Expert Systems with Applications 39 (2012) 4274–4286

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

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

Pedestrian detection for intelligent transportation systems combining AdaBoost algorithm and support vector machine

Lie Guo [a,*], Ping-Shu Ge [b], Ming-Heng Zhang [a], Lin-Hui Li [a], Yi-Bing Zhao [a]

[a] School of Automotive Engineering, Faculty of Vehicle Engineering and Mechanics, State Key Laboratory of Structural Analysis for Industrial Equipment, Dalian University of Technology, Dalian 116024, PR China

[b] College of Electromechanical & Information Engineering, Dalian Nationalities University, Dalian 116000, PR China

ARTICLE INFO

Keywords:
Pedestrian detection
Two-stage classifier
Feature extraction
Support vector machine

ABSTRACT

Pedestrians are the vulnerable participants in transportation system when crashes happen. It is important to detect pedestrian efficiently and accurately in many computer vision applications, such as intelligent transportation systems (ITSs) and safety driving assistant systems (SDASs). This paper proposes a two-stage pedestrian detection method based on machine vision. In the first stage, AdaBoost algorithm and cascading method are adopted to segment pedestrian candidates from image. To confirm whether each candidate is pedestrian or not, a second stage is needed to eliminate some false positives. In this stage, a pedestrian recognizing classifier is trained with support vector machine (SVM). The input features used for SVM training are extracted from both the sample gray images and edge images. Finally, the performance of the proposed pedestrian detection method is tested with real-world data. Results show that the performance is better than conventional single-stage classifier, such as AdaBoost based or SVM based classifier.

1. Introduction

Pedestrians are the vulnerable participants among all the objects involved in the transportation system when crashes happen, especially those in motion on streets and roads under urban traffic situations. Therefore, road traffic safety has received much concern by governments and social organizations in China, such as the developments associated with intelligent transportation system (ITS) and safety driving assistant system (SDAS) technologies. The object of these technologies is to enhance comfort and safety of driver and road user (Gandhi & Trivedi, 2007; Vallejo, Albusac, Jimenez, Gonzalez, & Moreno, 2009).

Among the factors that may contribute to traffic accidents, human error is one of the most important factors, such as driver's inattention and wrong decisions. A World Health Organization report described traffic accident as one of the major causes of death and injuries around the world, accounting for an estimated 1.2 million fatalities and 50 million injuries (Peden et al., 2004). Unfortunately, a large majority of such deaths were not vehicle occupants but road users, consisting of pedestrians, bicyclists, two wheelers, and other small vehicles. According to the National Highway Traffic Safety Administration report, there were an estimated 5,811,000 police-reported traffic crashes in 2008 in the United States. Among those accidents, there were 69,000 pedestrians injured and 4,378 pedestrians killed. The pedestrian deaths accounted for 11.7% of the total fatalities. Most pedestrian fatalities occurred in urban areas (72%) under normal weather conditions (89%).

In developing countries such as India, Brazil and China, the problem is much worse. Accidents and the fatalities on road are the result of reciprocity of a number of factors. Road users of those countries, especially in India, are multiplex in nature, ranging from pedestrians, bi-cycles, and tractor trolleys, to various categories of three wheelers, motor cars, buses, trucks, and multi-axle commercial vehicles etc. India has the second largest road network in the world with over 3 million km of roads, 46% of which are paved. As a result, there were 105,725 people killed in road traffic crashes in India in 2006, ranking the top of other countries (Mohan, Tsimhoni, Sivak, & Flannagan, 2009). Pedestrians, bicyclists and motorized two-wheelers riders constitute 60%–80% of all traffic fatalities in India (Mohan, 2002). Fortunately, facing the terrible traffic situation, India government has done something to improve pedestrian safety. A project named "Traffic calming strategies to improve pedestrian safety in India" has been initiated, which is aimed to produce a theoretical and practical background to produce guidelines for India on traffic calming measures.

As for China, the traffic status is not satisfactory, according to the report of China Ministry of Public Security. There were 327,209 road traffic accidents in 2007, resulted in 380,442 people injured and 81,649 people killed. Among those accidents, there were 69,000

* Corresponding author.
E-mail address: lmonkygl@163.com (L. Guo).

0957-4174/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2011.09.106

pedestrians injured and 21,106 pedestrians killed. Compared with developed countries, the number of accidents and fatalities is much higher and the traffic problem is much worse. The pedestrian deaths accounted for 25.9% of the total fatalities, compared to 11.4% in the United States in 2007. Therefore, the solutions for China to improve the safety of pedestrian are critical. Commonly, pedestrian safety improvements could be made by using targeted countermeasures based on scientific, system-wide understanding of vehicle surroundings. If there is a possible collision between vehicles and pedestrians, something has to be done to warn the driver. Obviously, drivers receive a good deal of visual information while driving a vehicle (Armingol et al., 2007). In similar way, machine vision plays an important role in enhancing traffic safety and provides plentiful information to the driver. This paper aims to propose a pedestrian protection method based on monocular vision, which can detect potentially dangerous situations involving pedestrians ahead of time. Unlike the conventional methods, two-stage classifiers have been trained to realize pedestrian detection. The features used to train the classifier during each stage are different in view of different strategies.

The reminder of this paper is organized as follows: Section 2 reviews relevant researches related to pedestrian detection problem. Section 3 presents the entire pedestrian detection system in detail, including pedestrian segmentation based on cascaded classifiers trained by AdaBoost and pedestrian recognition using SVM training with multi-features. The experiment results are shown in Section 4. Finally, Section 5 concludes the paper with some possible directions for future work.

2. Literature review

Pedestrian safety has received much concern in recent years and considerable researches have been conducted by various groups to enhance the safety and mobility of pedestrians.

Inspired by the fact that human visual system can extract abundant information from the scene in real time, the commonly used sensors for detecting pedestrians are imaging sensors in visible light or infrared radiation images. Papageorgiou and Poggio (2000) described a monocular pedestrian detector based on two degree polynomial SVM. The features used to train the classifier were Haar wavelets that capture significant information of pedestrian. The same architecture was used to realize face and car detection tasks. They stated that it is the first people detection system described in literature, which is purely a pattern classification system and does not rely on motion, tracking, background subtraction, or any assumptions on the scene structure. Mohan, Papageorgiou, and Poggio (2001) located people with four distinct example-based detectors. Those detectors were trained to separately find four components of human body: the head, legs, left arm and right arm. After ensuring that these components were present in the proper geometric configuration, a second example-based classifier was applied to classify a pattern as either a person or not. They performed better results than a full-body person detector designed along similar lines. Sabzmeydani and Mori (2007) built shape-let features selected from low-level features to discriminate between pedestrians and non-pedestrians. Those features were used to train the final pedestrian classifier with AdaBoost algorithm. Dalal and Triggs (2005) proposed the grids of histogram of oriented gradient (HOG) descriptors with a linear SVM classifier. Their results showed that their feature sets significantly outperformed existing feature sets for human detection. Cheng, Zheng, and Qin (2005) proposed a pedestrian representation approach based on Spare Gabor Filters and SVM. Alonso, Llorca, and Sotelo (2007) described a comprehensive combination of feature extraction methods for vision-based pedestrian detection. They used different feature extraction methods for different sub-regions in the image and then combined with a SVM based classifier. Their results showed that combination of feature extraction methods is an essential clue for enhanced detection performance. Szarvas, Sakai, and Ogata (2006) presented a pedestrian detection method based on convolutional neural network (CNN). It could automatically optimize the feature representation to the pedestrian detection task and regularize the neural network. Compared with SVM classifier, they concluded that the accuracy of SVM classifier using the features learnt by CNN is equivalent to the accuracy of CNN. Furthermore, the computational demand of CNN classifier is lower than that of the SVM. Llorca, Sotelo, Parra, Ocaña, and Bergasa (2010) presented an analytical study of the depth estimation error of a stereo vision-based pedestrian detection sensor. Pedestrians were detected by combining a 3D clustering method with SVM classification. Their results indicated that the sensor provides suitable measurements despite its inner accuracy constraints due to the quantization error. In addition to the individual research, there exist several recent remarkable surveys and comparisons on pedestrian detection systems based on vision. For example, Gerónimo, López, Sappa, and Graf (2010) stated the problems arising in the research of pedestrian protection systems, such as the lack of public benchmarks and the difficulty to compare many of the proposed methods. They presented a more convenient strategy to survey the different approaches by dividing the pedestrian detecting problem into different processing steps. Enzweiler and Gavrila (2009) performed an elaborate experimental study of monocular vision based pedestrian detection by comparing the use of various features and classifiers. Their objective is to provide an overview of the current state of the art from both methodological and experimental perspectives. Their results indicated that a clear advantage of HOG/linSVM at higher image resolutions and lower processing speeds, and a superiority of the wavelet-based AdaBoost cascade approach at lower image resolutions and real-time processing speeds.

