively detect pedestrians, even if the pedestrians held umbrellas, as long as the frame of the umbrellas is not too big, as shown in Fig. 8(a) and (b). In Fig. 8(c), although pedestrians were partially overlapped. the classifier could also block them. Fig. 8(e) contains moving trains and vehicles, and the classifier could also block the pedestrians. There were pedestrians who were not detected at the upper right shadows in Fig. 8(f) due to brightness and obstruction. Fig. 9 shows some experimental scenes in set 2 along with pedestrian detection results. The samples of this set were selected from multiple scenes, making them complicated. Fig. 9 shows that there are other moving objects in the image, such as cars or motorcycles, but our proposed cascade classifier could still carry out the correct classification and correctly detect the locations of pedestrians. Fig. 9(e) and (f) shows the results of pedestrian detection at dusk, while Fig. 9(c) and (e) shows some cases of wrong detection. The error detection in Fig. 9(c) was caused by pedestrians obscured by the light pole while 9 (e) was caused by false alarm. To further validate the performance of our proposed pedestrian detection, we have also applied three commonly used pedestrian detection databases such as PETs (VS-PETS' 2003), INRIA (INRIA Person Dataset), and MIT (MIT Pedestrian Dataset), which are described below.

5.2. Experiment on PETs database

The PETs database (VS-PETS' 2003) was retrieved from the video image of a replayed football match. It included video images shot at different angles using cameras. We selected the video image of one scene and randomly extracted an experimental sample set of 10,560 sheets which consisted of 480 sheets of pedestrian samples and 10,080 sheets of non-pedestrian samples. Although the background of pedestrian samples is very simple, the pedestrians had complex movements such as walking, running and playing ball, etc. The pedestrians were highly similar in clothing, which increased the difficulty of detection. The non-pedestrian samples also contained some pedestrians, excluding those in the background. Fig. 10 shows some training samples retrieved from the PETs database and the size of each sample is fixed at 15×36 pixels.

In this experiment, the detection rate *d* of each layer of the cascade classifier was set to 0.99 and the false alarm rate *f* was set to 0.5. The maximum allowable number of weak classifiers n_{th} was set to 5. The cascade-AdaBoost classifier used AdaBoost classifiers with a total of 3 to 4 layers, and the average numbers of weak classifiers were 1, 2, 4, and 8. Our proposed cascade-AdaBoost-SVM classifier used 3-layer or 4-layer classifiers whose parameter *C* and variance σ of SVM were set to 10 and 2, respectively. Table 2 shows the experimental results. Our proposed cascade classifier had an *AR* of 99.36% which was better than the cascade-AdaBoost classifier's *AR* of 99.22%. With the detection rate, our proposed cascade classifier and cascade-AdaBoost classifier both reached the 97.78% and were higher



Fig. 9. Pedestrian detection results in several video images in set 2.



Fig. 10. Some training samples in PETs database, (a) pedestrian samples, (b) non-pedestrian samples.

Table 2

Comparison of identification results and training time in PETs database.

Classifier	Testing perfor	Training time (s)		
	AR (%)	DR (%)	FPR (%)	
Signal SVM (RBF Kernel, $C=100$, $\sigma=5$) Cascade-AdaBoost Cascade-AdaBoost-SVM (RBF Kernel, $C=10$, $\sigma=2$)	99.57 99.22 99.36	96.59 97.78 97.78	0.01 0.58 0.42	12675.4 633.7 529.3

than the SVM's *DR* of 96.59%. It can also be observed that our proposed cascade classifier had the shortest training time. The experimental results were similar to the results of our retrieved campus video images. Fig. 11 shows the pedestrian detection result of partial images in the PETs database, from which we can see that our proposed method was capable of detecting the locations of footballers effectively, but in case of overlapping or obstruction, some pedestrians could not be detected correctly.

5.3. Experiment on INRIA database

In the INRIA pedestrian database (INRIA Person Dataset) experiments, we also captured a total of 10,560 training samples, composed of 480 pedestrian samples and 10,080 non-pedestrian samples. The experimental samples included images of the countryside, city, snow and the sea as well as other sights. Therefore, the positions of pedestrians varied and most of them were overlapped or unsteady. The non-pedestrian samples had also collected various images which was quite challenging. Fig. 12 shows some pedestrian and non-pedestrian samples in the INRIA database. The size of each sample was normalized to 15×36 pixels. Based on the pictures, we can see that the samples of INRIA were more complex and varied compared to the previous experimental samples. In addition, the upper and lower boundaries of the pedestrian samples contained extra background which increased the difficulty of pedestrian detection.

