Robust Vehicle Detection Based on Cascade Classifier in Traffic Surveillance System

Y. Tang[1,*], Y. C. Xu[1] and C. Z. Zhang[2]

[1]*School of Automobile and Traffic Engineering, Nanjing Forestry University, Nanjing , 210037， P.R. China*

[2]*Guangzhou Smartspyeye Technologies Company, Guangzhou, 213000， P.R. China*

Abstract: Vehicle detection based on static images are highly practical and directly applicable for vehicle feature extraction and recognition in a traffic surveillance system. This paper will introduce the processing of automatic vehicle detection based on machine learning algorithm. Firstly, Haar-like feature is used to represent the appearance of vehicle, and then a learning algorithm, based on AdaBoost, is to train the strong classifier, at last a method for combining strong classifiers in a cascade is proposed, which allows background regions of the image can be removed quickly. The experimental result shows that the our classifier can achieve good performance of vehicle detection, its detection rate is more than 97% and its false alarm rate is only 3.4%.

Keywords: AdaBoost algorithm, Cascade, Haar-like feature, Intelligent transportation vehicle detection.

1. INTRODUCTION

A traffic surveillance camera system is an important part of an intelligent transportation system [1]. It mainly includes automatic monitoring digital cameras to take snapshots of passing vehicles and other moving objects, as is shown in Fig. (**1**). The recorded images are high-resolution static images, which can provide valuable clues for police and other security departments, such as a vehicle plate number, the time it passed, its movement path and the driver’s face, etc. In prior days, massive amounts of stored images were processed manually, but this required hard work and resulted in poor efficiency. With the rapid development of computer technology, the latest in automatic license plate recognition software is utilized at an increasing rate in the field with great success [2]. Unfortunately, sometimes we may not discover the license plate of a vehicle because of cloned license plates, missing license plates, or because the license plate can’t be recognized. This is why automatic vehicle detection and recognition is becoming the imminent requirement for traffic surveillance applications [3]. This technology will save a lot of time and effort for users trying to identify blacklisted vehicles or who are searching for specific vehicles from a large surveillance image database [4, 5].

Similar to face detection for face recognition, vehicle detection is the first step for vehicle feature extraction and recognition. Accuracy of vehicle detection directly impacts rate of vehicle type recognition. In order to detects a vehicle in a static image, almost all researchers make use of license plate locations to extract the vehicle area from the image [6, 7]. However, this is not a valid technique when there are vehicles with non-symmetrical front license plates and the vehicle contour may not be accurate enough for some types of vehicles, particularly those that are either larger or smaller than the average. This paper proposes a robust vehicle detection scheme based on an AdaBoost algorithm. Different vehicle image have same features, including bumper, windshield, light and so on. In this paper, an approach for AdaBoost vehicle detection using Haar-like features is proposed, for reference of face detection, which could be reached higher accuracy rate and lower false alarm rate for vehicle detection The basic idea is to extract the Haar-like features from vehicle samples and then use the AdaBoost algorithm to train classifiers for detection, which is distinct from previous research on vehicle detection for static images.

2. HAAR-LIKE FEATURE

2.1. Definition of Haar-like Feature

A Haar-like feature is well known as a local texture descriptor, which can be used to describe the local appearance of an object [8]. A Haar-like feature measures the average pixel intensity differences of the rectangular regions, as is shown in Fig. (**2**). The value of a Haar-like feature is that it defines the difference between the sum of the pixel values of the black region and the white region. When we change the position, size, shape, and the arrangement of the Haar-like temple, the object feature information, such as the intensity gradient, edge, or contour can be captured. As is shown in Fig. (**2**), the vehicle image includes saliency rectangle characteristics, where the Haar-like feature is especially suitable for description purposes.

