Signal and Image Processing Letters
Vol. 3., No. 1, 2021, pp. 13-20
ISSN 2714-6677
https://simple.ascee.org/

Horizontal Lines and Haar-like Features for Car Detection Using Support Vector Machine on Traffic Imagery

Aldi Khoirul Abdillah [a,1], Adhi Prahara [a,2,*]

[a] Department of Informatics, Universitas Ahmad Dahlan, Yogyakarta, Indonesia
[1] aldi1400018255@webmail.uad.ac.id; [2] adhi.prahara@tif.uad.ac.id
* corresponding author

ARTICLE INFO

Keywords
Car detection
Haar-like features
Horizontal line
Support Vector Machine
Computer vision

ABSTRACT

Traffic monitoring system in Indonesia is not yet efficient. CCTV cameras had been installed to monitor the traffic in strategic locations. However, it is difficult to monitor each traffic point all the time. This problem leads to the development of intelligent traffic monitoring system using computer vision technology. In this research, a car detection method is proposed. Car detection still poses challenges especially when dealing with various situations on the road. The proposed car detection method uses horizontal lines and Haar-like features trained with Support Vector Machine (SVM) to detect cars on traffic imagery. The car detector is trained on frontal-view car dataset. The test result shows 0.2 log average miss rate and 0.9 average precision. From the low miss rate and high precision, the proposed method shows promising solution in detecting cars on traffic imagery.

1. Introduction

Traffic monitoring is an important task, especially in large cities that have large volume of vehicles passing every day. In Indonesia, traffic monitoring is carried out at strategic points via CCTV cameras. Through CCTV cameras, operator can monitor congestion, traffic violations, and count the number of vehicles passing on the road. The monitoring results are used to control and regulate traffic. However, this method is not efficient because the operator cannot monitor the traffic all the time. The need of automatic traffic monitoring has led to the development of an intelligent traffic monitoring system using computer vision technology [1], [2]. The main purpose of computer vision is for computers or machines to imitate the perceptual abilities of the human eye and brain, or even able to surpass them for certain purposes [3]. Computer vision technique receives image or video as an input to be processed according to certain tasks e.g. in this case the task is detecting cars on traffic imagery.

Car detection is an important and challenging stage in intelligent traffic monitoring system due to the various cars and road conditions [2], [4], [5]. Many researchers have been developing methods to handle this problem. Prahara *et. al.* proposed a method to detect car using Histogram of Oriented Gradients (HOG) and Support Vector Machine (SVM) by estimating the road direction [4], [5]. The idea uses four categories of road direction to determine the pose of a car and choose the right detector. By doing that, the proposed method can handle different pose of cars from various viewpoint of CTTV camera. Bougharriou *et al.* also proposed a car detection method using HOG and linear SVM and achieves robustness and good precision from various scene [6]. Wen *et al.* proposed vehicle detection using Haar-like features and perform rapid and effective feature selection via AdaBoost [7]. An improved normalization for the selected features is used to reduce the intra-class difference and increase the inter-class variability. The result shows a speed up in the feature selection process and better detection performance than the state-of-the-art methods. Haselhoff and Kummert proposed a vehicle detection method using Haar-like features and triangle features which computed based on four integral images [8]. The proposed method is based on boosted cascaded classifiers, Haar-like features, triangle features, adaptive sliding window and kalman filter to locate and track the vehicles.

This research proposes car detection method using horizontal line detection and Haar-like features [9] trained using Support Vector Machine (SVM). The method receives image as an input captured from CCTV camera and perform preprocessing on the image before feature extraction. Haar-like features extracted from the integral image then classified using SVM to generate a car detector. The detector coupled with horizontal line detection are used to detect front-viewed car from the image. The rest of this paper is organized as follows. Section 2 presents the proposed car detection method, section 3 presents the result and discussion, and section 4 describes the conclusion of this work.

2. Car Detection Method

The general procedure of the proposed car detection method is shown in Fig. 1. Based on Fig. 1, the procedure is divided into two steps namely training step and test step. In the training step, the input data consists of a collection of frontal-view car images cropped from various traffic images as the positive data and other objects on the road such as motorcycle, traffic sign, building, trees, etc. as the negative data. Preprocessing is done to speed up the feature extraction process with Haar-like features. The features are trained using SVM to generate car detector. In the test step, traffic image is processed as in the training step. Horizontal line features which extracted using Sobel are used to localized the potential area of cars in the image then a car detector is applied to detect cars in that area.