Although imaging sensors can provide abundant contents, they can not recover depth information, which is important for pedestrian collision avoidance application. To enhance the advantages of each sensor and overcome its limitations, there is a combination of different kinds of sensors that give complementary information. Bertozzi et al. (2008) introduced a system to detect and classify road obstacles fusing data of a camera, radar, and an inertial sensor. Vision was used to preliminary detect the presence of pedestrians in a specific region of interest. Results were merged with a set of regions of interest provided by a motion stereo technique. They stated that the vision based filtering provides an effective reduction of radar's false positives. Broggi et al. (2009a) triggered a non-reversible system by searching for pedestrians in specific areas with a laser scanner and a camera. As shown in detail in (Broggi, Cerri, Ghidoni, Grisleri, & Gi, 2009b), they focused on a specific urban scenario in which the detection of pedestrian is conducted, Their objective is to protect pedestrians who are hidden by parked vehicle or stopped bus, as well as those crossing the road between two stopped vehicles on the other side of the road. Scheunert, Cramer, Fardi, and Wanielik (2004) combined far infrared camera and laser scanner to obtain robust detection and accurate localization of pedestrians. Kalman filter based data fusion handled the combination of the outputs from laser scanner and far infrared camera. The European Union funded projects PROTECTOR and its successor SAVE-U were focused on reducing accidents involving vulnerable road users with fusion of different kinds of sensors (Michael & Andrzej, 2005). The California partners for advanced transit and highways (PATH) conducted research on transportation safety issues, including pedestrian protection, driver behavior modeling, and intersection collision prevention (Chan, Bu, & Steven, 2006). In their recent report on pedestrian detection, the performance and limitations of various products and technological approaches were investigated.

Fig. 1. DLUTIV-I Intelligent vehicle platform.

As can be seen form the literature review, vision based pedestrian detection is a main research field. The lack of explicit pedestrian models leads to the use of machine learning techniques, such as AdaBoost algorithm, SVM and CNN etc. With the help of such learning methods, an implicit representation is learned from features obtained from thousands of samples. The difference between each method is the features used and the learning method. This paper combines diverse features to train the classifiers during each stage considering different demands. Usually, the first stage in most recognition systems consists of identifying generic obstacles as regions of interests (ROIs) using prior knowledge with a computationally efficient method. Subsequently, a more expensive pattern recognition step is adopted. Therefore, this paper uses the simple Haar-like features to train cascaded classifiers with AdaBoost algorithm in the first stage. They are used to quickly segment pedestrian candidates by sliding a search window through the image. Then a more expensive SVM based classifying method is used to train the final classifier in the second stage, which helps to eliminate some false positives. The features used for SVM training are extracted from both the sample gray images and edge images.

3. The proposed pedestrian detection system

3.1. Overview of system architecture

The research of pedestrian detection is carried out on the DLUTIV-I intelligent vehicle platform, as shown in Fig. 1. This experimental platform is a modified Chery Tiggo, which is capable of automatic steering and braking. It is equipped with vision sensors, infrared sensor and laser scanner to percept vehicle surroundings. At the same time, a GPS and inertial navigation system are needed to determine its state. The final goal is to realize obstacle collision avoidance maneuvers and autonomous navigation under normal

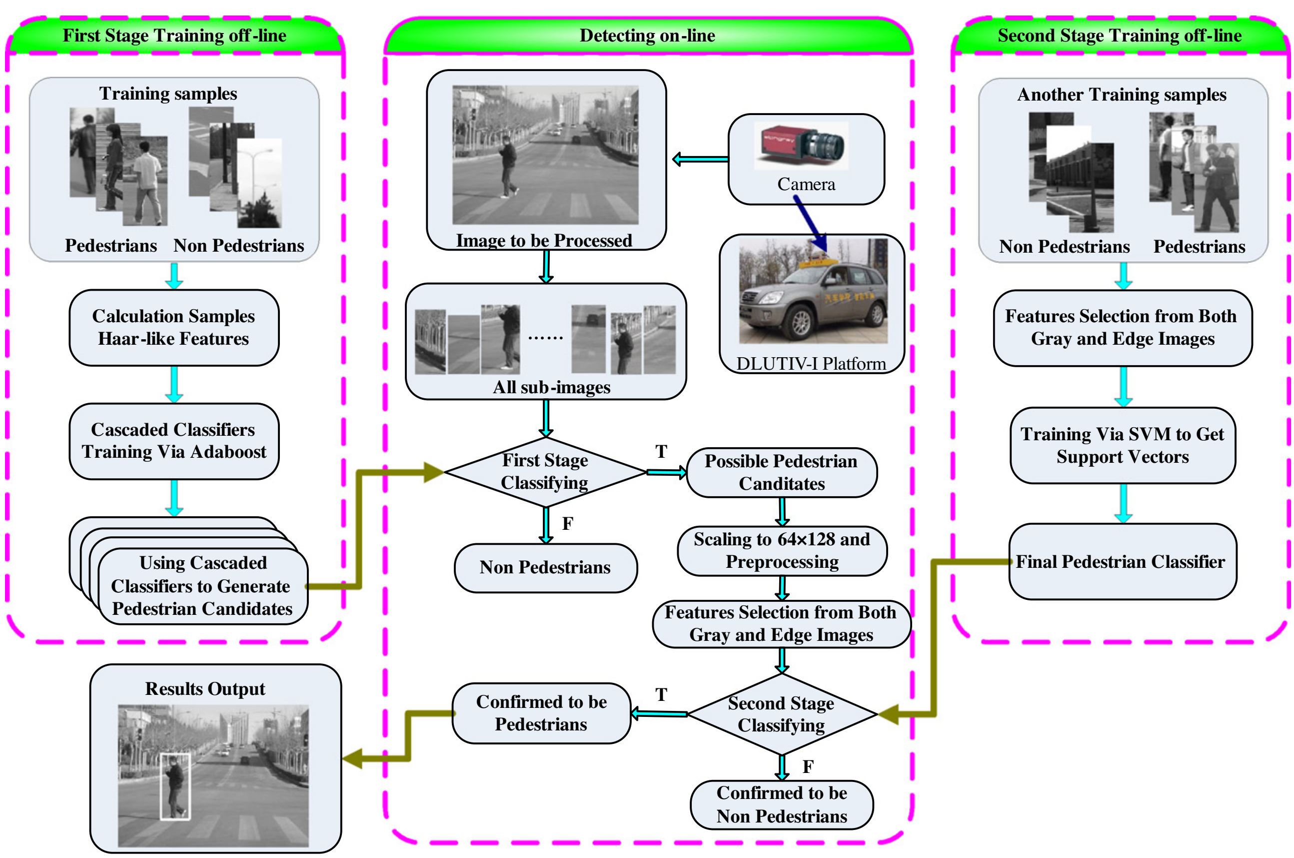


Fig. 2. Architecture of the proposed pedestrian detection system.

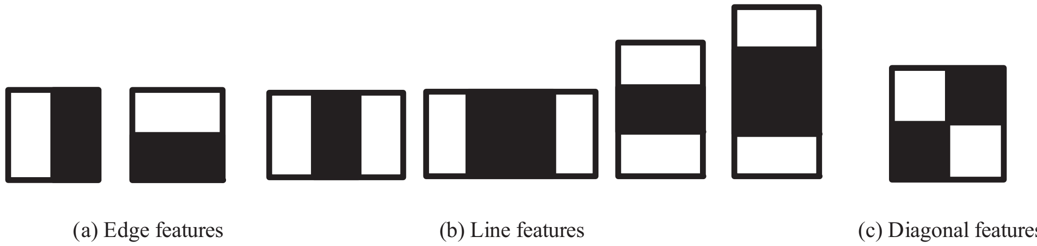


Fig. 3. Expression of Haar-like features.

urban situations. Pedestrian detection is an important sensing capability to achieve that goal.