In the INRIA database experiment, the detection rate *d* of each layer of the cascade classifier was set to 0.99 and the false alarm rate f was set to 0.5. In this experiment, the cascade-AdaBoost classifier used 6 layers of AdaBoost classifiers in total, in which the average numbers of weak classifiers were 8, 14, 18, 25, 26, and 35. We have set the maximum allowable number of weak classifiers n_{th} at 25. Therefore, our proposed cascade-AdaBoost-SVM classifier used 4 layers or 5 layers of classifiers and the fourth layer's classifier or the fifth layer's classifier was replaced by the SVM, of which the parameter *C* and variance σ of SVM were set at 10 and 3.5, respectively. Since the training samples were complex, some samples remained unclassified after completion of the SVM training. Therefore, another AdaBoost classifier of one layer was used to classify the remaining samples, in which the average number of weak classifiers was 4. Table 3 shows the experimental results. Our proposed cascade classifier had an AR of 96.44% which was better than the single SVM's AR of 95.49% and the cascade-AdaBoost classifier's AR of 96.02%. With the detection rate, our proposed cascade classifier reached a DR of 94.72% which was better than the cascade-AdaBoost classifier's DR of 94.24% and it was 27% higher than the single SVM's DR of 67.43%. Based on the above findings, our method had a better effect whether it was applied in a simple or complex pedestrian detection environment. In addition, our proposed cascade classifier had the shortest training time which was 3 times faster than a single SVM.



Fig. 11. Pedestrian detection results of some video images in PETs database.



Fig. 12. Some training samples in the INRIA database, (a) pedestrian samples, (b) non-pedestrian samples.

5.4. Experiment on MIT database

Finally, the MIT pedestrian database (MIT Pedestrian Dataset) was used for the pedestrian detection experiment. Its shooting angle of pedestrians was similar to that of the INRIA database; both had a horizontal shooting angle which was similar to the shooting angle of a camera inside a car. In addition, the database only contained pedestrian samples and none of the non-pedestrian

samples. Therefore we used the non-pedestrian samples of the INRIA database. In this experiment, we have extracted a total of 9900 sheets of training samples, composed of 900 pedestrian samples and 9000 non-pedestrian samples. Furthermore, the pedestrian samples were all frontal or posterior images of pedestrians, and some pedestrian samples were overlapped. Figure 13 shows some pedestrian training samples in the MIT database and the size of each sample was normalized to 15×36 pixels.

Table 3

Comparison of identification results and training time in INRIA database.

Classifier	Testing performances			Training time (s)	
	AR (%)	DR (%)	FPR (%)		
Signal SVM (RBF Kernel, C=100, σ =5) Cascade-AdaBoost Cascade-AdaBoost-SVM (RBF Kernel, C=10, σ =3.5)	95.49 96.02 96.44	67.43 94.24 94.72	0.51 3.72 3.31	12675.4 5273.9 4730.7	



Fig. 13. Some pedestrian training samples in the MIT database.

Table 4 Comparison of identification results and training time in MIT database.

Classifier	Testing performances			Training
	AR (%)	DR (%)	FPR (%)	time (3)
Signal SVM (RBF Kernel, $C=100, \sigma=5$) Cascade-AdaBoost Cascade-AdaBoost-SVM (RBF Kernel, $C=10, \sigma=0.5$)	95.68 96.22 97.02	83.89 97.26 97.30	0.4 4.12 3.07	11531.8 5243.9 4276.2

In the MIT database experiment, the cascade-AdaBoost classifier used a 6-layer AdaBoost classifier, and the average numbers of weak classifiers were 6, 12, 16, 20, 20 and 27. We have set the maximum allowable number of weak classifiers n_{th} to 20. Therefore, our proposed cascade-AdaBoost-SVM classifier used 4 layers or 5 layers of classifiers and the fourth layer's classifier or the fifth layer's classifier was replaced by SVM; the parameter C and variance σ of SVM were set to 10 and 0.5, respectively. Table 4 shows the experimental results. Our proposed cascade classifier had an AR of 97.02% which was better than the single SVM's AR of 95.68% and the cascade-AdaBoost classifier's AR of 96.22%. As for the detection rate, our proposed cascade classifier reached a DR of 97.30% which was slightly higher than the cascade-AdaBoost classifier's DR of 97.26% and 13.4% higher than a single SVM's DR of 83.89%. Our proposed cascade classifier had the shortest training time which was 2.5 times faster than a single SVM.