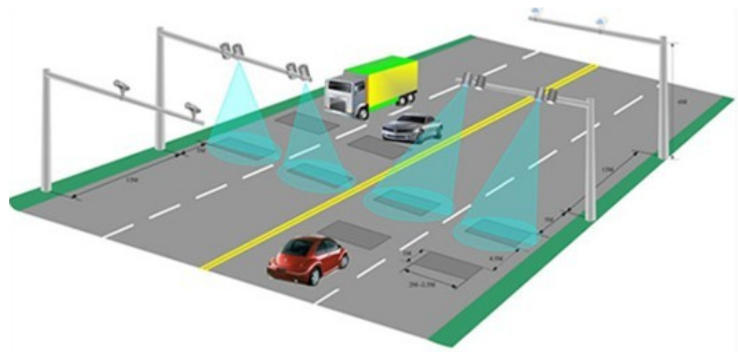

Fig (1). Traffic surveillance camera system.

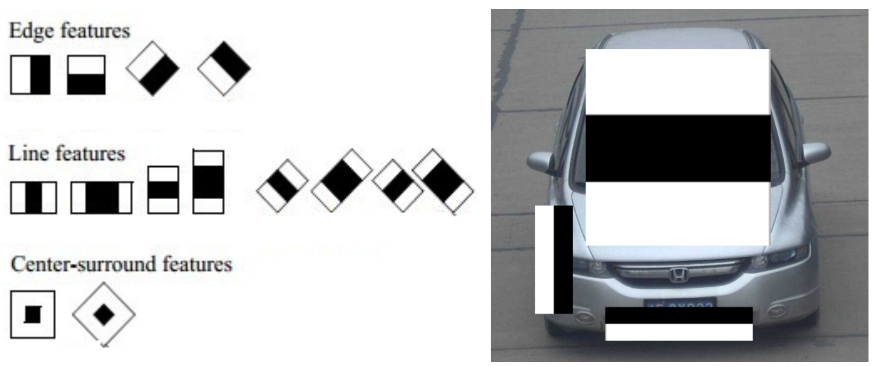

Fig (2) Haar-like feature.

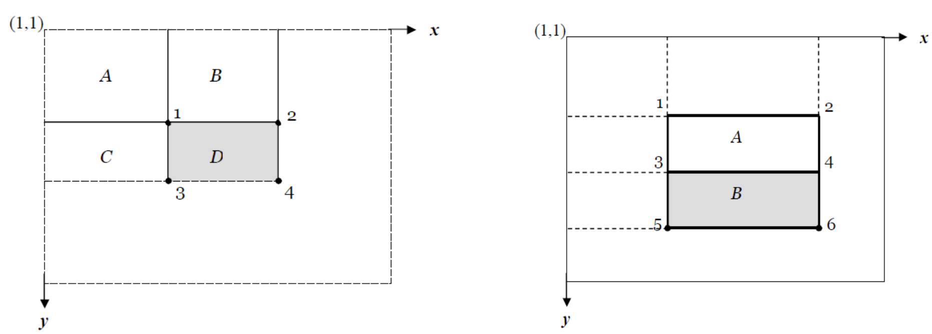

Fig (3) Integral image.

2.2. Integral Image

In order reduce compute time of the Haar-like feature, integral image [9] can be adopt to as th intermediate representation. The integral image at location (x,y), is the sum of the pixel values above and to the left of (x,y), inclusive.

$$I'(x,y)=\sum_{i=0}^{i\le x}\sum_{j=0}^{j\le y} I(x,y) \tag{1}$$

Any rectangular sum can be computed in four array references, when we use the integral image as shown in Fig. (**3a**), the value of rectangular sum can be generated as follow formula:

$$Sum(R)=I_1'+I_4'-(I_2'+I_3') \tag{2}$$

Given a edge feature as shown in Fig. (**3b**), the value of this feature can be computed as:

$$\begin{aligned} f &= Sum(A)-Sum(B)=[I'(4)+I'(1)- \\ &I'(2)-I'(3)]-[I'(6)+I'(3)-I'(4)-I'(5)] \end{aligned} \tag{3}$$

The value of each feature can be generated only use add and subtract with constant time from formula (3). For rotated features, a separate rotated integral image must be computed, as Lienhart [10] describes.