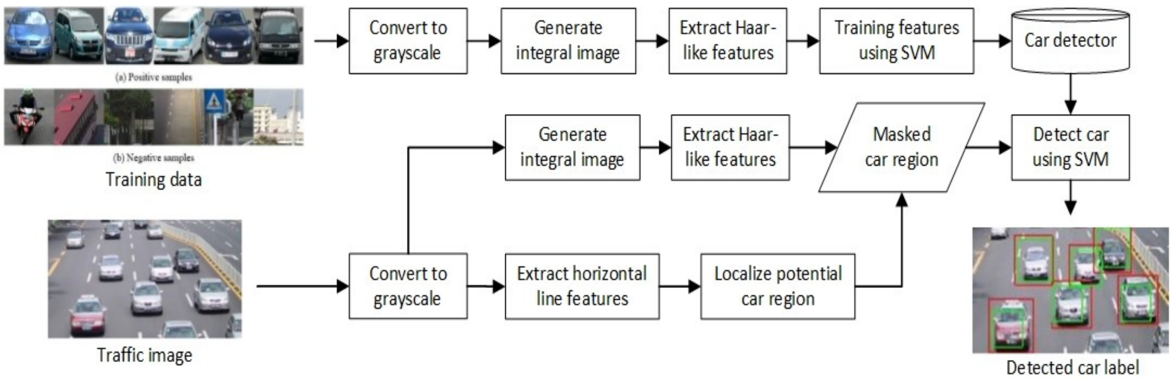
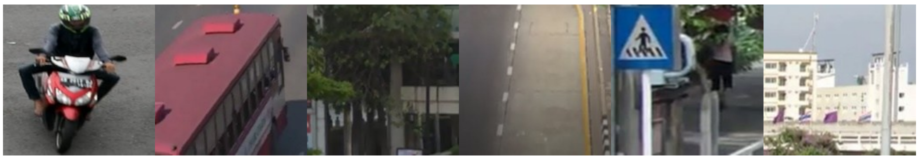


Fig. 1. General procedure of the proposed car detection method

2.1. Dataset

The training data consists of car images as the positive data and other objects as the negative data. The dataset is taken from the internet and cropped to 300x300 pixels. The examples of training data used in this research are shown in Fig. 2a and Fig. 2b. Based on Fig. 2a, the positive image samples consist of various types of vehicle models such as sedans, wagons, SUVs, MPVs, hatchbacks, coupes and pickups which are viewed from the front. Based on Fig. 2b, samples of negative images consist of motorbikes, buses, trucks, people, trees, roads, road signs, buildings and the sky. The test data is traffic images captured from the CCTV video.

(a) Positive samples



(b) Negative samples

Fig. 2. Training data (a) Positive samples (b) Negative samples

2.2. Grayscale Conversion

Grayscale images usually have a pixel range of 0-255 where the color gradation starts from 0 (black) to 255 (white). The number of colors depends on the number of bits available in memory to accommodate the color requirements. The greater the number of color bits provided in memory, the better the color gradation that is formed. The formula to convert RGB color space to grayscale is shown in (1) [10].

$$0.2989 \times R + 0.5870 \times G + 0.1140 \times B \tag{1}$$

2.3. Edge Detection

Edge detection is an image processing technique to find boundaries between regions in an image. The edge detection algorithm is used to separate objects from the background. In this research, Sobel edge detection will be used. The Sobel method [11] is a development of the Robert method by using an HPF (high pass filter) which is given a zero buffer. The advantage of this Sobel method is the ability to reduce noise before performing edge detection calculations.

This study utilizes a gradient on the horizontal axis obtained from the Sobel edge detection method in the horizontal direction. This process aims to determine the number of horizontal edges of the car image. The steps to perform edge detection using the Sobel method are described as follows.

1. Define the configuration of pixel (x, y) (see (2)).

$$\begin{vmatrix} a_0 & a_1 & a_2 \\ a_7 & (x,y) & a_3 \\ a_6 & a_5 & a_4 \end{vmatrix} \tag{2}$$

2. Sobel is computed from the gradient magnitude using (3).

$$M = \sqrt{sx^2 + sy^2} \tag{3}$$

3. Compute the partial derivative Sx and Sy using (4) and (5) respectively.

$$Sx = (a_2 + ca_3 + a_4) - (a_0 + ca_7 + a_6) \tag{4}$$

$$Sy = (a_0 + ca_1 + a_2) - (a_6 + ca_5 + a_4) \tag{5}$$

4. Using a constant c = 2, Sx and Sy kernel are shown in (6).

$$Sx = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \text{ and } Sy = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \tag{6}$$

   Example of Sobel edge detection is shown in Fig. 3.