Since the settings of parameters greatly affect the performances of classifiers, we discuss the parameter setting method of our proposed cascade classifiers in the last part. Parameter *C* and sigma of SVM is set to get optimal parameters by the use of grid algorithm (Hsu et al., 2003). Since the optimal value *C* is not unique, we set a fixed value *C* in this paper in order to speed up the effectiveness of grid algorithm and adjust sigma to achieve the best classification results. The experiment results show that, for the experiments of different databases, a single SVM classifier aims at the classifier trainings based on full training samples and all feature dimensions, thus it gets the same optimal parameter setting. However, when our proposed cascade classifier is used, since the previous layers' AdaBoost classifiers have filtered over 90% of the training samples and perform SVM training based on the feature dimensions selected by AdaBoost classifier without calculating all dimensions. Therefore, different databases result in different parameter setting. In addition, since parameter n_{th} is the basis of cascade classifier algorithm determines whether AdaBoost classifier is replaced by SVM, the increased number of weak classifiers of a certain AdaBoost classifier layer is multiplied by several times of the average of increased numbers of previous several layers, then we set n_{th} as the number of weak classifiers of previous layers. For example, in the MIT database experiment, the average numbers of weak classifiers were 6, 12, 16, 20, 20 and 27, the increased numbers from the second layer are 6, 4, 4, 0 and 7 respectively. Therefore, the increased number of the sixth layer is greater than or equal to two times of the average 3.5. Thus, n_{th} is set to 20 which is the number of weak classifiers of previous layers; otherwise, if the number of weak classifiers is not increased by multiples, n_{th} is set to the weak classifiers number of AdaBoost classifier at penultimate layer of cascade classifier. In other words, the last layer is replaced by SVM directly, for example, in the Set 1 database of the experiment A.

6. Conclusion

In this paper, we have proposed a new cascade classifier combining AdaBoost classifier and SVM. This self-constructing classifier is based on the training sample set and preset target, therefore it does not have a fixed structure. It is called a selfconstructing cascade-AdaBoost-SVM classifier which addresses the problems of the original cascade-AdaBoost classifier and at the same time improves the time consumption of SVM while being applied in large-scale training sample sets. Furthermore, we have also improved the initial setting method of the AdaBoost classifier algorithm so that the AdaBoost classifier would be able to focus on the classification detection rate, since the calculation of the AdaBoost classifier is simple and the SVM classifier has many unique advantages in solving problems for small samples and non-linear problems. Our proposed self-constructing cascade-AdaBoost-SVM classifier combines the advantages of these two classifiers. Based on the experimental results. both our proposed cascade classifier and other two classifiers have good results in simple pedestrian detection environments like in plazas or playgrounds and other places. Our proposed cascade classifier has a better performance than other two classifiers in complex and diversified pedestrian detection environments like the INRIA or MIT database. In addition, our proposed cascade classifier has the shortest training time, making it is easier to conduct numerous tests and verifications. On the basis of the same input features, our proposed classifier has an overall detection rate of 97.04%, which is better than 88.26% of single SVM and 96.87% of cascade-AdaBoost classifier. Our proposed classifier has an overall false alarm rate of 1.70%, which is lower than 2.06% of cascade-AdaBoost classifier; however, it is higher than 0.28% of the single SVM classifier. One of reasons causing high false alarm rate is that the high detection rate is set as the prerequisite when our proposed classifier is constructed, which results in more false alarm cases introduced. For this case, the false alarm rate could have been reduced by the use of cascade classifiers. Another reason is that samples used in the experiments are not sufficient enough, which leads to the fact that a few layers are required to construct cascade classifiers to complete training, so the false alarm rate of grade 1e-6 which is achieved by the application of cascade-AdaBoost classifier to face the detection can not be reached. However, the overall accuracy rate of the proposed method is better than other two classifiers. Future studies will firstly perform a great deal of database experiments, and we secondly intend to conduct pedestrian tracking to further determine the pedestrian paths and behaviors needed for additional researches.