The architecture of the proposed pedestrian detection system based on monocular vision is shown in Fig. 2, which includes offline training and online detection. The pedestrian classifiers are trained offline using the samples manual selected from different kinds of circumstances and loaded by the system to realize real-time pedestrian detection. The system is running on a Core2 2.66G PC and exhaustively search a 320×240 image captured by an AVT Stingray camera installed on the top front of DLUTIV-I. The first stage scans the whole image by sliding a search window at multiple scales. An initial set of possible pedestrian candidates are generated using a 20-stage cascaded classifiers trained by AdaBoost algorithm. Then an additional classifier is claimed to determine whether each candidate is pedestrian or not. Therefore a second stage is designed to eliminate some candidates that are believed to be false detections. In this stage, all pedestrian candidates are scaled to 64×128 and judged by another classifier trained with SVM. Only those candidates that successfully pass the two stages are considered to be pedestrians.

3.2. First stage pedestrian segmentation using cascaded classifiers

In this first stage, the cascaded classifiers are trained to generate pedestrian candidates. At the same time, a majority of sub-images that do not contain pedestrian features obviously are excluded. The training of cascaded classifiers includes feature selection and learning via AdaBoost algorithm.

3.2.1. Feature expression and calculation

In order to reduce the inter-class variability and make classification easier, the simple features instead of sample raw pixel values are selected as the input to a learning algorithm. Features usually encode knowledge about the object to be detected, which are difficult to learn from the raw and finite set of input data. This paper uses the simple Haar-like features inspired by Mohan et al. (2001) to train the cascaded classifiers. The Haar-like feature refers to a pair of rectangle, as shown in Fig. 3. They are simple 2D wavelet consisting of at least two non-overlapping rectangular regions, which express feature differences among adjacent areas. Those features can be calculated by the method proposed by Viola and Jones (2001). Further information about these features can be found in (Lienhart, Kuranov, & Pisarevsky, 2002). These simple features consist of significant domain-knowledge information, and the detector operates much faster with features compare to pixels. In the first pedestrian segmentation stage, some feasible Haar-like features representing pedestrian are selected by AdaBoost learning algorithm to efficiently reject non-pedestrian patterns.

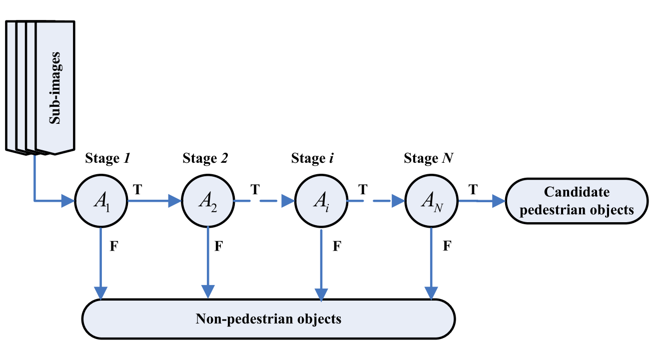


Fig. 4. Cascade structure with *N* stages.

By using integral image definition, the Haar-like feature value can be calculated as a weighted sum of the pixels within two components: the pixel gray level values sum over the black rectangle and the values sum over the white rectangle. Each feature is computed by summing up pixels within rectangles:

$$Feature_I = \sum_{i \in I = \{1,\ldots,N\}} \omega_i * RecSum(r_i) \quad (1)$$

where $\omega_i \in R$ is the weight of rectangle, $RecSum(r_i)$ is gray scale integral in rectangle r_i, and N is the number of rectangle.

3.2.2. Cascaded classifiers training by AdaBoost algorithm

In the first stage, the paper uses cascaded classifiers to segment possible pedestrian candidates. It can quickly remove most non-pedestrian sub-images and accelerate the detection algorithm. The Haar-like features are adopted as the basic elements to construct the cascaded classifiers, and each layer is trained by AdaBoost algorithm. The basic idea of AdaBoost algorithm is to use large capacity of general classification of the weak classifier by a certain method of cascade to form a strong classifier. It can automatically select the most discriminating features considering all possible feature types, sizes and locations.

The cascade structure containing *N* stages is illustrated in Fig. 4, where A_i is referred to as an AdaBoost classifier in the *i*th stage. As can be seen from the structure, the cascade classifier is a degenerated decision tree. At each stage, a classifier is trained to detect almost all interest pedestrians while rejecting a certain fraction of non-pedestrian objects. Therefore, negative sub-image that do not contain pedestrian can be discarded in some early stages of the cascade. Only the sub-image passing all stages is identified to be pedestrian. With this flexible structure, easily distinguish non-pedestrian objects like homogeneous texture ones can be simply rejected by simple Haar-like feature classifiers.

Each stage of the cascaded classifiers is trained using the discrete AdaBoost algorithm (Freund & Schapire, 1996). Detailed training procedure is described as following:

(1) Give training samples $S = \{(x_1,y_1),(x_2,y_2),\ldots(x_N,y_N)\}$, where $y_i = \{0,1\}$ corresponds to the non-pedestrian and pedestrian respectively. The total training samples consist of k non-pedestrian samples and l pedestrian samples.
(2) Initialize the weights $w_{1,i} = D(i)$. If the sample is non-pedestrian, the weight $D(i) = 1/2k$; if the sample is pedestrian, the weight $D(i) = 1/2l$.
(3) For $t = 1,2,\ldots,T$ (where T is the training number), doing following:

① Normalize the weights

$$q_{t,i} = \frac{w_{t,i}}{\sum_{j=1}^{N} w_{t,j}} \tag{2}$$

② For each feature, the corresponding weak classifier is trained as:

$$h_j(x) = \begin{cases} 1 & p_j f_j(x) < p_j \theta_j \\ 0 & otherwise \end{cases} \tag{3}$$

where p_j denotes the direction of the inequality sign, and its value only refers to ±1, $f_j(x)$ is feature value and θ_j is the threshold.

③ Choose the simple classifier $h_t(x)$ with the lowest error ε_t.

$$\varepsilon_j = \min_{f,p,\theta} \sum_i q_i |h_j(x_i) - y_i| \tag{4}$$

④ Update the weights according to the best simple classifier $h_t(x)$.

$$w_{t+1,i} = w_{t,i} \beta_t^{1-e_i} \tag{5}$$

where $e_i = 0$ if sample x_i is classified correctly, $e_i = 1$ otherwise, $\beta_t = \varepsilon_t / 1 - \varepsilon_t$.

(4) Finally, strong classifier is formed as:

$$R(x) = \begin{cases} 1 & \sum_{t=1}^{T} \partial_t h_t(x) \geqslant \frac{1}{2} \sum_{t=1}^{T} \partial_t \\ 0 & otherwise \end{cases} \quad where \ \partial_t = \log \frac{1}{\beta_t} \tag{6}$$

As can be seen from the training procedure, the boosting method can learn a strong classifier based on a set of weak classifiers by re-weighting the training samples. At each round of boosting, the feature-based classifier is added that best classifies the weighted training samples. With increasing of stage number, the number of weak classifiers increases, which are needed to achieve the desired false alarm rate at the given hit rate. At last, all the weak classifiers are combined to be a strong classifier by different weights.

3.2.3. Realization of pedestrian segmentation

In this stage, a total of 2080 hand labeled samples were adopted to train the cascaded classifiers. It includes 1100 pedestrian samples and 980 non-pedestrian samples. Pedestrian images with different size, pose, gait and clothing were collected to form the pedestrian samples. Non-pedestrian samples contain other objects except pedestrian, such as roads, vehicles, poles and other infrastructures. Each sample was resized to 16×32 pixels and its illumination is normalized to make the training samples the same size and illumination conditions. Fig. 5 shows some training samples.

This paper uses the open computer vision library developed by Intel to realize the features calculation and classifiers training offline. The total train stage is set to 20, during each stage, the minimum hit rate is 0.995 and the maximum false alarm rate is 0.5. According to these preferences, we got 20 strong classifiers and each strong classifier contains a number of weak classifiers. The weak classifier is constituted by feature, threshold, and the symbol indicating the direction of the inequality sign. With the increase of stage, the strong classifier contains more number of weak classifiers. Table 1 shows the type and number of each Haar-like feature of the first 10 strong classifiers trained to segment pedestrian

Fig. 5. Partial sample images used for training.

Table 1
Haar-like feature and fumber of the first 10 classifiers.

Stage	1	2	3	4	5	6	7	8	9	10
	Number									
Type	5	11	9	16	24	32	23	33	35	42
Edge feature	0	3	1	0	3	5	6	10	8	10
Line feature	3	4	3	6	5	9	10	12	10	13
Edge feature	1	1	1	2	5	8	2	3	6	4
Line feature	1	2	4	7	8	7	3	6	8	11
Diagonal feature	0	1	0	1	3	3	2	2	3	4

candidates. Fig. 6 is a visualization of the selected Haar-like features that make pedestrian distinguishable.