5. The horizontal line features then extracted from the edge by using Hough transform and a threshold to filter the length of the detected horizontal lines.

6. Count the number of horizontal lines and localize the potential car region based on the lines width and height.

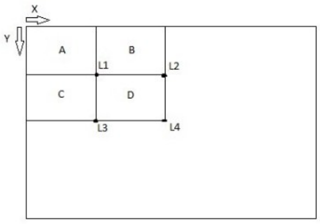
Fig. 3. Result of Sobel edge detection

2.4. Integral Image

Integral image [12] is a medium used to calculate the Haar-like features by converting the input image into an integral image representation. The integral image is used to calculate the sum of all pixels in a rectangle using only four values efficiently. These values are pixels in the integral image that coincide with the corners of the rectangle in the input image. To find out the pixel value of some other rectangles, such as the D rectangle in the image, the computation can be done by combining the number of pixels in the A + B + C + D rectangular area, plus the number of pixels in rectangle A, minus the number of pixels in rectangles A + B and A + C. The value of integral image at location L1 is the sum of the pixels in rectangle A, the value at location L2 is A + B, the value at location L3 is A + C and the value at location L4 is A + B + C + D. So that the result of quadrilateral D can be calculated using (7).

$$D = L4 + L1 - (L2 + L3) \tag{7}$$

An example of integral image computation is shown in Fig. 5.



1	2	3
4	5	6
7	8	9

1	3	6
5	12	21
12	27	45

Fig. 5. Example of integral image computation

2.5. Haar-like Features

Haar-like feature is a feature extraction method that was first introduced by Paul Viola and Michael Jones [9], [12]. Since then, Haar-like feature has been used in many application such as vehicle detection [7], [8], [13]–[16], face detection [9], [17], and pedestrian detection [18]. Haar-like features are rectangular features, which can provide a specific indication of an image. Haar-like feature is used to recognize objects based on the simple value of a feature, not the pixel value contained in the object's image. Each Haar-like feature consists of a combination of black and white rectangle. There are three types of rectangular features such as shown in Fig. 6a, 6b and 6c.

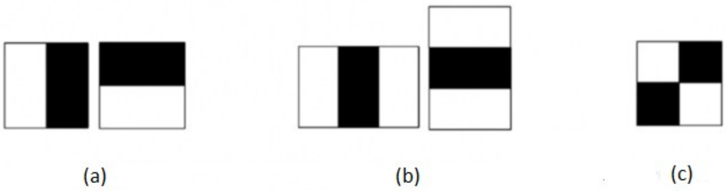


Fig. 6. The shape of rectangular features of Haar-like features (a) Edge features (b) Line features (c) Four rectangle features

2.6. Support Vector Machine

The Support Vector Machine (SVM) [19] comes from two classes classification problem that requires positive and negative training sets. SVM also has been widely used in object detection [4]–[7], [20]–[23] and achieves good result. SVM tries to find the best hyperplane (separator) to separate two classes and maximize the margin between the two classes. In some cases, the data cannot be classified using linear SVM method, so a kernel function was developed to classify data in non-linear form. SVM classification are carried out with the functions which shown in (8).

$$f(x) = w.x + b$$

$$f(x) = \sum_{i=1}^{m} a_i\, y_i\, K(x, x_i) + b \tag{8}$$

Where K(x, xi) is the kernel function. The value of w and b can be computed using (9) and (10).

$$w = \sum_{i=1}^{m} a_i\, y_i\, x_i \tag{9}$$

$$b = -\frac{1}{2}\,(w.x_{+} + w.x_{-}) \tag{10}$$

Fig. 7(a) and 7(b) shows the illustration of classifier with maximum margin.