References

- Cao, X.-B., Qiao, H., Keane, J., 2008. A low-cost pedestrian-detection system with a single optical camera. IEEE Trans. Intell. Transp. Syst. 9 (1), 58–67.
- Chen, D., Cao, X.-B., Xu, Y.-W., Qiao, H., 2006. An evolutionary support vector machines classifier for pedestrian detection. In: Proceedings of the International Conference on Intelligent Robots and Systems, pp. 4223–4227.
- Chen, J.-H., Chen, C.-S., 2004. Reducing SVM classification time using multiple mirror classifiers. IEEE Trans. Syst. Man Cybern. 34 (2), 1173–1183.
- Chen, Y.-T., Chen, C.-S., 2008. Fast human detection using a novel boosted cascade structure with meta stages. IEEE Trans. Image Process. 17 (8), 1452–1464.
- Cheng, H., Zheng, N., Qin J., 2005. Pedestrian detection using sparse gabor filter and support vector machine. In: Proceedings of the IEEE Intelligent Vehicles Symposium, pp. 583–587.
- Cheng, W.-C., Cheng, U.-Y.,2010. Pedestrian detection in video using shape features and mixture of SVMs. In: Proceedings of the 3rd International Congress on Image and Signal Processing, Chain, pp. 16–18.
- Dalal, N., Triggs, B., 2005. Histograms of oriented gradients for human detection. In: Proceedings of the Computer Vision Pattern Recognition, vol. 1, pp. 886– 893.
- Ding, X., Xu, H., Cui, P., Sun L., 2009. A Cascade SVM approach for headshoulder detection using histogram of oriented gradient. In: Proceedings of the IEEE International Symposium on Circuits and Systems, pp. 1791–1794.
- Dollar, P., Tu, Z., Perona, P., Belongie, S., 2009. integral channel features. In: Proceedings of the British Machine Vision Conference.
- Dollar, P., Belongie, S., Perona, P., 2010. The fastest pedestrian detector in the west. In: Proceedings of the British Machine Vision Conference.
- Dollar, P., Wojek, C., Schiele, B., Perona, P., 2012. Pedestrian detection: an evaluation of the state of the art. IEEE Trans. Pattern Anal. Mach. Intell. 34 (4), 743–761.

Engel, K., 2006. Real-time Volume Graphics, pp. 112-114.