With the cascaded classifiers trained above, the pedestrian segmentation can be done by sliding a search window over the image at multiple scales and checking whether each sub-image is pedestrian or not. This image size to be processed is 320×240 pixels. During the segmentation, the minimum rectangle size to set to 16×32. Good results are obtained with scale factor of 1.25, indicates that window size increased at 25% rate between following scans. The white rectangles, as shown in Fig. 7, are the first stage pedestrian candidate segmentation results.

3.3. Second stage pedestrian classification based on SVM

The results of the first stage show that most possible pedestrian sub-images can be correctly segmented, whereas some false positives are generated. In order to reduce such false positives, something has to be done subsequently. Therefore, in the second stage, another classifier is trained to determine whether each candidate is pedestrian or not. There are several learning method used for object detection, such as SVM, AdaBoost and neural networks etc (Wang et al., 2009). Out of recent development in machine learning algorithms, SVM has shown prominent superiority, especially in pattern recognition applications, such as face detection(Hu & Zheng, 2009; Le & Satoh, 2004), 3D object recognition (Avidan, 2003). SVM is a learning algorithm developed by Vapnik (1995), which is based on the small sample statistical theory and the maximum class interval thought. In the binary classification case, the objective of SVM is to compute the optimal separating hyperplane between data points of two classes. Through choosing appropriate kernel function, SVM is able to map the misalignment separable input vectors from the input space to a higher dimensional feature space. Then a classification optimal hyperplane is established in the feature space. That is why this paper adopted SVM to train the final pedestrian classifier to reduce false positives.

3.3.1. Support vector machine

Given a set of points, which belong to either of two classes, SVM can find a hyperplane leaving the largest possible fraction of points of the same class on the same side, while maximizing the distance of either class from the hyperplane. This is equivalent to performing structural risk minimization to achieve good generalization, as opposed to empirical risk minimization commonly employed with other statistical methods (Burges, 1998). The structural risk minimization technique consists of finding the optimal separation surface between classes due to the identification of the most representative training samples called the support vectors. If the training dataset is not linearly separable, a kernel method is used to simulate a non-linear projection of the data in a higher dimensional space, where the classes are linearly separable. The main classification principle is briefed as following.

Let (x_i, y_i), $i = 1, \ldots, n$, $x_i \in R^d$ and $y_i \in \{+1, -1\}$ be the train data set, where $\{x_i\}$ are the input vectors and $\{y_i\}$-are the class labels. The decision boundary should classify all points correctly. Each hyperplane can be expressed as $(w \cdot x) + b = 0$. Thus, for linearly separable data, optimal hyperplane with minimum distance to the origin can be written as the following constrained optimization problem:

$$\begin{array}{ll} \min & \phi(w) = \frac{1}{2}\|w\|^2 = \frac{1}{2} w \cdot w \\ \text{subject to} & y_i[(w \cdot x_i) + b] - 1 \geqslant 0, \quad i = 1, \ldots, n \end{array} \tag{7}$$

The above optimization problem represents the minimization of a quadratic program under linear constraints. A convenient way to solve the constrained minimization problems is to transform the problem to its dual using Lagrange optimizing method, then solve the quadratic programming problem (Candade, 2004; Ivanciuc, 2007):

$$\begin{array}{ll} \max & \varphi(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) \\ \text{subject to} & \sum_{i=1}^{n} \alpha_i y_i = 0, \quad \alpha_i \geqslant 0, \quad i = 1, \ldots, n \end{array} \tag{8}$$

where α_i are the Lagrange multipliers. Generally, most α_i of the samples are zero, and the samples corresponding non-zero α_i are called support vectors. Then the following discrimination function is used to classify train samples:

$$f(x) = sgn\{(w \cdot x) + b\} = sgn\left\{ \sum_{i=1}^{n} \alpha_i^* y_i (x_i \cdot x) + b^* \right\} \tag{9}$$

As to linearly non-separable data, no linear classifier can be computed for the train set, but several hyperplanes can be calculated in such a way as to minimize the number of classification errors. To obtain an optimum linear classifier for those data, a penalty is introduced for misclassified data, denoted with ε and called a slack variable. Therefore, the optimization problem with a linear classifier and classification errors are:

$$\begin{array}{ll} \min & \phi(w, \varepsilon) = \frac{1}{2}\|w\|^2 + C\left(\sum_{i=1}^{n} \varepsilon_i\right) \\ \text{subject to} & y_i[(w \cdot x_i) + b] - 1 + \varepsilon_i \geqslant 0, \quad i = 1, \ldots, n \end{array} \tag{10}$$

Similarly, the above problem can be solved similar to the approach for linearly separable data based on Lagrange multipliers. However, the Lagrange multipliers are restrained by the penalty parameter as $0 \leqslant \alpha_i \leqslant C$. The penalty parameter C is a parameter that can be adjusted by the user, which can either increase or decrease the penalty for classification errors.

In reality, problems are seldom linearly separable. Thus, a non-linear kernel function that meets the Mercer's theorem (Cristianini

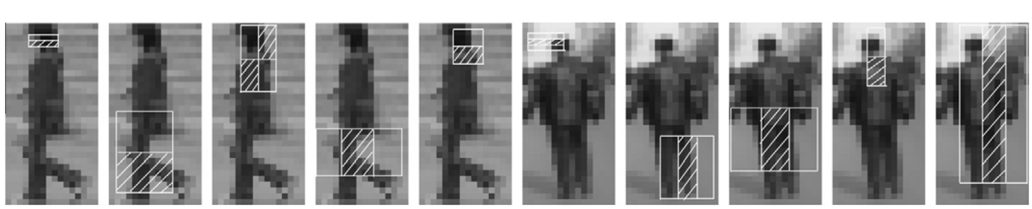


Fig. 6. Visualization of partial selected Haar-like features.

Fig. 7. Results of the pedestrian candidate segmentation.

& Taylor, 2000) is needed to map the input space to higher dimensions. In that higher dimensional space, linear classification can be used to solve a non-linear problem, as shown in Fig. 8. In Eq. (9), the inner dot product $(x_i \cdot x)$ can be replaced with a kernel function $K(x_i,x)$ that obeys Mercer's theorem, and the decision function of SVM is defined as:

$$f(x) = sgn\left\{\sum_{i=1}^{n} \alpha_i^* y_i K(x_i, x) + b^*\right\} \tag{11}$$

The commonly used kernel functions for SVM are the following three kinds: polynomial kernel, radial basis functions (RBF) kernel and Sigmoid kernel. These functions are computed in a high dimensional space and have a nonlinear character. They are mathematically defined in Eqs. (12)–(14).

Polynomial kernel:

$$K(x, x_i) = (x \cdot x_i + 1)^d \tag{12}$$

RBF kernel:

$$K(x, x_i) = \exp\left\{-\frac{|x - x_i|^2}{\sigma^2}\right\} \tag{13}$$

Sigmoid kernel:

$$K(x, x_i) = \tanh(\gamma(x \cdot x_i) + c) \tag{14}$$

There is no theory regarding which kernel is the best to a given recognition problem. It is important to select the appropriate kernel according to specific application.

3.3.2. Feature extraction for SVM training

Feature extraction is the first step in most object detection and pattern recognition algorithms. The ideal feature would be the one that can differentiate objects in the same category from objects in different categories. The number and kinds of features selected for pedestrian detection reported in literature varies with the training approach employed. There are many benefits of feature extraction: it facilitates data visualization and understanding, reduces the storage requirements, reduces training times and improves prediction performance. In order to eliminate the influence of pedestrian shape variability and illumination, this paper extracts the texture and symmetry features of pedestrian gray image, as well as the boundary moment and histogram of oriented gradients features of pedestrian edge image. The sample images as well as the pedestrian candidates are rescaling into a common size of 64×128 pixels.

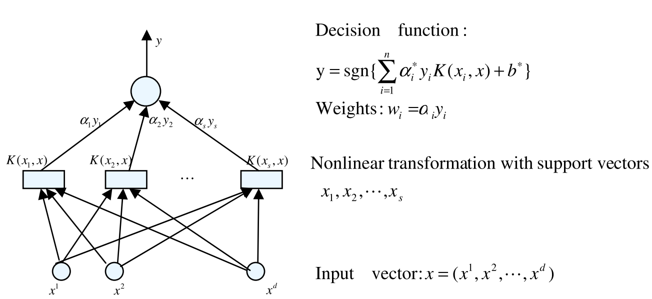


Fig. 8. Structure of SVM.

