

Boosted Vehicle Detection Using Local and Global Features

Chin-Teng Lin¹, Sheng-Chih Hsu¹, Ja-Fan Lee², Chien-Ting Yang¹

¹Institute of Electrical and Control Engineering, College of Electrical and Computer Engineering of the National Chiao-Tung University, Hsinchu, Chinese Taipei; ²Institute of Biomedical Engineering, College of Computer Science of the National Chiao-Tung University, Hsinchu, Chinese Taipei.
Email: a58705077@gmail.com

Received June 16th, 2013; revised July 18th, 2013; accepted July 26th, 2013

Copyright © 2013 Chin-Teng Lin *et al.* This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

ABSTRACT

This study presents a boosted vehicle detection system. It first hypothesizes potential locations of vehicles to reduce the computational costs by a statistic of the edge intensity and symmetry, then verifies the accuracy of the hypotheses using AdaBoost and Probabilistic Decision-Based Neural Network (PDBNN) classifiers, which exploit local and global features of vehicles, respectively. The combination of 2 classifiers can be used to learn the complementary relationship between local and global features, and it gains an extremely low false positive rate while maintaining a high detection rate. For the MIT Center for Biological & Computational Learning (CBCL) database, a 96.3% detection rate leads to a false alarm rate of approximately 0.0013%. The objective of this study is to extract the characteristic of vehicles in both local- and global-orientation, and model the implicit invariance of vehicles. This boosted approach provides a more effective solution to handle the problems encountered by conventional background-based detection systems. The experimental results of this study prove that the proposed system achieves good performance in detecting vehicles without background information. The implemented system also extract useful traffic information that can be used for further processing, such as tracking, counting, classification, and recognition.

Keywords: Vehicle Detection; AdaBoost; Probabilistic Decision-Based Neural Network (PDBNN); Gaussian Mixture Model (GMM)

1. Introduction

Recent research in computer vision has involved the use of video processing to improve human safety in various areas. For example, advanced techniques for improving and controlling traffic conditions are top priorities in developed countries. Traditional traffic surveillance systems often use sensors to detect the passage of vehicles and gather simple information. Other systems use cameras and manual video review when incidents occur. These methods are inefficient in information extraction, and warrant improvement. Therefore, vision-based Intelligent Transportation Systems (ITSs) have recently attracted great attention in computer vision research. For example, many systems now use video-based automatic surveillance, behavior analysis for accident prediction, and traffic monitoring.

For most of these application systems, foreground object extraction is a fundamental and crucial step that must occur before object tracking, classification, and recogni-

tion. Foreground segmentation typically involves the use of conventional background subtraction and temporal difference methods. However, the various factors that may affect the results of foreground segmentation make foreground detection a challenging task. Background subtraction, which relies on a background to extract foreground objects in a frame, is inefficient in certain circumstances. For example, swaying trees might be treated as foreground objects although they are not the objects of interest. Moreover, background construction relies on the frequent appearance of true information; the background of a road scene is difficult to construct when traffic is heavy because the frequency of the foreground is higher than that of background. Although the background information can be built in advance, it is likely to be adversely affected by standing vehicles or heavy traffic. The temporal difference method, which relies on the motion and texture of objects to identify the background, fails if the texture of objects is smooth. This method also

malfunctions when objects periodically remain still. The objective of this study is to analyze the characteristic of vehicles in local and global features. This boosted approach may provide a more effective solution to the problems encountered by conventional background-based detection systems in the area of vehicle detection.

An exhaustive search of all positions in the image is a solution to vehicle detection without a background model. However, this solution is not satisfactory for real-time applications. Most of the methods used to solve this problem can be decomposed to the two following steps:

- Hypothesis Generation (HG): This step provides the potential positions of vehicles in a simple and rapid manner, producing a reduced search area.
- Hypothesis Verification (HV): The residual candidate regions in the HG step are verified using complex algorithms to validate the exact positions of vehicles.

The following paragraphs discuss studies that have included these two steps.

Various HG approaches have been proposed in the literature. The objective of the HG step is to find candidate vehicle locations in an image quickly, reducing the computational requirements for further searching. The HG step is typically based on simple, low-level algorithms that show potential vehicle locations. The hypothesized locations produced during the HG step serve as the input for the HV step, where tests are performed to verify the accuracy of the hypotheses. The principle of the HG step is to minimize unqualified search windows while maximizing the overall detection rate.

The rear and frontal views of vehicles are typically symmetrical in the horizontal direction; A. Bensrhair *et al.* [1], A. Kuehnle [2], and T. Zielke *et al.* [3] used symmetry (which is a main feature of artificial objects) to determine the existence of vehicles. S. D. Buluswar *et al.* [4] and D. Guo *et al.* [5] used RGB color information and the $L^* a^* b$ color space, respectively, to separate vehicles from the image background. Bertozzi *et al.* [6] proposed a corner-based method to hypothesize vehicle locations. Matthews *et al.* [7] used edge detection to determine strong vertical edges. By computing the vertical profile of the edge image (*i.e.*, by accumulating the pixels in each column) and smoothing, using a triangular filter, they were able to identify the local maximum peaks of the vertical profile, including the left and right borders of the vehicle. Goerick *et al.* [8] proposed Local Orientation Coding (LOC) to extract edge information. An image obtained using this method consists of strings of binary code representing the directional gray-level variation in the pixel's proximity. These codes, which carry essential edge information, are used as the hypothesis. The presence of vehicles in an image causes local intensity changes. This property can be used as a cue to limit the search area for vehicle detection. U.