- Enzweiler, M., Gavrila, D.M., 2009. Monocular pedestrian detection: survey and experiments. IEEE Trans. Pattern Anal. Mach. Intell. 31 (12), 2179–2195.
- Freund, Y., Schapire, R. E., 1996. Experiments with a new boosting algorithm. In: Proceedings of the 13th Conference on Machine Learning, pp. 148–156.
- Freund, Y., Schapire, R.E., 1997. A decision-theoretic generalization of on-line learning and an application to boosting. J. Comput. Syst. Sci. 55 (1), 119–139. Gandhi, T., Trivedi, M.M., 2007. Protection systems: issues, survey, and challenge.
- IEEE Trans. Intell. Transp. Syst. 8 (3), 413–430. Gavrila, D., 2000. Pedestrian detection from a moving vehicle. In: Proceedings of
- Gavrila, D., 2000. Pedestrian detection from a moving venicle. in: Proceedings of ECCV, pp. 37–49.
- Geronimo, D., Lopez, A.M., Sappa, A.D., Graf, T., 2010. Survey of pedestrian detection for advanced driver assistance systems. IEEE Trans. Pattern Anal. Mach. Intell. 32 (7), 1239–1258.
- Grubb, G., Zelinsky, A., Nilsson, L., Rilbe, M., 2004. 3D vision sensing for improved pedestrian safety. In: Proceedings of the IEEE Intelligent Vehicles Symposium, pp. 19–24.
- Hiromoto, M., Sugano, H., Miyamoto, R., 2009. Partially parallel architecture for AdaBoost-based detection with Haar-like features. IEEE Trans. Circuits Syst. Video Technol. 19 (1), 41–52.
- Hsu, C.-W., Chang C.-C., Lin, C.-J., 2003, A Practical Guide to Support Vector Classification .
- INRIA Person Dataset. < http://pascal.inrialpes.fr/data/human/>.
- Kukenys, I., McCane, B., 2008. Classifier cascades for support vector machines. In: Proceedings of the 23rd International Conference on Image and Vision Computing, New Zealand, pp. 1–6.
- Li, X., Wang, L., Sung, E., 2008. AdaBoost with SVM-based component classifiers. Eng. Appl. Artif. Intell. 21 (5), 785–795.
- Liue, M.-B., Fuentes, O., 2009. Object detection using image reconstruction with PCA. Image Vision Comput. 27 (1–2), 2–9.
- MIT Pedestrian Dataset. http://cbcl.mit.edu/software-datasets/PedestrianData.html>.
- Ma, G., Muller, D., Park, S.-B., Muller-Schneiders, S., Kummert, A., 2009. Pedestrian detection using a single-monochrome camera. IET Intell. Transp. Syst. 3 (1), 42–56.
- Ma, Y., Ding, X., 2002. Face detection based on cost-sensitive support vector machines. Pattern Recognition Support Vector Mach., 260–267.
- Nam, B., Kang, S., Hong, H., 2011. Pedestrian detection system based on stereo vision for mobile robot. In: Proceedings of the 17th Frontiers of Computer Vision, pp. 1–7.
- Nello, C., John, S.T., 2000. An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge University Press.
- Sahbi, H., Geman, D., Perona, P., Dec. 2006. A hierarchy of support vector machines for pattern detection. J. Mach. Learn. Res. 7 (2), 2087–2123.
- Schauland, S., Kummert, A., 2007. Implementation and optimization of wavelet and symmetry features for vision-based pedestrian detection. In: Proceedings of the of IASTED International Conference on Graphics and Visualization in Engineering, pp. 19–24.
- Sessler, G., Martoyo, I., Jondral, F., 2001. RBF based multiuser detectors for UTRA-TDD. In: Proceedings of the IEEE VTS 54th VTC—Fall, vol. 1, pp. 484–486.
- Szarvas, M., Sakai U., Ogata J., 2006. Real-time pedestrian detection using LIDAR and convolutional neural networks. In: Proceedings of the Intelligent Vehicles Symposium, pp. 213–218.
- Tian, Q., Sun, H., Luo, Y. P., Hu, D. C., 2005. Nighttime pedestrian detection with a normal camera using SVM classifier. In: Proceedings of the International Symposium on Neural Networks, pp. 189–194.
- Tsai, H.-M., Huang, S.-S., Hsiao, P.-Y., 2010. Combining histograms of oriented gradients with shape of head for human detection. In: Proceedings of the 23rd IPPR Conference on Computer Vision, Graphics, and Image Processing, pp. 208–218.

VS-PETS'2003—INMOVE. <hr/>
http://www.cvg.cs.rdg.ac.uk/VSPETS/vspets-db.html >.</hr>

Viola, P., Jones, M.J., 2004. Robust real-time face detection. Int. J. Comput. Vision 52

- (2), 137–154.
- Viola, P., Jones, M.J., 2005. Detecting pedestrians using patterns of motion and appearance. Int. J. Comput. Vision 63 (2), 153–161.
- Wang, R., Fang, B., Hu, Z., Chen, L., Wang, W., 2010. Cascaded SVMs in pattern classification for time-sensitive separating. In: Proceedings of Third International Symposium on Intelligent Information Technology and Security Informatics, pp. 640–644.
- Wei, C., Cao, X., Xu, Y., Qiao, H., Wang, F., 2009. The treelike assembly classifier for pedestrian detection. In: Proceedings of the PAISI, pp. 232–237.
- Xu, F., Liu, X., Fujimura, K., 2005. Pedestrian detection and tracking with night vision. IEEE Trans. Intell. Transp. Syst. 6 (1), 63–71.
- Xu, Y.-W., Cao, X.-B., Qiao, H., 2011. An efficient tree classifier ensemble-based approach for pedestrian detection. IEEE Trans. Syst. Man Cybern. Part B: Cybern. 41 (1), 107–117.
- Zhang, J., Cheng, L., Ma, J., 2006. A new multiple classifiers combination algorithm. In: Proceedings of the First International Multi-Symposiums on Computer and Computational Sciences, pp. 287–291.