3.3.2.1. Feature extraction of sample gray images. Sometimes objects differ from the surrounding background, and each other, in texture but not in average brightness (Castleman, 1996). In this paper, the target is pedestrian. As can be seen form the gray image, the existence of pedestrian will conduct to certain texture features in the image. For instance, there exist strong texture features as the feet of pedestrian contact with the ground, the boundary of pedestrian head and shoulders with the background. While some noise, such as vehicles, trees adjacent to roads, or road constructions, their gray distribution are generally clutter and texture features are not regular. A commonly used texture analysis method is gray co-occurrence matrix (Haralick, 1979). Suppose that a direction and distance are established in an image. The i, jth element of the co-occurrence matrix p for an object is the times of appearance divided by M, where M is the number of pixel pairs contributing to p. The size of matrix p is L by L, where the image has L levels of gray. Once the co-occurrence matrix has been formed, texture features can be computed from it. A number of co-occurrence matrix-based features have been defined and tested. However, which of these features has discriminatory power for pedestrian classification must be determined by experiment results. Through comparison, this paper adopted the following four texture features to express a pedestrian: energy,

$$E = \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} (p(i,j)^2) \tag{15}$$

entropy,

$$H = -\sum_{i=0}^{L-1} \sum_{j=0}^{L-1} p(i,j) \;\; \lg \;\; p(i,j) \tag{16}$$

contrast,

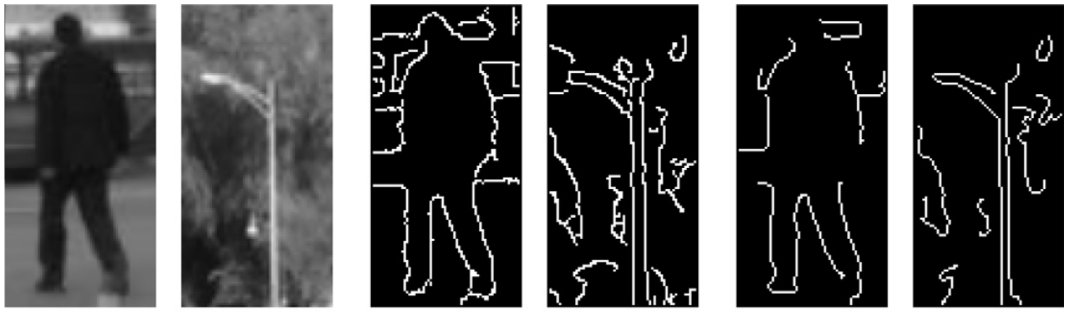
$$I = \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} (i-j)^2 p(i,j) \tag{17}$$

and local stability,

$$L = \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} p(i,j)/(1 + (i-j)^2) \tag{18}$$

Another feature extracted from sample gray images is the existence of strong symmetry, whether to standing or walking pedestrian. Suppose the gray value of a row in image is shown by one dimension of level pixel coordinates(x), it is $g(x)$. Any function is the sum of odd function $g_o(x)$ and even function $g_e(x)$, the symmetry is shown by their weights. The calculation of symmetry is defined as following:

$$s(x_s, w) = \frac{E_e'(x_s, w) - E_o(x_s, w)}{E_e'(x_s, w) + E_o(x_s, w)} \tag{19}$$



(a)Original sample image (b) Edge image (c) Boundary tracking

Fig. 9. Edge extraction of sample images.

where x_s is the level position of symmetry axis, w is the width of symmetry areas, E_o is the energy of odd function and E'_o is the normalized odd function, s is the measurement of symmetry. The range of symmetry measure is $(-1,1)$, when $s = 1$, it is a full symmetry, when $s = -1$, it is not symmetry. The symmetry depends on parameters x_s and w.

3.3.2.2. Feature extraction of sample edge images. Because pedestrians may appear in many colors, it is not advisable to use features based on color to detect pedestrians. With the idea of obtaining the typical silhouette of pedestrian and eliminating useless information for the classifier, this paper adopts the features extracted from sample edge images to explain pedestrian. Pedestrian in the image shows a certain degree of edge features, in particular, the edges of legs are more visible. The areas of non-pedestrian show irregular edges, such as vertical and horizontal edges. In addition, the edges features extracted from image can better reduce the influence of illumination.

Taking into account that Canny operator can achieve good balance between denoising and retaining the details edge (Canny, 1986). This paper uses Canny operator to abstract edges of sample images, as shown in Fig. 9(b). Canny operator chooses the following Gaussian function to process noise:

$$G(x) = \frac{1}{2\pi\sigma^2} e^{\frac{-x^2}{2\sigma^2}} \tag{20}$$

where σ is the standard deviation.

The gradient magnitude and gradient orientation are computed as

$$\begin{aligned} M(x,y) &= \sqrt{g_x^2 + g_y^2} \\ \theta(x,y) &= \arctan\frac{g_y}{g_x} \end{aligned} \tag{21}$$

where the gradient $g_x = f(x+1,y) - f(x-1,y), g_y = f(x,y+1) - f(x,y-1)$.

Fig. 9(b) shows that the edges of pedestrian can be extracted well. However, there exists some interference, such as the pedestrian head contact with the background. In order to reduce the computing complexity, the edge in the image is tracked and the length of the boundary is recorded to reserve those that its length is large than 20. The tracking results are shown in Fig. 9(c). The shape of pedestrian makes its boundary has certain contour and moment features. In order to explain the pedestrian boundary, this paper adopted the first two features of Hu's moment invariants for SVM training. The moment invariants can provide a scale, orientation and position invariant characterization of pedestrian shape (Chen, 1993).

Moreover, the histogram of gradient orientation is generated from the orientations in the gradient image. The orientations are transformed to the interval $[0, \pi)$ to avoid a difference in representation between black/white and white/black transitions. The weighted directions are sorted into histogram bins. The number of bins has to be chosen in a way so that noisy variations in direction are eliminated, but at the same time enough difference is uphold to discriminate between directions. This paper chooses to divide the gradient orientation into 8 bins evenly with the interval of $\pi/8$. The probability of each orientation belongs to each interval is calculated. Instead of forcing a single complex representation to capture the entire structure of the pedestrian, the candidate window is divided into sub-regions, as shown in Fig. 10(d). Therefore, a less complex representation can capture the class characteristics more easily in that specific region. The orientation histogram representation is ideal to work on small regions with only one major direction. The sub-regions I and II mainly contain the main directions of legs, therefore, this paper adopted the histogram of gradient orientation of each sub-region. The probability in each histogram is normalized and used as a feature vector for SVM training. The total feature number is 16 with each sub-region has 8 bins.

3.3.3. Realization of pedestrian classification

Pedestrian classification is performed using the classifier trained by SVM algorithms. Firstly, the train datasets are handled and features are extracted to form a training vector. Then, with proper SVM kernel function and parameters, SVM training is

(a)Original sample image (b) Gradient image (c) Gradient orientation (d) Sub-regions

Fig. 10. Sample image's gradient, gradient orientation and the sub-regions to be considered.