Handmann *et al.* [9] proposed that the intensity changes follow a certain texture pattern, and used entropy as a measure for texture detection. They chose a small window around each image pixel, and the entropy of that window was considered as the entropy of the pixel. Only regions with high entropy were considered for further processing. Optical flow can also provide strong information for HG. For example, C. Demonceaux *et al.* [10] and A. Giachetti *et al.* [11] used optical flow to identify the motion of any objects (including preceding vehicles) from the road motion and segment the objects.

The combination of multiple cues should also be explored as a viable means of developing reliable and robust systems. The main motivation behind this approach is that the use of a single cue suitable for all conceivable scenarios may be impossible. The combination of different cues has produced promising results (*e.g.*, combining LOC, entropy, and shadow [9], shape, symmetry, and shadow [12], color and shape [13], and motion with appearance [14]). Effective fusion mechanisms and cues that are fast and easy to compute are critical research issues.

The set of hypothesized locations from the HG step is the input in the HV step. During the HV step, tests are performed to verify the accuracy of a hypothesis. A. Khammari *et al.* [15] classified HV methods into two categories: 1) template-based and 2) appearance-based.

In template-based methods, predefined patterns of the vehicle class are used to determine the correlation between an input image and the template. M. Betke *et al.* [16] proposed a multiple vehicle detection approach by using deformable gray-scale template matching. J. Ferryman *et al.* [17] proposed a deformable model formed from manually sampled data using Principal Component Analysis (PCA). The structure and pose of a vehicle can be recovered by fitting the PCA model to the image.

Appearance-based methods acquire vehicle class characteristics from a set of training images that capture the variability in vehicle appearance. The variability of the non-vehicle class is also modeled to improve performance. First, each training sample is represented by a set of local or global features. The decision boundary between the vehicle and non-vehicle classes is then learned by training a classifier (*e.g.*, Support Vector Machine (SVM) [18], Neural Network (NN) [19,20]) or by modeling the probability distribution of the features in each class (*e.g.*, Bayes rule assuming Gaussian distributions [21]). In [22], wavelet transform was used for feature extraction and SVMs were used for classification.

Most research efforts have focused on feature extraction and classification based on learning and statistical models. Efforts in this direction should continue to capitalize on recent advances in statistical and machine learning. The most crucial task in the object detection

literature is to select a good set of features. In most cases, numerous features are used to compensate for the fact that relevant features are unknown a priori. However, without employing a feature selection strategy, many features are redundant, which could negatively affect classification accuracy and efficiency. It is highly desirable to use only those features that have high separability, and ignore or focus less attention on the remaining features. For example, features that encode fine details (*i.e.*, those that might present only in certain vehicles) should be excluded to achieve suitable generalization in a vehicle detector. Feature selection is the process of determining which feature to use for classification and recognition.

R. Wang *et al.* [23] proposed a vehicle detection system based on the local features located within three significant vehicle subregions. By combining PCA and Independent Component Analysis (ICA), they projected each examined subregion onto its associated eigenspace and independent basis space to generate a PCA weight vector and an ICA coefficient vector, respectively. They then performed a likelihood evaluation process based on the estimated joint probability of the projection weight vectors and the coefficient vectors of the subregions with position information. This use of subregion position information minimizes the risk of false acceptances, whereas the use of PCA to model the low-frequency components of the eigenspace and ICA to model the high-frequency components of the residual space improves the tolerance of the detection process toward variations in the illumination conditions and vehicle pose.

P. Viola and M. J. Jones [24] proposed an original feature selection scheme for object detection. Their approach consists of a cascade of boosted classifiers with increasing complexity: each layer in the cascade reduces the search zone and rejects regions that do not contain objects of interest. This method uses Haar-like features, also called rectangular filters (proposed by Papageorgiou *et al.* [25]) and AdaBoost learning [26], which allows the selection of a limited number of features in each layer. The use of integral images to calculate Haar-like features and the cascade approach produces a real-time face-detection application. This approach has recently inspired considerable research in vehicle detection.

Various researchers have proposed improvements to these features. For example, P. Negri *et al.* [27] combined the rectangular filters (Haar-like features) and the histogram of oriented gradient (HoG) methods with the AdaBoost algorithm. This fusion combines the advantages of two other detectors: generative classifiers composed of Haar-like features “easily” eliminate negative examples in the early layers of the cascade, whereas the discriminative classifiers composed of HoG features generate a fine decision boundary removing the negative examples

near the vehicle model in later layers. Therefore, this fusion achieves a stronger performance than either method alone.

The HG methods in this study are based on symmetry and statistic of edge intensity, and serve as candidate region filters. The cascaded Haar feature classifier refined by the AdaBoost algorithm is used as an initial hypothesis verification method, and plays the role of vehicle detector. Subsequently, a probabilistic variant of a decision-based neural network [19,20] is used as a second HV method and a target verifier, considerably reducing the false positive rate. The nature and decision boundary of the first method differ from those of the second: the first is local-feature oriented, whereas the second is global-feature oriented. Therefore, these two types of classifiers act in a complementary manner when dealing with this problem.