3. ADABOOST ALGORITHM

3.1. Algorithm

The purpose of the AdaBoost algorithm is to use the feature (i.e., the Haar-like feature previously discussed) to dis-

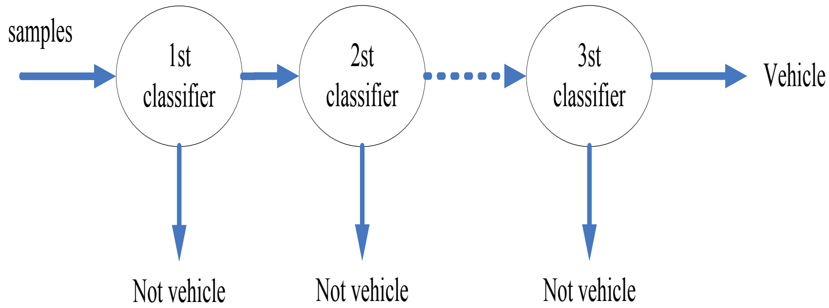

Fig (4): The cascade classifiers architecture.

cover the best weak classifiers, and then combining these weak classifiers to a strong classifier after a series of training exercises with a huge set of positive and negative samples of a target object. After long time training, the strong classifier can be applied to detect the target object.

The Adaboost algorithm can be described as follow:

A. Giving a set of samples $(x_1, y_1), (x_2, y_2), \ldots (x_n, y_n)$, While, $y_i = 1$ denotes a positive sample (vehicle) $y_i = 0$ denotes a negative sample (non vehicle). n is the number of samples.

B. Normalize the weights $w_{1,i} = D(i)$

C. For $t = 1,2,3 \ldots T$:

(1). Normalize the weights:

$$q_{t,i} = \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}}$$

(2). For each feature f, Training a weak classifier $h(x, f, p, \theta)$, and generate the weight sum of error rate $\varepsilon_f = \sum_i q_i |h(x_i, f, p, \theta) - y_i|$. The weak classifier $h(x, f, p, \theta)$ is defined as:

$$h(x, f, p, \theta) = \begin{cases} 1 & pf(x) < p\theta \\ 0 & otherwise \end{cases}$$

(3). Choose the best weak classifier $h_t(x)$ (with the lowest error ε_t): $\varepsilon_t = \min_{f,p,\theta} \sum_i q_i |h(x_i, f, p, \theta) - y_i|$

(4). Update the weights: $w_{t+1,i} = w_{t,i} \beta_t^{1-e_i}$.

D. The last strong classifier is:

$$H(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t, \alpha_t = \log \frac{1}{\beta_t} \\ 0 & \text{otherwise} \end{cases}$$

3.2. Weak Classifier

The detection ability of strong classifier depend on the weak classifier, thus the weak learning algorithm is designed to select a rectangle feature that can best separates the positive and negative samples. For each Haar-like feature, the weak learner algorithm should find the optimal threshold for classification, assure that the minimum number of samples are misclassified. The weak classifier consists of feature f, threshold θ and parity p. p stands for the direction of the inequality sign, as shown in formula (4):

$$h(x, f, p, \theta) = \begin{cases} 1 & pf(x) < p\theta \\ 0 & otherwise \end{cases} \quad (4)$$

3.3. Cascade Algorithm

To further reduce the false alarm rate, a cascade of classifier is proposed by Viola P [11], The detection process is that of a degenerate decision tree, which is shown in Fig. (**4**). A positive result from the first classifier triggers the evaluation of a second classifier which has also been adjusted to achieve very high detection rates. If a sample is recognized to be negative via front classifier, which will be removed from the training sample set and will not enter the next stage of classifier. Thus number of samples decreases quickly with the increase of stages. If a sample is determined to be positive, it will enter the next stage of classification until the last stage, and last it will be classified as a positive sample.