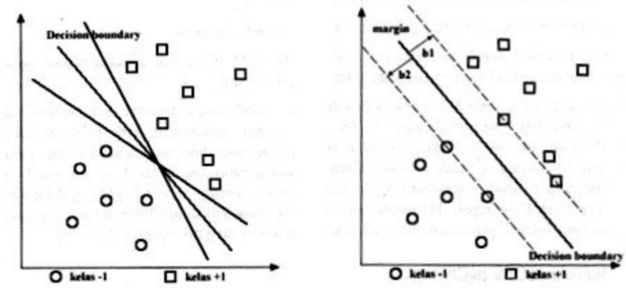


Fig. 7. Linear SVM (a) Decision boundary (b) Decision boundary with maximum margin

2.7. Non-maximum Suppression

Non-maximum suppression is a technique to combine objects in the image with an unknown number of objects. This technique combines the possibility of detection from the same class into one object if the detection results are overlapping. Non-maximum suppression works by eliminating non-maximum values. This process performs a comparison between the value of a pixel with the pixel value around it. If the value is greater, it is maintained, otherwise it is changed to zero. An illustration of the non-maximum suppression is shown in Fig. 8. Based on Fig. 8, the three regions that are colored in green are selected because they are the areas with the highest probability in the non-overlapping area. The area which is colored in red will be suppressed because the occupied area has been occupied by a fraction greater than α by the higher probability area.

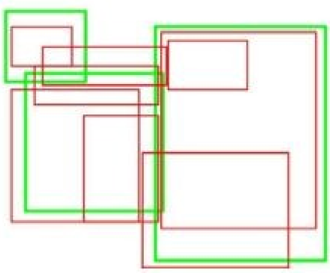


Fig. 8. Illustration of Non-maximum suppression

2.8. Performance Evaluation Metric

The performance evaluation uses detection miss rate and average precision. The metrics measure whether the system is sure of what it has detected as true positive and false positive. This test compares the detection results between the method and human. The evaluation is calculated by matching the coordinates of each detection bounding box X, Y, W, H from the human annotation and compared with the coordinates of the detection bounding box X, Y, W, H predicted by the system. Miss rate and precision of the classification can be calculated using (11) and (12) where TP is the true positive, TN is the true negative, FP is the false positive and FN is the false negative.

$$Miss\ rate = \frac{TP + TN}{TP + FP + TN + FN} \tag{11}$$

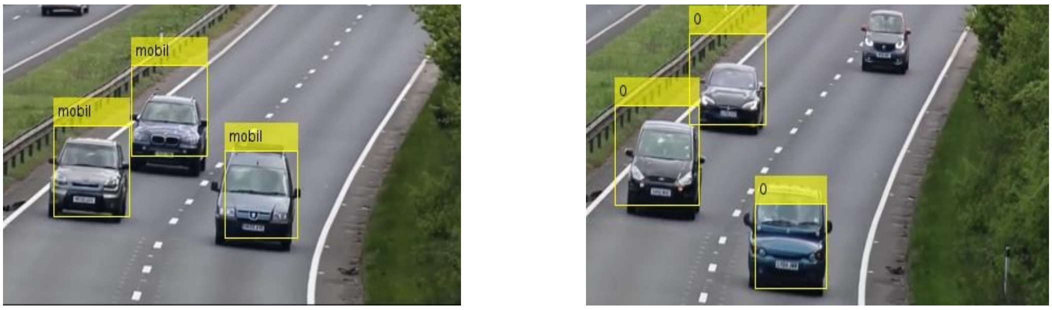
$$Precision = \frac{TP}{TP + FP} \tag{12}$$

3. Result and Discussion

The car detection method was developed using Matlab and runs on laptop with an Intel Core i5 processor and 16 GB RAM. The training data consists of 530 positive image samples and 1,302 negative image samples with a size of 300x300 pixels. This method applies edge detection to see the number of horizontal lines that represents the car objects in the image using Sobel. The features are extracted from the integral image using Haar-like feature then trained using SVM to produce a car detector. The detector is used to detect car objects in the test image. The performance of the proposed method is tested on 20 traffic images captured from 4 CCTV videos.

3.1. Detection Result

The proposed car detection uses Haar-like features to extract features and Linear-SVM to classify cars in the image. Sobel edge detection is used to localize the cars area by utilizing a gradient on the x-axis or horizontal. At the detection stage, a multi-scale sliding window technique is applied with scale starts from 66x66 pixels. The results of car detection (marked by the yellow bounding box) are shown in Fig. 9a for a success detection and Fig. 9b for a failure detection. Based on Fig. 9, the method can detect cars in the image. Sometimes the method fails to detect cars because the car is too close or too far from the CCTV camera and outside the range of multi-scale detection techniques.