2. Vehicle Detection System

The proposed system structure consists of five sub-systems: pre-processing, candidate region filtering, vehicle detection, target validation, and post-processing. First, the raw data of a surveillance video serves as the input to the pre-processing module. This sub-system transforms the raw color image into a gray-level image. To accelerate performance, the gray-level image is downsized using a simple interpolation algorithm. Sobel edge operation is then applied to the raw-size gray-level image to preserve the detailed edge intensity of the image, and the edge intensity image is then resized using the same interpolation algorithm to generate a downsized edge intensity image. This smaller image serves as the input in the next sub-system: candidate region (CR) filtering. **Figure 1** shows the eliminated search scope of CR filtering. After filtering out unqualified regions by using CR filters, the residual positions of the downsized gray-level image that might contain objects of interest are examined and located using the AdaBoost classifier. These positions are then verified using the Probabilistic Decision-Based Neural Network (PDBNN) classifier based on probability estimation to confirm the existence of interested objects. Finally, the post-processing step addresses the redundant overlapping windows that belong to the same object, showing the exact locations of the objects. **Figure 2**

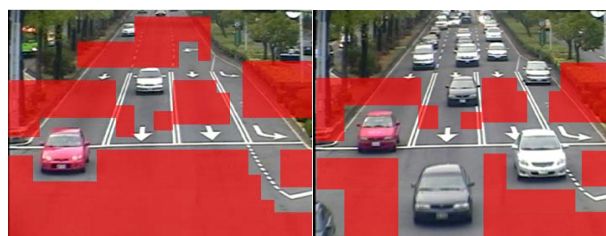


Figure 1. The eliminated search scope of CR filtering.

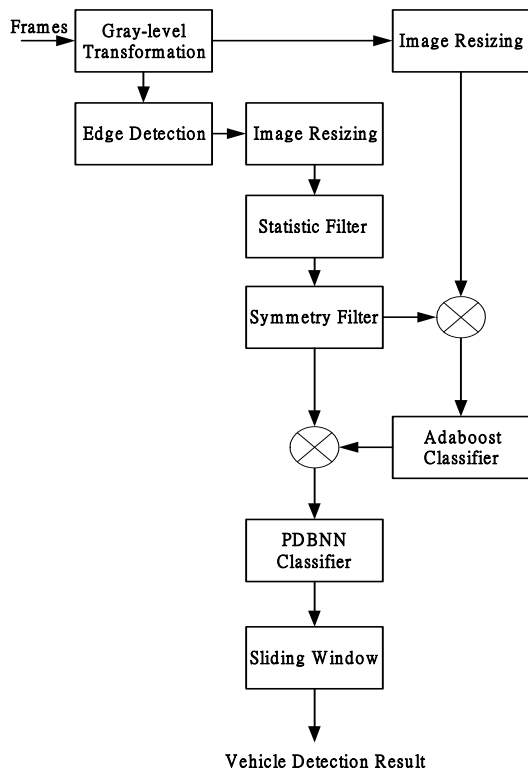


Figure 2. System diagram.

shows a diagram of the global system. The following sections present the main models of the proposed system.

2.1. Candidate Region Filtering

In this study, CR filters serve as the alias of HG methods. Based on simple, fast, and low-level algorithms, the purpose of CR filtering is to eliminate regions that contain no objects. Therefore, the potential locations of vehicles are hypothesized, and the follow-up HV steps (*i.e.*, vehicle detection and target validation in this study) can focus on the regions that are likely to contain objects of interest. This step considerably reduces computation time, and can simultaneously lower the false positive rate. The purpose of candidate region filtering is to minimize unqualified searching windows while maximizing the overall detection rate.

2.2. Statistic Filter

The rational edge intensity distributions of objects of interest can be estimated using a simple statistical method. Therefore, the candidate regions whose texture is too sparse or too dense can be filtered out rapidly. **Figure 3** shows the distribution of the edge intensity statistic and the threshold for the scope.

The edge intensity distribution of positive samples (*i.e.*, vehicles) is approximately a Gaussian distribution. Therefore, the lower and upper thresholds can be ob-

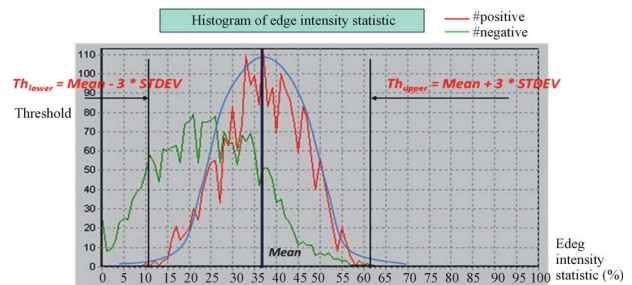


Figure 3. The scope of edge intensity statistic.

tained by mean and standard deviation of this Gaussian distribution with Equations (1) and (2).

$$\text{Threshold}_{\text{lower}} = \text{Mean} - 3 * \text{STDEV} \quad (1)$$

$$\text{Threshold}_{\text{upper}} = \text{Mean} + 3 * \text{STDEV} \quad (2)$$

where Mean and STDEV represent the mean and standard deviation of the Gaussian distribution, respectively. It has the desirable property that a known percentage of all possible values of data lie within a certain number of standard deviations from the mean. Theoretically, 99.7% of the rates will fall within plus or minus 3 standard deviations of the mean.

2.3. Symmetry Filter

The lower parts of vehicles are visually symmetrical and textured. Therefore, the symmetry of the candidate region can be used to hypothesize whether the candidate region contains a vehicle. By accumulating the histogram of intensity (HoI) vertically and horizontally, the rough degree of symmetry can be obtained using Equation (3):

$$\text{Degree}(A = B) = \frac{A \cap B}{A \cup B} \quad (3)$$

where A is the measure of the left-half HoI, and B is the measure of the right-half HoI. The threshold of the minimum symmetry degree which a vehicle should hold can be obtained using a statistical method. **Figure 4** shows that any region of interest with a symmetry degree below the threshold is removed.