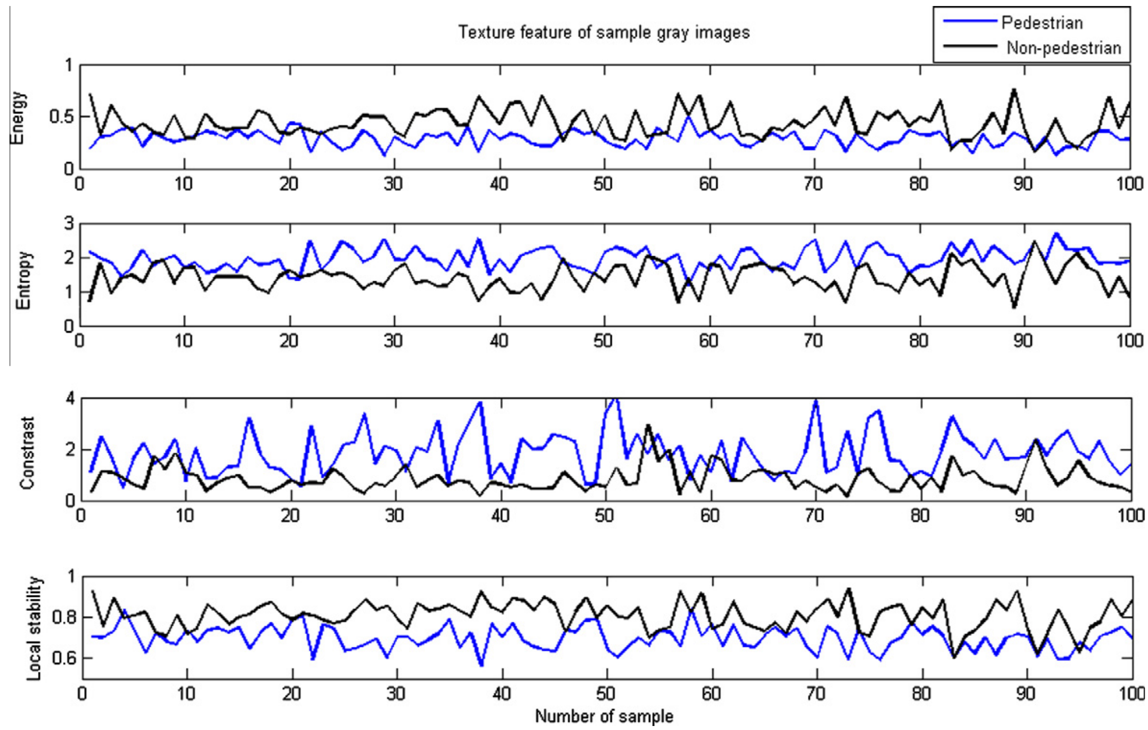


Fig. 11. Texture features of pedestrian and non-pedestrian sample images.

conducted to gain the support vectors, Lagrange multipliers and bias, which constitutes the pedestrian classifier. Finally, some test datasets are used to evaluate the classifier.

3.3.3.1. Description of feature vector for SVM training. Commonly, the supervised classification involves two stages: training and testing. The training datasets include 1100 pedestrian and 980 non-pedestrian samples and the testing datasets include 500 pedestrian and 500 non-pedestrian samples. The sample images are scaled to 64×128 pixel size and preprocessed to account for different lighting conditions and contrast. The feature vector x_i used for training includes four texture features and a symmetry feature, as well as two Hu's moment invariants and sixteen features of the gradient orientation histograms got from Section 3.3.2. Partial sample features are visualized in Figs. 11–13. Therefore each sample is expressed by a 23-dimensional vector x_i. The classifier output $y_i = 1$ if the image is pedestrian or $y_i = -1$ if it is a non-pedestrian image.

3.3.3.2. Selection of SVM kernel and parameters. The training for SVM needs to choose proper kernel function and penalty factor C, as well as parameters of kernel function. As introduced before, the commonly used SVM kernels are polynomial kernel, RBF kernel and Sigmoid kernel. In order to evaluate the performance and select the proper kernel function and parameters, the following evaluation criteria are defined.

Accuracy: probability of total number of predictions those are correct.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \times 100\% \qquad (22)$$

Pedestrian detection rate (PDR): probability of correctly classifying cases that a pedestrian sample is classified to be pedestrian.

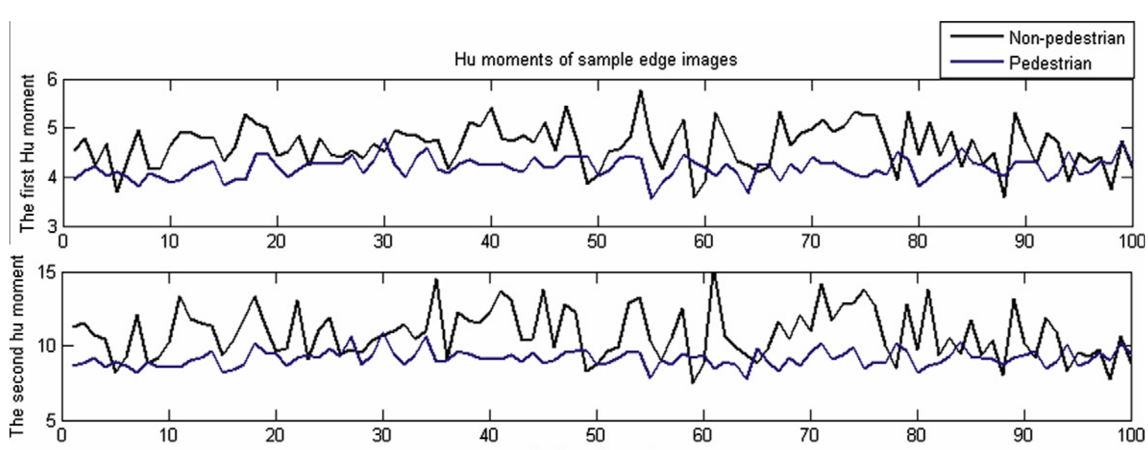


Fig. 12. Hu moment invariants features of pedestrian and non-pedestrian sample images.

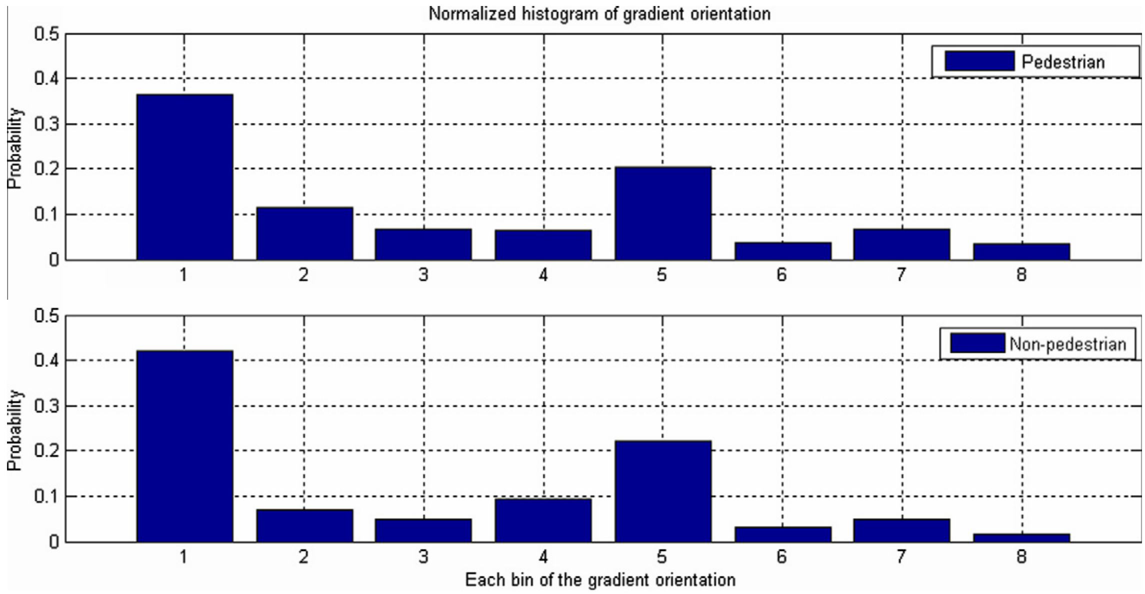


Fig. 13. Example of a normalized orientation histogram for sub-region I of pedestrian and non-pedestrian sample image. The interval $[0, \pi)$ is divided into eight bins on the x-axis, and the y-axis is normalized occurrence probability of that direction in the area.

$$PDR = \frac{TP}{TP + FN} \times 100\% \tag{23}$$

False alarm rate (FAR): probability of incorrectly classifying cases out of the samples classified to be pedestrians.

$$FAR = \frac{FP}{FP + TP} \times 100\% \tag{24}$$

where TP = number of correct predictions that a pedestrian sample is classified to be pedestrian; FP = number of incorrect predictions that a non-pedestrian sample is classified to be pedestrian; TN = number of correct predictions that a non-pedestrian sample is classified to be non-pedestrian; FN = number of incorrect predictions that a pedestrian sample is classified to be non-pedestrian.