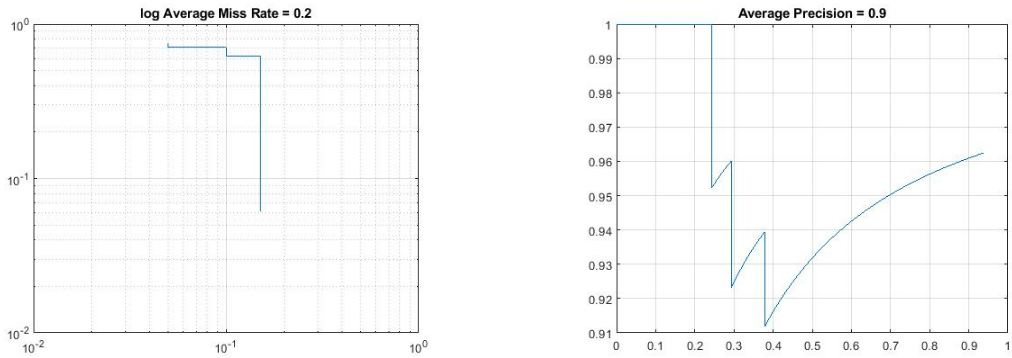


(a) Sample of success detection (b) Sample of failure detection

Fig. 9. Car detection result (a) sample of success detection (b) sample of failure detection

3.2. Performance Evaluation

The performance evaluation of the car detection method is shown in Fig. 10a and Fig. 10b. Based on Fig. 10a and Fig. 10b, the proposed method gets accurate results with an average precision score of 0.9 and average miss rate score of 0.2. The high precision score is due to the effectiveness of the model in estimating the car's frontal pose. The miss rate score happens because the object to be detected is too small or too large due to the cars position that is too far or too close to the camera. Another thing that makes the cars are not detected properly is when the most of the cars are closed with each other. The testing step is carried out by using 20 samples of test image taken from 4 CCTV traffic videos where each image has a resolution of 436x240 pixels. Fig. 11 shows the detection result where the green detection box is the prediction made by the method while the red detection box is the annotation from human.



(a) Log average miss rate evaluation. (b) Average precision evaluation

Fig. 10. Performance evaluation result (a) log average miss rate (0.2) (b) average precision (0.9)

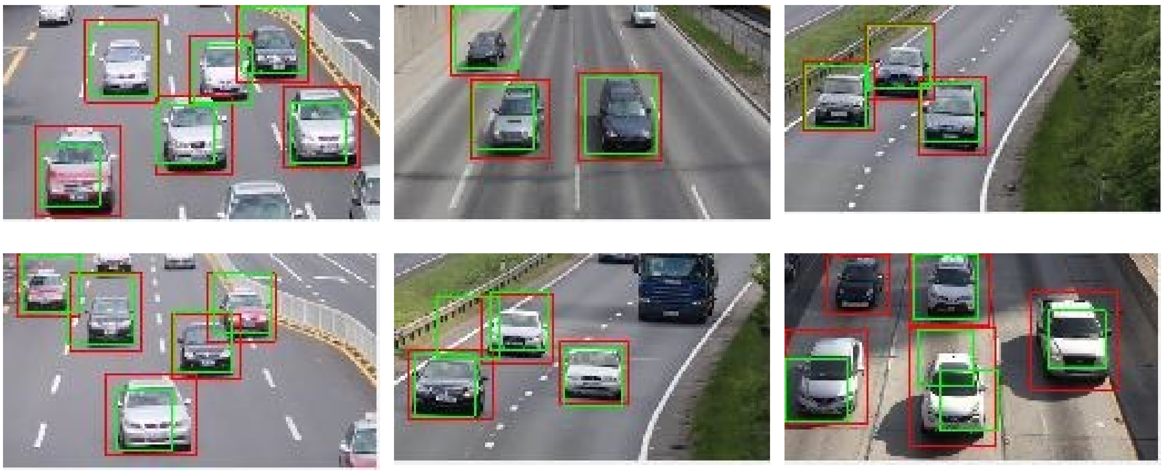


Fig. 11. Bounding box comparison between human annotation and the detection result

4. Conclusion

Based on the performance evaluation of the proposed car detection method, it can be concluded that the method produces accurate results in detecting cars with the average miss rate of 0.2 and average precision score of 0.9. The small miss rate and high precision score indicate that the method is accurate. This research can still be developed to be able to detect cars at night, adding a dataset of cars from another angle so that it can detect various poses and optimizing the method so that it can be applied in real time.

References

[1] S. Messelodi, C. M. Modena, and M. Zanin, “A computer vision system for the detection and classification of vehicles at urban road intersections,” *Pattern Anal. Appl.*, vol. 8, no. 1–2, pp. 17–31, 2005.