2.4. Vehicle Detection Using AdaBoost

This sub-system is the main component of the entire system. Most research efforts have focused on feature extraction and classification by learning and statistical models. Therefore, the most critical issue in object detection literature is to select a suitable set of features that can represent the implicit invariant of objects of interest. An intuitive method of feature extraction is to focus on the common components of objects of interest. Any perceptual characteristic (e.g., color, edge, texture, entropy, symmetry) can be used as a feature solely or coopera-

tively. Although this method is easy to imagine and implement, it has limitations. One of these limitations is that manually chosen features are inferior in both value and quantity. These manually chosen features tend to be apparent, and the physical nature of human's perception is typically insufficiently consistent to substitute for these objects. Therefore, efforts in feature extraction have focused on statistical and machine learning areas.

The famous AdaBoost algorithm proposed by Y. Freund *et al.* [26] is a machine learning technique that has been widely used in pattern recognition. AdaBoost combines weak classifiers in a weighted voting machine, and achieves strong performance in various fields. **Figure 5** shows the schema of an AdaBoost classifier.

An AdaBoost classifier consists of several strong classifiers, and a strong classifier is actually a linear combination of several weak classifiers, whose natures are Haar features (rectangle features).

The Haar basis functions (Haar features) used by Pageorgiou *et al.* [28] provide information on the grey-level distribution of two or more adjacent regions in an image. **Figure 6** shows a set of simple Haar features. These features consist of two to four rectangles. To compute the output of a Haar basis function for a certain region of image, the sum of all pixel intensities in the black region is subtracted from the sum of all pixel intensities in the white region and normalized using a coefficient. This serves as a precaution against a filter whose square



Figure 6. Haar features.

measures of white and black regions are different. To reduce computation time for the filters, P. Viola *et al.* [24] introduced the integral image, which is an intermediate representation for an input image. By using integral images, the sum of a rectangular region can be calculated using only four references in the integral image. Consequently, the difference between two adjacent rectangular regions can be computed. This computation requires only six references in the integral image, eight in the case of the three-rectangle filters, and nine for four-rectangle filters.

A weak classifier h consists of a rectangle feature f , a threshold (θ) and a polarity (p) indicating the direction of the inequality in Equation (4):

$$h = \begin{cases} 1 & \text{if } p_j f_j < p_j \theta_j \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

where f_j is the absolute value (value subtracted) of the feature j , θ_j is the threshold, and p_j is the parity. For each feature j , the AdaBoost algorithm is used to determine an optimal threshold θ_j that minimizes the classification error in the training database (with positive and negative samples). The positive samples and negative samples can be separated with a lowest classification error by selecting the appropriate threshold.

The AdaBoost algorithm performs effectively in various fields. This algorithm combines weak classifiers with a strong classifier using a weighted voting mechanism. Viola and Jones [24] proposed a novel feature selection algorithm for face detection. Their algorithm uses training data to construct a cascade-booster classifier. The proposed method also uses Haar-like features to generate weak classifiers and determine the precise hypotheses of a vehicle by iteratively combining weak classifiers, which typically exhibit moderate precision, into a strong classifier, defined by

$$C(x) = \begin{cases} 1 & \sum_{t=1}^T \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^T \alpha_t \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

where h and C are weak and strong classifiers, respectively, and α is the weight coefficient for each h . The AdaBoost algorithm can be described as follows:

Step 1: Given example images $(x_1, y_1), \dots, (x_n, y_n)$, where $y_i = 0, 1$ for negative and positive examples, respectively.

Step 2: Initialize weights $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ for $y_i = 0, 1$,

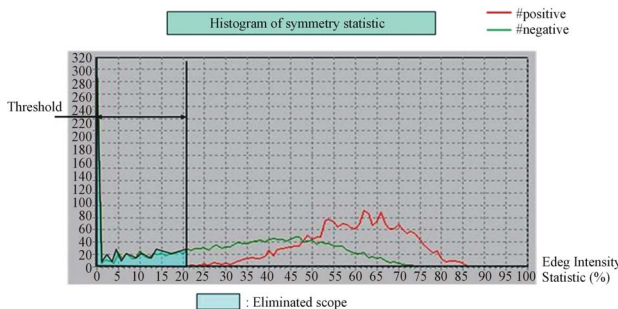


Figure 4. The statistic scope of symmetry.

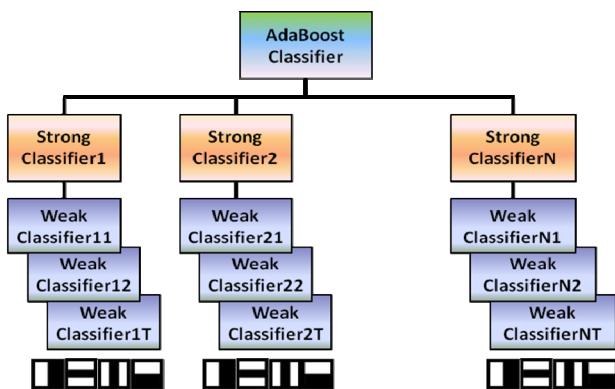


Figure 5. The schema of an AdaBoost classifier.

respectively, where m and l are the number of negatives and positives, respectively.

Setp 3: For $t = 1, \dots, T$

1) Normalize the weights

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

2) Select the best weak classifier with respect to the weighted error

$$\varepsilon_t = \min_{f,p,\theta} \sum_i w_i |h(x_i, f, p, \theta) - y_i|$$

3) Define $h_t = h(x, f_t, p_t, \theta_t)$, where f_t, p_t and θ_t are the minimizers of ε_t .

4) Update the weights:

$$w_{t+1,i} = w_{t,i} \beta_t^{1-\theta_i}$$

where $e_i = 0$ if example x_i is classified correctly,

$e_i = 1$ otherwise, and $\beta_t = \frac{\varepsilon_t}{1 - \varepsilon_t}$.