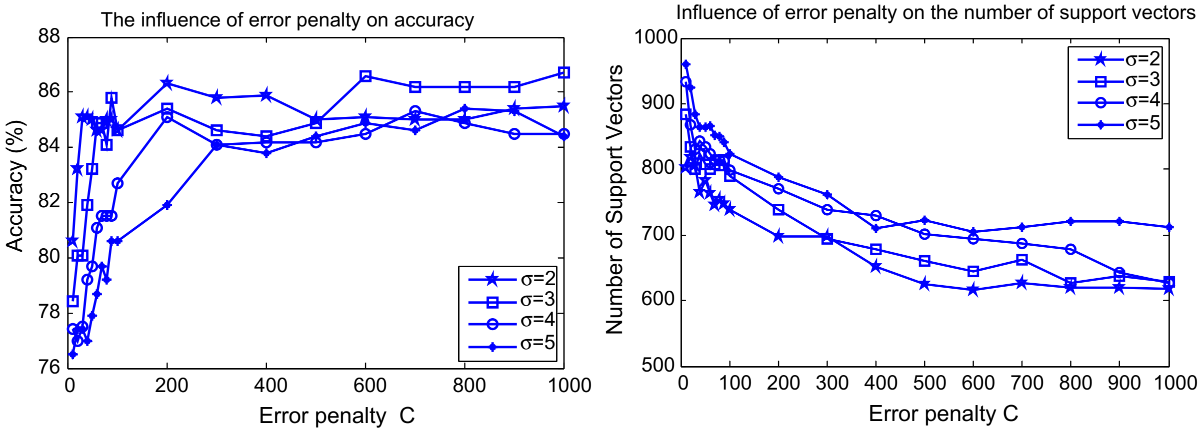
In fact, accuracy alone is not an adequate measure of performance especially when the number of negative cases is much greater than the number of positive cases. Thus it is important to study the other criteria described above. Pedestrian detection rate (PDR) refers to the proportion of pedestrian sample which is correctly classified to be pedestrian. False alarm rate (FAR) measures the fraction of false alarms that non-pedestrian is classified to be a pedestrian.

For the reason that kernel function defines the structure of high dimensional feature space where a maximal margin hyperplane will be found, the choice of kernel for SVM is important. This paper chooses the proper kernel function by comparing the results of different kernels. The training datasets consisted of 2080 samples, 1100 of which were pedestrians and 980 were non-pedestrians. The classifiers trained by different kernels were evaluated on the testing datasets, which consisted of 500 pedestrians and 500 non-pedestrians. Table 2 shows the number of the support vectors using different kernel functions with the training samples and its corresponding performance to classify the testing samples.

Table 2 shows that polynomial kernel function has the same performance with homogeneous polynomial kernel function. However, out of the three kernel functions, RBF kernel has the best predictions than other kernels. Expectations on the best classifier are to gain higher accuracy and PDR with lower FAR and less number of support vectors. For example, when RBF kernel $\sigma = 3$, the PDR is 88.6% and the FAR is 17.5%, both better than other kernels and even other parameters of the RBF kernel. Although the number of support vectors is a little higher than other situations. Therefore, this paper chooses the RBF kernel function to realize the input vectors dot product.

Table 2
Comparisons of different SVM kernel functions ($C = 1000$).

Kernel function	Parameter	Number of support vectors	Accuracy (%)	PDR (%)	FAR (%)
Homogeneous polynomial kernel: $K(x,x_i) = (x \cdot x_i)^d$	$d = 1$	566	82.1	87.6	21.6
	$d = 2$	474	81.9	81.2	17.9
	$d = 3$	400	81.8	82.0	18.3
	$d = 4$	316	81.2	82.0	19.2
Polynomial kernel: $K(x,x_i) = [(x \cdot x_i) + 1]^d$	$q = 1$	566	82.1	87.6	21.1
	$q = 2$	476	81.9	81.2	16.9
	$q = 3$	402	81.8	82.0	18.3
	$q = 4$	301	80.8	81.2	19.4
RBF kernel: $K(x, x_i) = \exp\left\{-\frac{\lvert x-x_i \rvert^2}{\sigma^2}\right\}$	$\sigma = 0.5$	606	78.5	83.6	25.4
	$\sigma = 1$	590	76.7	81.4	26.5
	$\sigma = 2$	625	85.1	88.0	16.9
	$\sigma = 3$	661	84.9	88.6	17.5
	$\sigma = 4$	702	84.2	85.2	16.4
	$\sigma = 5$	722	84.4	89.8	19.1



(a) Influence of C on accuracy

(b) Influence of C on the number of support vectors

Fig. 14. Influence of training parameters on the performance of classifier.

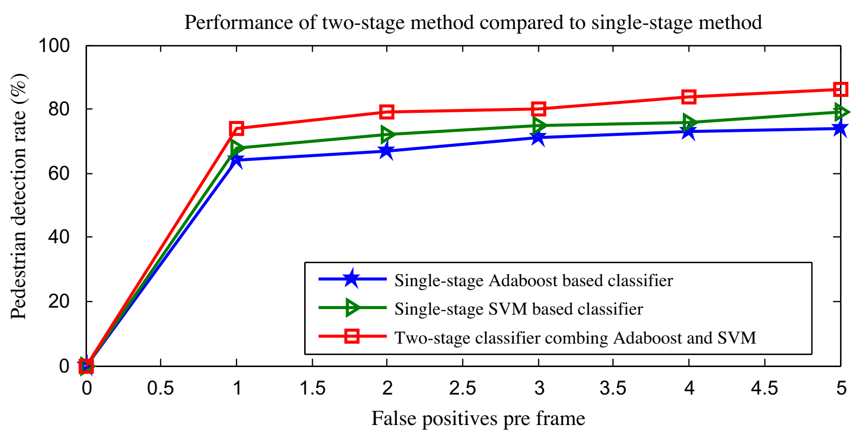


Fig. 15. Performance comparison of two-stage method with single-stage AdaBoost or SVM based method.

The performance of RBF kernel depends on the value of error penalty C and parameter σ. The error penalty C is tradeoff between error and margin. It is an upper bound of Lagrange multiplier α_i. Commonly, accuracy decreases as value of error penalty C decreases. A high value of error penalty would force the SVM training to avoid classification errors, thus resulting in a larger search space for the QP optimizer. The parameter σ of the RBF kernel, which is the radius of the RBF kernel, will also affect the performance of classifier. The final value of C and σ are selected through comparing their influence on the performance of classifier, as shown in Fig. 14.

According to the above comparisons, the final parameters are selected as follows: $\sigma = 3$, $C = 1000$, which can achieve high accuracy and lower number of support vectors. The trained pedestrian classifier consists of 628 support vectors, 319 of which are pedestrian samples and 309 are non-pedestrian samples. Using the testing datasets to validate the classifier, the PDR can achieve 89.4% with the FAR is 15.1%.

4. Experimental results

As explained in the previous section, the pedestrian segmentation and classification classifiers were trained offline using the samples hand labeled from different kinds of circumstances. To test the performance of the proposed pedestrian detection method, the system was integrated onboard the DLUTIV-I Intelligent vehicle platform for live testing. The system is running on a Core2 2.66G PC and exhaustively search a 320 × 240 image captured by an AVT Stingray camera installed on the top front of DLUTIV-I. The

(a) (b) (c) (d)

Fig. 16. Examples of some experiment results under urban scenarios.

pedestrian detection procedure works as following: Firstly, the pedestrian segmentation stage was executed to generate pedestrian candidates using the cascaded classifiers trained by AdaBoost algorithm. Then all pedestrian candidates were scaled to 64×128 and its features were extracted to form a 23-dimensional vector. In the second stage, each candidate was judged by the classifier trained with SVM. Only those that successfully passed the two stages were considered to be pedestrians.

Firstly, the proposed two-stage pedestrian detection method was compared with conventional single-stage classifier, such as AdaBoost algorithm based or SVM based classifier. The system was tested with an image sequence of multi-pedestrian walking ahead of the moving vehicle platform in rear and lateral view. The image sequence contained 486 frames, with 845 pedestrians, which was captured in urban scenarios under normal illumination. Fig. 15 shows the performance curves of the two-stage pedestrian detection method compared with single-stage AdaBoost based or SVM based method with similar features. The result indicates that performance of the proposed two-stage pedestrian detection method is superior to single-stage classifier trained with the same features. For example, the pedestrian detection rate can achieve about 80% at 2 false positive per frame, comparing with 64% of single-stage AdaBoost based method. The detection results also show that the system can achieve reliable pedestrian detection at a range of 30m and run at about 10-15Hz.

Fig. 16 shows some detection results of the proposed pedestrian detection system under different urban scenarios. Fig. 16(a) shows the result of single pedestrian detection. Fig. 16(b) and (c) are the results of multi-pedestrian detection. Although all the pedestrians can be correctly detected, there still exist some false positives, as shown in Fig. 16(d), the telegraph pole, which appears the same features as pedestrian, is likely to be classified as pedestrian. Furthermore, the occluding and overlapping of multi-pedestrian are other source of false alarms.