[2] Z. Yang and L. S. C. Pun-Cheng, “Vehicle detection in intelligent transportation systems and its applications under varying environments: A review,” *Image and Vision Computing*, vol. 69, pp. 143–154, 2018,

[3] T. Huang, “Computer vision: Evolution and promise,” 1996.

[4] A. Prahara and Murinto, “Car detection based on road direction on traffic surveillance image,”

in *Proceeding - 2016 2nd International Conference on Science in Information Technology, ICSITech 2016: Information Science for Green Society and Environment*, pp. 344–349, 2017.

[5] A. Prahara, A. Azhari and Murinto, “Vehicle pose estimation for vehicle detection and tracking based on road direction,” *Int. J. Adv. Intell. Informatics*, vol. 3, no. 1, pp. 35–46, 2017.

[6] S. Bougharriou, F. Hamdaoui and A. Mtibaa, “Linear SVM classifier based HOG car detection,” in *2017 18th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering, STA 2017 - Proceedings*, vol. 2018, pp. 241–245, 2018.

[7] X. Wen, L. Shao, W. Fang and Y. Xue, “Efficient feature selection and classification for vehicle detection,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 3, pp. 508–517, 2015.

[8] A. Haselhoff and A. Kummert, “A vehicle detection system based on haar and triangle features,” in *IEEE Intelligent Vehicles Symposium, Proceedings*, pp. 261–266, 2009.

[9] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, vol. 1, pp. I-511-I–518, 2001.

[10] ITU-R, “BT.601 : Studio encoding parameters of digital television for standard 4:3 and wide screen 16:9 aspect ratios,” *International Telecommunication Union*, 2011.

[11] I. Sobel, “History and definition of the so-called” sobel operator”, more appropriately named the sobel-feldman operator,” *Sobel, I., Feldman, G.,” A 3x3 isotropic gradient Oper. image Process. Present. stanford Artif. Intell. Proj.*, vol. 1968, 2015.

[12] P. Viola and M. Jones, “Robust real-time object detection,” *Int. J. Comput. Vis.*, vol. 57, p. 137-154, 2004.

[13] U. Kumar, “Vehicle detection in monocular night-time grey-level videos,” in *International Conference Image and Vision Computing New Zealand*, pp. 214–219, 2013.

[14] Y. Tang, C. Zhang, R. Gu, P. Li and B. Yang, “Vehicle detection and recognition for intelligent traffic surveillance system,” *Multimed. Tools Appl.*, vol. 76, no. 4, 2017.

[15] D. Chen, G. Jin, L. Lu, L. Tan and W. Wei, "Infrared Image Vehicle Detection Based on Haar-like Feature," *2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, pp. 662-667, 2018.

[16] S. M. Elkerdawi, R. Sayed and M. ElHelw, “Real-Time Vehicle Detection and Tracking Using Haar-Like Features and Compressive Tracking,” *Springer International Publishing*, pp. 381–390, 2014.

[17] T. Mita, T. Kaneko and O. Hori, “Joint Haar-like features for face detection,” in *Proceedings of the IEEE International Conference on Computer Vision*, vol. II, pp. 1619–1626, 2005.

[18] Y. Wei, Q. Tian, and T. Guo, “An improved pedestrian detection algorithm integrating haar-like features and hog descriptors,” *Adv. Mech. Eng.*, vol. 5, p. 546206, 2013.

[19] C. C. Chang and C. J. Lin, “LIBSVM: A Library for support vector machines,” *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, 2011.

[20] D. Balcones *et al.*, “Real-time vision-based vehicle detection for rear-end collision mitigation systems,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5717 LNCS, pp. 320–325, 2009.

[21] S. Song and J. Xiao, “Sliding Shapes for 3D Object Detection in Depth Images,” in *Computer Vision -- ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VI*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, pp. 634–651, 2014.

[22] G. Guo, S. Z. Li, and K. Chan, “Face recognition by support vector machines,” in *Proceedings - 4th IEEE International Conference on Automatic Face and Gesture Recognition, FG 2000*, pp. 196–201, 2000.

[23] Y. Xu, G. Yu, Y. Wang, X. Wu and Y. Ma, “A Hybrid Vehicle Detection Method Based on Viola-Jones and HOG + SVM from UAV Images,” *Sensors*, vol. 16, no. 8, p. 1325, 2016.