Step 4: The final strong classifier is:

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

where $\alpha_t = \log 1/\beta_t$

2.5. Target Validation Using a PDBNN

A PDBNN assumes that the class likelihood function for the object class (*i.e.*, vehicle class) can be represented by a mixture of Gaussian distributions. The training scheme for these Gaussian distributions includes two phases: 1) a locally unsupervised (LU) phase and 2) a globally supervised (GS) phase.

The values of the weight vector parameters in the network are initialized in the LU learning phase. An unsupervised clustering algorithm (k-means) is used to segment the training samples of the object class into several clusters and determine the initial positions of the cluster centroids. **Figure 7(a)** shows that the expectation-maximization (EM) algorithm [29] is then used to learn the cluster parameters of the object class, obtain maximum likelihood estimation (MLE) of the data distribution, and determine the parameters of Gaussian distributions. **Figure 7(b)** shows that, in the global supervised (GS) training phase, two sets of patterns (*i.e.*, both positive and negative samples) are involved. Teacher information is used to fine-tune the decision boundary, which is determined by a threshold. When a training pattern is misclassified, the reinforced or anti-reinforced learning technique [19] is applied. If the misclassified

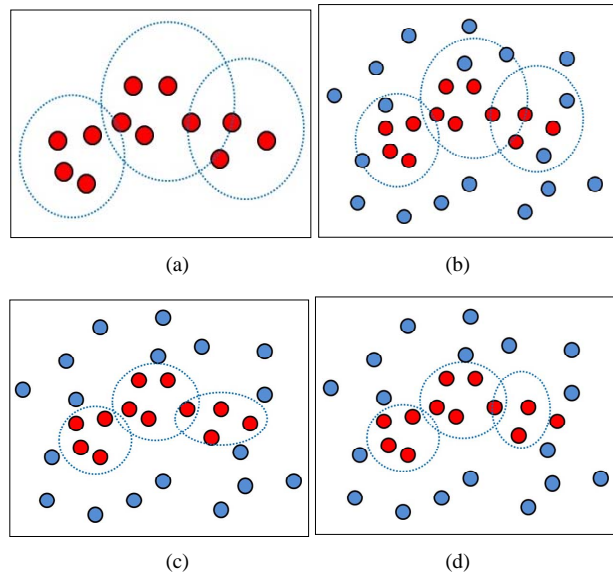


Figure 7. The diagram of PDBNN learning.

training pattern is from the positive training set (*i.e.*, the vehicle training data set in the LU phase), reinforced learning is used to adjust the subnet parameters and the boundary by updating the weight vector in the direction of the gradient of the discriminant function. **Figures 7(c)** and **(d)** show that if the training pattern belongs to the so-called negative training (*i.e.* non-vehicle) set, the anti-reinforced learning rule is executed to adjust only the boundary in the opposite direction of the gradient of the discriminant function.

2.6. Sliding Window Detector

The Haar-like feature is computed rapidly for many types, scales, and positions. The number of Haar-like features within any image sub-window is large. Although each Haar-like feature can be computed efficiently using an integral image algorithm, computing the complete set remains extremely time consuming and costly. This study presents a position-based sliding window detector to solve this problem. This notion is based on the relationship between vehicle positions to the image coordinate. **Figure 8(a)** shows the image captured using a camera. The vehicle size in the top window has a smaller size than that in the bottom image. By using this characteristic, several scanning windows are defined (**Figure 8(b)**) with the following scale factor:

$$\text{Scale Factor} = \frac{w_1 - w_0}{y_1 - y_0} \tag{6}$$

where w_0 and w_1 denote the largest and smallest vehicle size, respectively. y_0 and y_1 are their corresponding positions in image coordinate. The sliding window size at t -position is defined in the following equation:

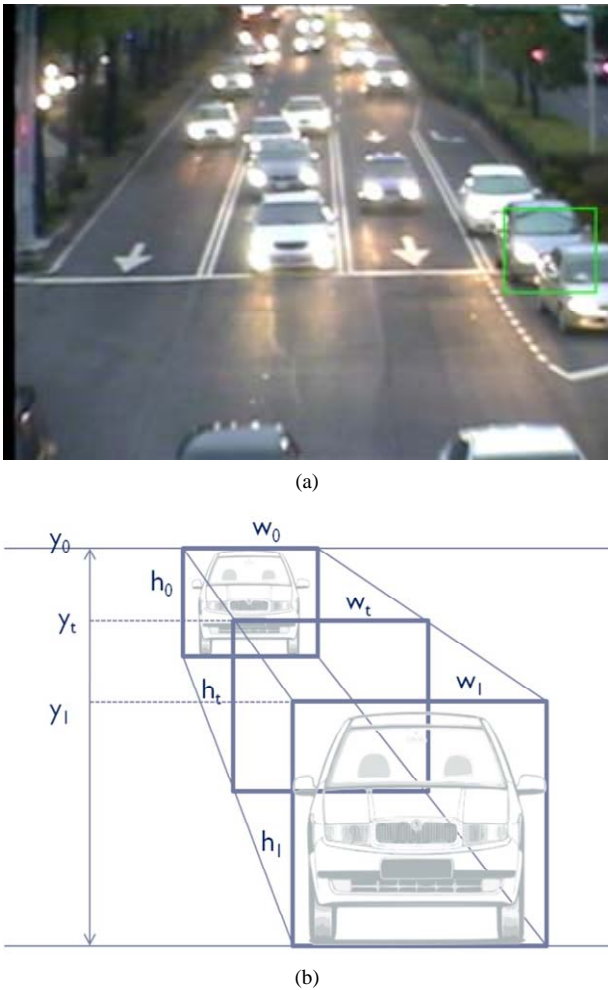


Figure 8. Sliding window detector: (a) original image and (b) size of window detector at different position of image coordinate.

$$w_t = w_0 + (y_t - y_0) \times \text{Scale Factor} \quad (7)$$

By using w_t , the unreasonable size of vehicle detector, whose width is excessively large or small for a particular position, can be filtered out by the tolerance range for acceptable width and final filtering rule as follows:

$$w_t - \varepsilon < \text{AdaBoost vehicle detector result} < w_t + \varepsilon \quad (8)$$

where ε is the empirical threshold for efficient falsification.