Training algorithm for constructing cascade-Adaboost classifier as followed:

A. Given the maximum false alarm rate f and the minimum detection rate d for per stage.

B. Given target over false alarm rate F.

C. P is set of positive examples.

D. N is set of negative examples.

E. Set $F_i > F_0, D_0 = 1$; i=0, i is the stage index

F. While $F_i > F$

(1). $i = i + 1; n_i = 0; F_i = F_{i-1}$

(2). While $F_i > f \times F_{i-1}$

$n_i = n_i + 1$

& Training a strong classifier with n_i feature with P and N

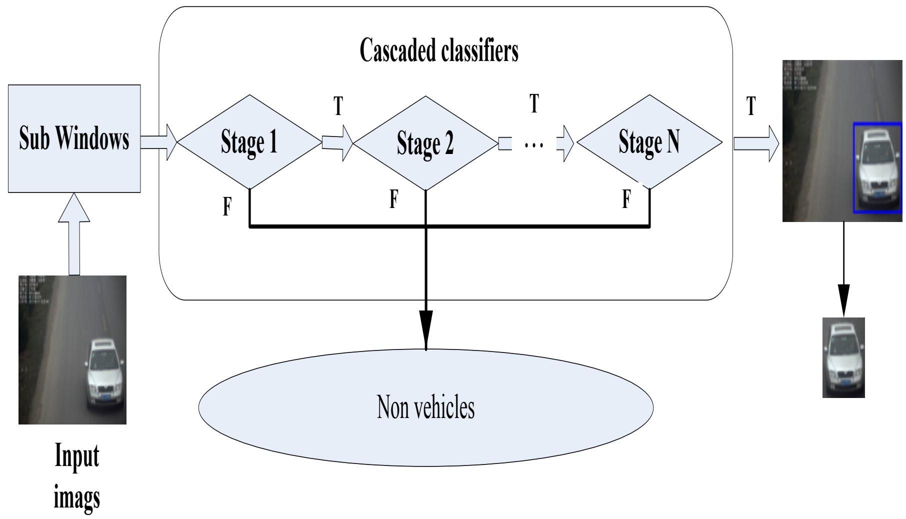

Fig (5) Vehicle detection.

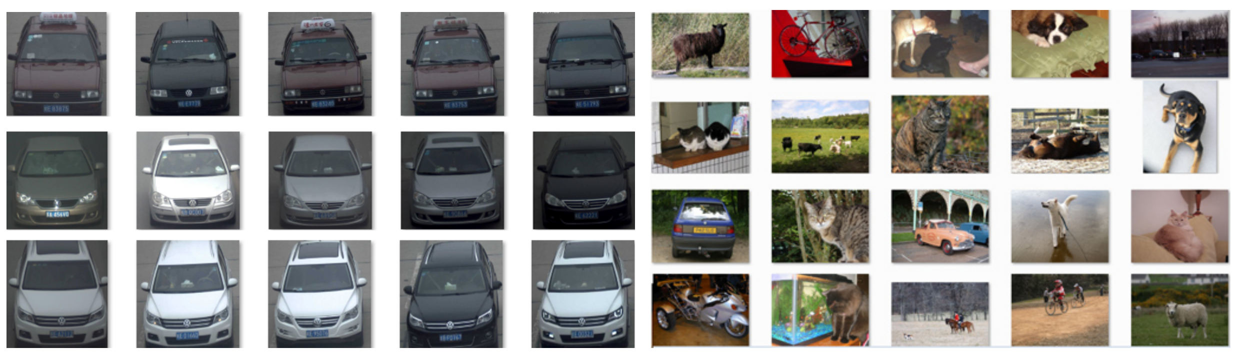

Fig (6). Examples of the positive samples and negetive sanples.

& Evaluate current cascade classifier on validation set to determine F_i , D_i

& Decease threshold for i th classifier until the current cascade classifier's detection rate at least of $d \times D_{i-1}$

(3). N is empty.

First we set detection rate d , the false alarm rate f for each stage and the overall target false rate F . And then assume the sets of positive samples and negative samples are called P and N . The cascade algorithm is mainly composed of two loops. Weak classifier is train and added to strong classifier in internal loop via above mentioned Adaboost Algorithm, until it has reached the expected target f and d at current stage. Then, determine whether the false alarm rate F_i of the cascade classifier is below F . If bellow, terminate training process. If not, update negative samples and go back to external loop to train Adaboost classifier of the next stage until the overall false alarm rate of cascade classifier is below F .