5. Conclusions

In this paper, a two-stage pedestrian detection system for intelligent transportation system is proposed to protect the vulnerable road users of pedestrian. During the first stage, some pedestrian candidates were generated using the cascaded classifiers trained by AdaBoost algorithm. At the same time, most non-pedestrian sub-images could be quickly eliminated, which is helpful to improve the detection time. At the second stage, all pedestrian candidates were scaled to 64×128 and its features were extracted and judged by the classifier trained with SVM. The features used to classify pedestrian included the texture and gray symmetry features of sample gray images, as well as the boundary moment and histogram of gradient orientation features of sample edge images. Testing results indicated that the two-stage detection method could achieve higher performance than single-stage detection methods. The proposed method is able to detect single and multi-pedestrian ahead of vehicle with different sizes and poses. The success of the proposed pedestrian detection method comes from its capability to extract most prominent features about pedestrian using both gray and edge images.

The current pedestrian detection system does not work well for pedestrians overlapped or occluded seriously. To avoid this kind of false detection, the different views and partial overlapped pedestrian samples can be added to the training datasets. Meanwhile, the performance of the detection system can be improved by obtaining more training samples. Furthermore, the pedestrian protection system has to warn the driver if there is a possible collision. So the system has to detect all the pedestrians around the vehicle, but in addition, has to analyze their activity and movement. In the future work, the pedestrian tracking method should be studied, which is helpful to pedestrian behavior modeling and prediction.

Acknowledgements

The research is financed by the National Natural Science Foundation of China 61104165 and the Fundamental Research Funds for the Central Universities through project 893308.

References

Alonso, I. P., Llorca, D. F., & Sotelo, M. A. (2007). Combination of feature extraction methods for SVM pedestrian detection. *IEEE Transactions on Intelligent Transportation Systems, 8*(2), 292–307.

Armingol, J. M., Escalera, A. D. L., Hilario, C., Collado, J. M., Carrasco, J. P., Flores, M. J., et al. (2007). IVVI: Intelligent vehicle based on visual information. *Robotics and Autonomous Systems, 55*(12), 904–916.

Avidan, S. (2003). Subset selection for efficient SVM tracking. In *Proceedings IEEE conference on computer vision and pattern recognition, Wisconsin, USA* (pp. 85–92).

Bertozzi, M., Bombini, L., Cerri, P., Medici, P., Antonello, P. C., & Miglietta, M. (2008). Obstacle detection and classification fusing radar and vision. In *Proceedings IEEE intelligent vehicles symposium, Eindhoven, Netherlands* (pp. 608–613).

Broggi, A., Cerri, P., Gatti, L., Grisleri, P., Jung, H. G., & Le, J. H. (2009a). Scenario-driven search for pedestrians aimed at triggering non-reversible systems. In *Proceedings IEEE intelligent vehicles symposium, Shaanxi, China* (pp. 285–291).

Broggi, A., Cerri, P., Ghidoni, S., Grisleri, P., & Gi, H. (2009b). A new approach to urban pedestrian detection for automatic braking. *IEEE Transactions on Intelligent Transportation Systems, 10*(4), 594–605.

Burges, C. (1998). Tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery, 2*(2), 955–974.

Candade, N. V. (2004). Application of support vector machines and neural networks in digital mammography: A comparative study. Thesis for Master degree, University of South Florida (pp. 30–76).

Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 8*(6), 679–698.

Castleman, K. R. (1996). *Digital image processing*. New Jersey, USA: Prentice Hall.

Chan, C. Y., Bu, F. P., & Steven, S. (2006). Experimental vehicle platform for pedestrian detection. California PATH Research Final Report for Task Order 5200, UCB-ITS-PRR-2006-16 (pp. 19–60).

Chen, C. C. (1993). Improved moment invariant for shape discrimination. *Pattern Recognition, 26*(5), 683–686.

Cheng, H., Zheng, N. N., & Qin, J. J. (2005). Pedestrian detection using sparse gabor filter and support vector machine. In *Proceedings IEEE intelligent vehicles symposium, Las Vegas, USA* (pp. 583–587).

Cristianini, N., & Taylor, J. S. (2000). *An introduction to support vector machines: and other kernel-based learning*. Cambridge, UK: Cambridge University Press.

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. In *IEEE computer society conference on computer vision and pattern recognition* (pp. 886–893).

Enzweiler, M., & Gavrila, D. M. (2009). Monocular pedestrian detection: Survey and experiments. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 31*(12), 2179–2195.

Freund, Y., & Schapire, R. E. (1996). Experiments with a new boosting algorithm. In *Proceedings of the 13th international conference on machine learning, San Francisco, USA* (pp. 148–156).

Gandhi, T., & Trivedi, M. M. (2007). Pedestrian protection systems: Issues, survey, and challenges. *IEEE Transactions on Intelligent Transportation Systems, 8*(3), 413–430.

Gerónimo, D., López, A. M., Sappa, A. D., & Graf, T. (2010). Survey of pedestrian detection for advanced driver assistance systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 32*(7), 1239–1258.

Haralick, R. M. (1979). Statistical and structural approaches to texture. *Proceedings of the IEEE, 67*(5), 786–804.

Hu, Y., & Zheng, G. (2009). Driver drowsiness detection with eyelid related parameters by support vector machine. *Expert Systems with Applications, 36*(4), 7651–7658.

Ivanciuc, O. (2007). Applications of support vector machines in chemistry. *Reviews in Computational Chemistry, 23*, 291–400.

Le, D. D., & Satoh, S. (2004). Feature selection by Adaboost for SVM-based face detection. In *Forum on information technology* (pp. 183–186).

Lienhart, R., Kuranov, A., & Pisarevsky, V. (2002). Empirical analysis of detection cascade of boosted classifiers for rapid objects detection. MRL Technical Report, Intel Corporation.

Llorca, D. F., Sotelo, M. A., Parra, I., Ocaña, M., & Bergasa, L. M. (2010). Error analysis in a stereo vision-based pedestrian detection sensor for collision avoidance applications. *Sensors, 10*(4), 3741–3758.

Michael, M. M., & Andrzej, O. M. (2005). Enhancing pedestrian safety by using the SAVE-U pre-crash system, In *IEEE World Congress on ITS, San Francisco, USA*.

Mohan, A., Papageorgiou, C., & Poggio, T. (2001). Example-based object detection in images by components. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 23*(4), 349–361.

Mohan, D. (2002). Traffic safety and health in Indian cities. *Journal of Transport and Infrastructure, 9*(1), 79–94.
Mohan, D., Tsimhoni, O., Sivak, M., & Flannagan, M. J. (2009). Road safety in India: Challenges and opportunities. *UMTRi-2009-1*, 1–57.
Papageorgiou, C., & Poggio, T. (2000). A trainable system for object detection. *International Journal of Computer Vision, 38*(1), 15–33.
Peden, M., Scurfield, R., Sleet, D., Mohan, D., Hyder, A. A., Jarawan, E., et al. (2004). *World report on road traffic injury prevention*. Switzerland: World Health Organization.
Sabzmeydani, P., & Mori, G. (2007). Detecting pedestrians by learning shapelet features. In *Proceedings of IEEE conference on computer vision and pattern recognition, Minneapolis, USA* (pp. 1–8).
Scheunert, U., Cramer, H., Fardi, B., & Wanielik, G. (2004). Multi-sensor based tracking of pedestrians: A survey of suitable movement models. In *Proceedings of IEEE intelligent vehicles symposium, Parma, Italy* (pp. 774–778).
Szarvas, M., Sakai, U., & Ogata, J. (2006). Real-time pedestrian detection using lidar and convolutional neural networks. In *Proceedings of IEEE intelligent vehicles symposium, Tokyo, Japan* (pp. 213–218).
Vallejo, D., Albusac, J., Jimenez, L., Gonzalez, C., & Moreno, J. (2009). A cognitive surveillance system for detecting incorrect traffic behaviors. *Expert Systems with Applications, 36*(7), 10503–10511.
Vapnik, V. N. (1995). *The nature of statistical learning theory*. New York, USA: Springer.
Viola, P., & Jones, M. J. (2001). Rapid object detection using a boosted cascade of simple features. In *IEEE computer society conference on computer vision and pattern recognition, Kauai, HI, USA* (pp. 511–518).
Wang, S. J., Mathew, A., Chen, Y., Xi, L. F., Ma, L., & Lee, J. (2009). Empirical analysis of support vector machine ensemble classifiers. *Expert Systems with Applications, 36*(3), 6466–6476.