3. Experimental Results

The proposed vehicle-detection system, which is based on machine learning algorithms, consists of training processes and testing processes. Sample collection and feature extraction are key steps in object detection and classification. To build the proposed system, the common components of vehicles must be determined first (e.g., windshields, lamps, roofs, and wheels). The basis of col-

lecting samples is to retain the common features and remove unnecessary features to reduce the interference of noise. Consequently, the images are tightly cropped. Second, manually collected samples are typically not uniform in size, point of view, and illumination. Therefore, a normalization process is necessary, and an identical criterion is set before the samples enter the training or testing step. The proposed method investigates the specific application of a vehicle-counting system at an intersection.

The proposed system captures frontal views of small vehicles such as sedans, sport-utility vehicles (SUVs), and minivans, which comprise the majority of traffic. The images collected from a proprietary database were selected from videos captured using a hand-held camera in daytime or early evening. The captured sources contain several scenes, most of which were captured at an intersection. Therefore, stop-and-go traffic was the general case investigated in this study. While collecting vehicle samples, the pose of captured cars was limited to frontal-views and the angle tolerance of pan-rotation range was approximately $\pm 20^\circ$. **Figure 9** shows the collected vehicle and non-vehicle samples.

$$\text{Detection Rate} = \frac{\text{Number of detected vehicles}}{\text{Total number of vehicles}} \quad (9)$$

$$\text{False Rate} = \frac{\text{Number of false alarms}}{\text{Total number of nonvehicles windows}} \quad (10)$$



Figure 9. Proprietary samples: (a) Positive samples; (b) Negative samples.

Training data was collected from images captured on urban roads in Taiwan. The Boosted vehicle detector database consisted of 2344 positive and 30,694 negative samples. These samples were normalized to 21×18 pixels. The proposed system (AdaBoost + PDBNN) was tested using the MIT CBCL database, and its performance was compared with that of the PCA + ICA and AdaBoost and PDBNN methods. **Table 1** shows the detailed characteristics of each road. Each image in the MIT CBCL car database was extracted from raw data and normalized to a size of 128×128 pixels. The position of each car was aligned so that it was in the center of the image. **Figure 10** and **Table 2** show the result. The proposed approach showed the lowest false alarm rate

Table 1. Characteristic of urban road in Taiwan.

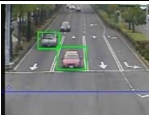

Traffic Surveillance	Traffic Surveillance Characteristic	Place	Vehicle Speed
	More likely intersection which vehicle gets to use change the directoin to other path of road.	Taichung	Normal
	The vehicles stay for a long time due to the traffic lights. Heavy traffic including motorbike	Hsinchu	Slow

Table 2. Comparison using MIT CBCL database [30].

Method	Detection Vehicle	False Alarm
PCA + ICA [23]	95%	0.002%
AdaBoost	97.5%	0.0018%
PDBNN	96.6%	0.0078%
Proposed System	96.3%	0.0013%

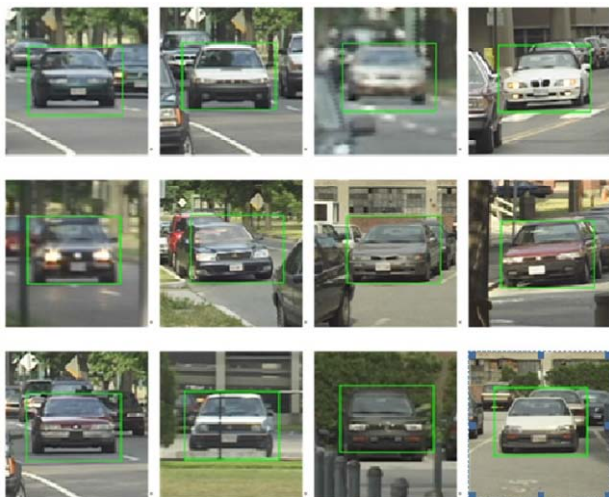


Figure 10. Some detection results on MIT CBCL car database.

among all the compared methods. Equations (9) and (10) show the MIT CBCL database criteria of performance measurement.

To estimate the performance of the overall system in a more quantitative manner, the performance of the system was also tested by estimating the number of detected vehicles. The compared methods have GMM. The testing set consisted of Hsinchu-Guang Fu and Taichung-Da Dun roads. **Figure 11** shows the result for each method for Taichung-Da Dun roads. The left column shows the vehicle detector based on the GMM method and the right column shows the results of the proposed method.

The method performs well, but GMM produced more false alarms than the proposed system. **Table 3** shows the detailed performance of each method. **Figure 12** and **Table 4** show the performance of Hsinchu-Guang Fu road. Because the scenes we used included heavy traffic and contained both vehicles and motorcycles, it was difficult for the GMM method to segment a vehicle and to distinguish a motorcycle from other types of vehicles.