4. VEHICLE DETECTING

The detection of vehicle is done by sliding a sub-window across the image at multiple scales and locations. In order to reduce detection time, scaling is achieved by changing the detector itself but not scaling the image to ensure the same computational cost at any scale. The initial size of the detector is 15x15, the detector window is scaled at 1.2, and transform step is 2 pixels. Using the cascade classifier obtained in the previous approach, we can decide whether an image region at certain location is classified as vehicle or non vehicle , as shown in Fig. (**5**).

5. EXPERIMENT

This section will introduce the results of the vehicle detection efforts. Our results are all calculated on a desktop computer with an Intel Core i7, 3.4 GHz CPU, and 4 GB RAM.

A total of 5,000 vehicle images were used in the training procedure. The positive samples were obtained by manually cropping the vehicle area of the vehicle images recorded with traffic surveillance cameras. All samples were resized to 30x30 pixels, as shown in Fig. (**6**). In order to get sufficient negative samples, we download several image dataset

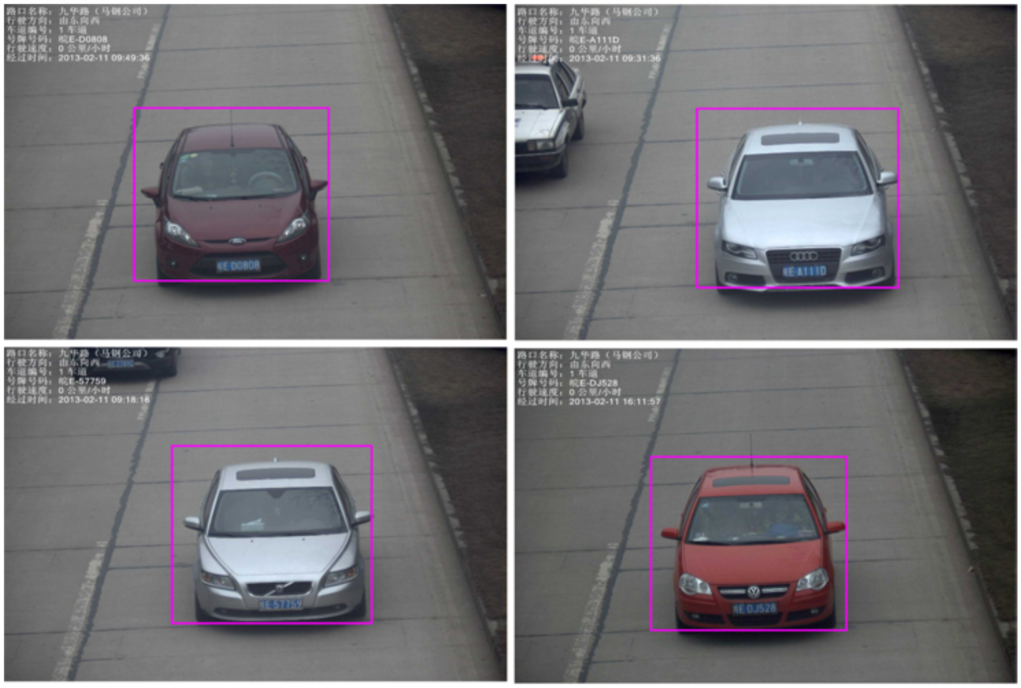

Fig (7). Images of vehicle detection.

from the net, which resulted in more than 15,000 negative samples, at least. In the training processing, the maximum detection rate was set as 99.5%, and the minimum false rate was 50%.

To get the numerical result of vehicle detections, we defined the total detection rate (TDR), the detection rate (DR), and the false rate (FAR).

$$TDR = \frac{TP+TN}{P+N}\times 100\% \qquad DR = \frac{TP}{P}\times 100\%$$

$$FAR = \frac{FP}{N}\times 100\% \tag{5}$$

Where P is the number of positive samples and N is the number of negative samples or the non-vehicle samples. Where TP stands for the number of vehicle images detected correctly, TN stands for the number of non-vehicle images detected correctly, and FP stands for the number of non-vehicle samples detected to be vehicle images. So, the higher the DR value and the lower the FAR value are, the better the result will prove the efficiency of this detection method. The results of the experiment are displayed in Table **1**. The image of the vehicle detection result is shown in Fig. (**7**).