4. Discussion and Conclusion

This study presents a boosted vehicle detector that in-

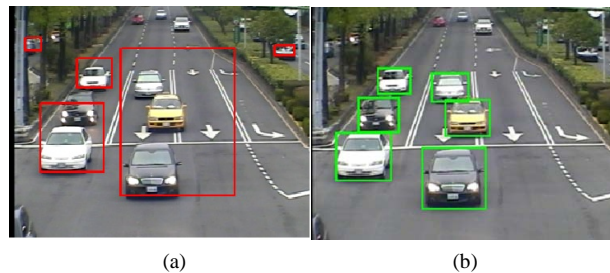


Figure 11. Taichung-Da Dun road result: (a) GMM method and (b) Proposed Method.

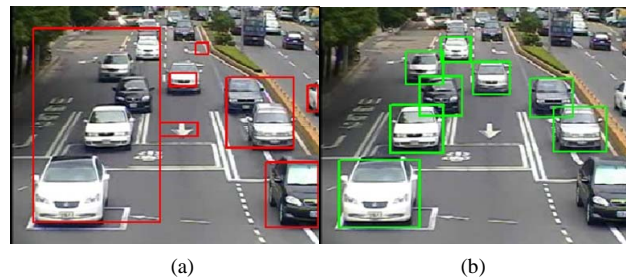


Figure 12. Hsinchu-Guang Fu road result: (a) GMM Method and (b) Proposed Method.

Table 3. The comparison of Taichung-Da Dun road result.

Method	Detection Vehicle	False Alarm	Total Vehicle
GMM	110 (82.09%)	7 (5.98%)	134
Proposed systme	131 (97.76%)	2 (1.5%)	

Table 4. The comparison of Hsinchu-Guang Fu road result.

Method	Detection Vehicle	False Alarm	Total Vehicle
GMM	103 (78.62%)	110 (83.96%)	131
Proposed system	125 (95.41%)	8 (6.1%)	

volves the use of local and global features. The HG methods in this study provide a more effective method to solve this problem; using local or global features alone is insufficient to extract the implicit invariance of vehicles. The AdaBoost algorithm shows considerable potential as a capable classifier for dense training data, and provides a robust generalization ability.

By combining the local features and global features of vehicles, the proposed system can be used to achieve a higher detection rate than other systems, while successfully suppressing the false positive rate. The proposed system works effectively for both light and heavy traffic scenes without relying on background information, and can be applied to static images or video frames.

The proposed vehicle detection system can also operate effectively in the varying conditions that occur in real environments.

5. Acknowledgements

The study was supported by the "Aim for the Top University Plan" of National Chiao Tung University and the Ministry of Education, Taiwan.

REFERENCES

- [1] A. Bensrhair, M. Bertozzi, A. Broggi, P. Miche, S. Mousset and G. Toulminet, "A Cooperative Approach to Vision-Based Vehicle Detection," *Proceedings of the 4th IEEE Conference on Intelligent Transportation Systems (ITSC'01)*, Oakland, August 2001, pp. 207-212.
- [2] A. Kuehnl, "Symmetry-Based Recognition of Vehicle Rears," *Pattern Recognition Letters*, Vol. 12, No. 4, 1991, pp. 249-258. [doi:10.1016/0167-8655\(91\)90039-0](https://doi.org/10.1016/0167-8655(91)90039-0)
- [3] T. Zielke, M. Brauckmann and W. V. Seelen, "Intensity and Edge-Based Symmetry Detection with an Application to Car-Following," *CVGIP: Image Understanding*, Vol. 58, No. 2, 1993, pp. 177-190. [doi:10.1006/ciun.1993.1037](https://doi.org/10.1006/ciun.1993.1037)
- [4] S. D. Buluswar and B. A. Draper, "Color Machine Vision for Autonomous Vehicles," *International Journal of Engineering Applications of Artificial Intelligence*, Vol. 1, No. 2, 1998, pp. 245-256. [doi:10.1016/S0952-1976\(97\)00079-1](https://doi.org/10.1016/S0952-1976(97)00079-1)
- [5] D. Guo, T. Fraichard, M. Xie and C. Laugier, "Color Modeling by Spherical Influence Field in Sensing Driving Environment," *IEEE Intelligent Vehicles Symposium*, Dearborn, 3-5 October 2000, pp. 249-254.
- [6] M. Bertozzi, A. Broggi and S. Castelluccio, "A Real-Time Oriented System for Vehicle Detection," *Journal of Systems Architecture*, Vol. 43, No. 1-5, 1997, pp. 317-325.
- [7] N. Matthews, P. An, D. Charnley and C. Harris, "Vehicle Detection and Recognition in Greyscale Imagery," *Control Engineering Practice*, Vol. 4, 1996, pp. 473-479. [doi:10.1016/0967-0661\(96\)00028-7](https://doi.org/10.1016/0967-0661(96)00028-7)
- [8] C. Goerick, N. Detlev and M. Werner, "Artificial Neural Networks in Real-Time Car Detection and Tracking Applications," *Pattern Recognition Letters*, Vol. 17, 1996, pp. 335-343. [doi:10.1016/0167-8655\(95\)00129-8](https://doi.org/10.1016/0167-8655(95)00129-8)
- [9] U. Handmann, T. Kalinke, C. Tzomakas, M. Werner and W. V. Seelen, "An Image Processing System for Driver Assistance," *Image and Vision Computing*, Vol. 18, No. 5, 2000, pp. 367-376. [doi:10.1016/S0262-8856\(99\)00032-3](https://doi.org/10.1016/S0262-8856(99)00032-3)
- [10] C. Démonceaux, A. Potelle and D. Kachi-Akkouche, "Obstacle Detection in a Road Scene Based on Motion Analysis," *IEEE Transactions on Vehicular Technology*, Vol. 53, No. 6, 2004, pp. 1649-1656. [doi:10.1109/TVT.2004.834881](https://doi.org/10.1109/TVT.2004.834881)
- [11] A. Giachetti, M. Campani and V. Torre, "The Use of Optical Flow for Road Navigation," *IEEE Transactions on Robotics and Automation*, Vol. 14, No. 1, 1998, pp. 34-48. [doi:10.1109/70.660838](https://doi.org/10.1109/70.660838)
- [12] J. Collado, C. Hilario, A. de la Escalera and J. Armingol, "Model Based Vehicle Detection for Intelligent Vehicles," *IEEE Intelligent Vehicles Symposium*, Parma, 14-17 June 2004, pp. 572-577.
- [13] K. She, G. Bebis, H. Gu and R. Miller, "Vehicle Tracking Using On-Line Fusion of Color and Shape Features," *The 7th International IEEE Conference on Intelligent Transportation Systems*, Washington, 3-6 October 2004, pp. 731-736.
- [14] J. Wang, G. Bebis and R. Miller, "Overtaking Vehicle Detection Using Dynamic and Quasi-Static Background Modeling," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Diego, 25-25 June 2005, p. 64.
- [15] A. Khammari, E. Lacroix, F. Nashashibi and C. Laureau, "Vehicle Detection Combining Gradient Analysis and AdaBoost Classification," *IEEE Conferences on Intelligent Transportation Systems*, Vienna, 13-15 September 2005, pp. 66-71.
- [16] M. Betke, E. Haritaglu and L. Davis, "Multiple Vehicle Detection and Tracking in Hard Real Time," *IEEE Intelligent Vehicles Symposium*, Tokyo, 19-20 September 1996, pp. 351-356.
- [17] J. Ferryman, A. Worrall, G. Sullivan and K. Baker, "A Generic Deformable Model for Vehicle Recognition," *Proceedings of British Machine Vision Conference*, University of Birmingham, Birmingham, 1995, pp. 127-136.
- [18] Z. Sun, G. Bebis and R. Miller, "On-Road Vehicle Detection Using Gabor Filters and Support Vector Machines," *14th International Conference on Digital Signal*, Vol. 2, 2002, pp. 1019-1022.
- [19] S. Y. Kung and J. S. Taur, "Decision-Based Neural Networks with Signal/Image Classification Applications," *IEEE Transactions on Neural Networks*, Vol. 6, No. 1,

- 1995, pp. 170-181. [doi:10.1109/72.363439](https://doi.org/10.1109/72.363439)
- [20] S.-H. Lin, S.-Y. Kung and L.-J. Lin, "Face Recognition/Detection by Probabilistic Decision-Based Neural Network," *IEEE Transactions on Neural Networks*, Vol. 8, No. 1, 1997, pp. 114-132.
- [21] D. W. Ruck, S. K. Rogers, M. Kabrisky, M. E. Oxley and B. W. Suter, "The Multilayer Perceptron as an Approximation to a Bayes Optimal Discriminant Function," *IEEE Transactions Neural Networks*, Vol. 1, No. 4, 1990, pp. 296-298. [doi:10.1109/72.80266](https://doi.org/10.1109/72.80266)
- [22] Z. Sun, R. Miller, G. Bebis and D. Dimeo, "A Real-Time Pre-crash Vehicle Detection System," *6th IEEE Workshop on IEEE Intelligent Vehicles Symposium*, Dearborn, 2000, pp. 171-176.
- [23] C.-C. R. Wang and J.-J. J. Lien, "Automatic Vehicle Detection Using Local Features—A Statistical Approach," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 9, No. 1, 2008, pp. 83-96. [doi:10.1109/TITS.2007.908572](https://doi.org/10.1109/TITS.2007.908572)
- [24] P. Viola and M. J. Jones, "Robust Real-Time Face Detection," *International Journal of Computer Vision*, Vol. 57, No. 2, 2004, pp.137-154. [doi:10.1023/B:VISI.0000013087.49260.fb](https://doi.org/10.1023/B:VISI.0000013087.49260.fb)
- [25] C. Papageorgiou and T. Poggio, "A Trainable System for Object Detection," *International Journal of Computer Vision*, Vol. 38, No. 1, 2000, pp. 15-33. [doi:10.1023/A:1008162616689](https://doi.org/10.1023/A:1008162616689)
- [26] Y. Freund and R. E. Schapire, "Experiments with a New Boosting Algorithm," *Proceedings of the 13th International Conference on Machine Learning (ICML'96)*, Bari, July 1996, pp. 148-156.
- [27] P. Negri, X. Clady, S. M. Hanif and L. Prevost, "A Cascade of Boosted Generative and Discriminative Classifiers for Vehicle Detection," *EURASIP Journal on Advances in Signal Processing*, Vol. 2008, 2008, Article ID: 782432. [doi:10.1155/2008/782432](https://doi.org/10.1155/2008/782432)
- [28] C. Papageorgiou, M. Oren and T. Poggio, "A General Framework for Object Detection," *6th International Conference on Computer Vision*, Bombay, 4-7 January 1998, pp. 555-562.
- [29] A. P. Dempster, N. M. Laird and D. B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *Journal of the Royal Statistical Society. Series B*, Vol. 39, Mo. 1, 1976, pp. 1-38.
- [30] "MIT CBCL Center for Biological and Computational Learning Car Data," 2000. <http://cbcl.mit.edu/software-datasets/CarData.html>