There are 820 samples in our test and the total time it took to deal with these images is about 43 sec. Thus, the average detection time for each image is 53 milliseconds. As is shown in Table **1**, we achieved a high detection rate, which is higher than the previous method; given more samples, it is likely that the rate may be improved.

Table 1. Detection Result.

P	N	TP	TN	FP	TDR	DR	FAR
517	203	503	197	7	97.2%	97.3%	3.4%

CONCLUSION

Accurate and robust vehicle detection still a challenging task in the field of intelligent transportation surveillance systems. In this paper, we presented a cascade of boosted classifiers based on the characteristics of the vehicle images to be used for vehicle detection in on-road scene images. Haar-like features and an AdaBoost algorithm were used to construct the classifier for the vehicle detection, which is distinct from previous research published on vehicle detection. We have tested this method on a realistic data set of over 500 frontal images of cars that were used for vehicle detection, which achieved a high accuracy of 97.3%. In future, we will do further research on different AdaBoost versions such as Modest AdaBoost, Gentle AdaBoost etc, and more type of other feature, to further improve the efficiency of vehicle detection.

CONFLICT OF INTEREST

The author confirms that this article content has no conflict of interest.

ACKNOWLEDGEMENTS

The research has been supported by the fund of high academic qualification in Nanjing forestry university (GXL201315) in China, natural science fund for colleges and universities in Jiangsu Province (14KJB520017).

REFERENCES

[1] B. Zhang, Y. Zhou, H. Pan. "Vehicle classification with confidence by classified vector quantization." *Intelligent Transportation Systems Magazine, IEEE,* vol. 3, pp. 8-20, May. 2013.

[2] S. Du, M Ibrahim, M. Shehata, and W. Badawy, "Automatic license plate recognition (ALPR): a state-of-the-art review." *IEEE*

Transactions on Circuits and Systems for Video Technology, vol. 23, pp. 311-325, Feb. 2013.

[3] S. Sivaraman, and M. Trivedi. "General active-learning framework for onroad vehicle recognition and tracking". *IEEE Transaction on Intelligent Transportation System*, vol. 11, no. 2, pp. 267-276, Jun. 2010.

[4] B. L. Zhang. "Reliable classification of vehicle types based on cascade classifier ensembles." *IEEE Transactions on Intelligent Transportation Systems*, vol 14, pp. 322-332, Jan. 2013.

[5] G. Pearce and N. Pears. "Automatic make and model recognition fromfrontal images of cars." In: *Proceeding of 8th IEEE International Conference of AVSS. Klagenfurt*, Austria. pp. 373-378, 2011

[6] A. Psyllos, and C. N. Anagnostopoulos, E. Kayafas. "Vehicle model recognition from frontal view image measurements". *Computer Standards & Interfaces*. vol. 33. pp. 142-151, Feb. 2011.

[7] B. Zhang, and Y. Zhou. "Vehicle type and make recognition by combined features and rotation forest ensemble." *International Journal of Pattern Recognition and Artificial Intelligence*. vol. 26, pp. Mar. 2012.

[8] K. Y. Park, and S. Y. Hwang. "An improved Haar-like feature for efficient object detection." *Pattern Recognition Letters*. vol. 42: pp. 148-153. 2014.

[9] Crow F C. "Summed-area tables for texture mapping." *ACM SIGGRAPH Computer Graphics*. pp. 207-212, 1984.

[10] R. Lienhart and J. Maydt. "An extended set of Haar-like features for rapid object detection." *In IEEE International Conference on Image Processing*. USA pp. 900-903, 2002.

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

Received: September 22, 2014 Revised: November 03, 2014 Accepted: November 06, 2014