KINETIK, Vol. 3, No. 1, February 2018, Pp. 57-64
ISSN : 2503-2259
E-ISSN : 2503-2267

Vehicle Classification using Haar Cascade Classifier Method in Traffic Surveillance System

Moch Ilham Ramadhani*[1], Agus Eko Minarno[2], Eko Budi Cahyono*[3]
[1,2,3]Universitas Muhammadiyah Malang
moch.ilhamramadhani@gmail.com*

Abstract

Object detection based on digital image processing on vehicles is very important for establishing monitoring system or as alternative method to collect statistic data to make efficient traffic engineering decision. A vehicle counter program based on traffic video feed for specific type of vehicle using Haar Cascade Classifier was made as the output of this research. Firstly, Haar-like feature was used to present visual shape of vehicle, and AdaBoost machine learning algorithm was also employed to make a strong classifier by combining specific classifier into a cascade filter to quickly remove background regions of an image. At the testing section, the output was tested over 8 realistic video data and achieved high accuracy. The result was set 1 as the biggest value for recall and precision, 0.986 as the average value for recall and 0.978 as the average value for precision.

Keywords: *Haar-like Feature, AdaBoost, Cascade Classifier, Vehicle Detection, Digital Image Processing*

1. Introduction

Transportation nowadays is a primary need for every person in finding to most suitable daily transportation. However, there is an existing huge problem. The uncontrolled personal vehicle growth has become one of serious transportation problems. According to the previously conducted research by Indonesia Ministry of Transportation, Indonesian vehicle growth exhibits surprising results, 12% for motorcycle, 8.89% for car, and 2.2% for bus [1]. In addition, the research proves that the need of transportation is increasing every year with many factors triggering this increment exemplified by geography, technology, social and economy [2].

The vehicle growth has increased high traffic density, especially if not being handled properly. This trend can be the main cause of traffic jam [3]. For example, in Indonesia capital city, Jakarta has merely 0.01% of road growth. This imbalance vehicle growth increasing 11% from previous year results bad traffic jam becoming common situation in Jakarta. To solve this problem, there is an option as building more roads or establishing better traffic engineering [2].

Traffic engineering is an operation technique related with planning, geometric design, and traffic operation. One of its procedures is measuring PCU (Passenger Car Unit). As conducted by the previous research [4] to obtain relevant data, researchers manually count passing vehicles with specific criteria to find out the value of PCU. A manual counting by those researchers has high probability for any mistakes or human errors caused by, take for example, environmental conditions or internal disturbances experienced by researchers.

The conventional method to obtain data required big funds and considered as less efficient. Therefore, it has become a trigger for some research for developing programs intended for vehicle counting using video feed in obtaining the data. Afterwards, the data will be processed by using background subtraction method, separating between background and foreground to find the centroid position inside the frame and counting vehicles passing the counting line [5] and [6]. Unfortunately, the program only counts and assumes centroid as a vehicle without any further information on the vehicle type. Hence, in order to obtain PCU, the vehicle type should be recognized because each type of vehicle has different value to count PCU.

In solving this problem, an application to count specific type of vehicle (by focusing only for cars) using traffic video feed as data, processed by Haar Cascade Classifier method, will be important. Previously, this method has been also used in the research [7] to detect traffic cones as obstacles to be avoided by wheeled robot, it was also used in the research [8] to find region of eyes for making region of interest to capture eye winks as an alternative method to input

Ramadhani, M., Minarno, A., & Cahyono, E. (2017). Vehicle Classification using Haar Cascade Classifier Method in Traffic Surveillance System. Kinetik : Game Technology, Information System, Computer Network, Computing, Electronics, and Control, 3(1). http://dx.doi.org/10.22219/kinetik.v3i1.546
Paper submitted on October 11, 2017; Revision on November 02, 2017; Received December 15, 2017

password. According to these research [7], [9] Haar Cascade Classifier has been proven to be an effective and accurate method to detection and specific object recognition.

2. Research Method
2.1 Haar Cascade Classifier
Haar Cascade Classifier is a method utilized for detecting object, also called as Viola Jones method due to its introduction by Paul Viola and Michael Jones for face detection. This method has 4 points for detecting an object, such as Haar-like feature, integral image, AdaBoost learning and Cascade Classifier [8].

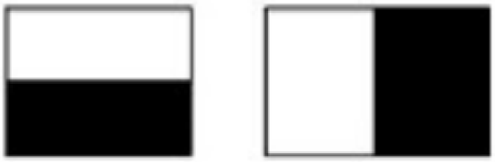


Figure 1. Edge Feature [10]

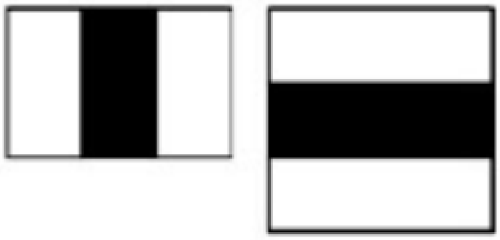


Figure 2. Line Feature [10]

Figure 3. Four-rectangle Feature [10]

Figure 4. Feature Detection Scheme

Haar-like feature is a rectangular feature providing specific indication to an image. Figure 1, Figure 2 and Figure 3 are the examples of common variety of Haar-like feature. Haar-like feature offers high speed computation depending the number of pixels inside the rectangle feature and not depending on each pixel value of the image [8]. In obtaining object detection value, Haar-like feature value was calculated using integral image. Integral image could calculate values accurately and relatively quick by creating new presentation of image by using value of region previously scanned by specific Haar-like feature as shown in Figure 4. The value of integral image was obtained by sum value of previous index, started by left top until right bottom; moreover, Integral image could be calculated using Equation 1, for example the input and new presentation depicted in Figure 5(a) and 5(b).

$$s(x,y) = i(x,y) + s(x,y-1) + s(x-1,y1) + s(x-1,y-1) \quad (1)$$

	1	2	3	4
1	3	7	7	3
2	1	3	3	1
3	5	9	9	5
4	3	6	6	3

a

	1	2	3	4
1	3	10	17	20
2	4	14	24	28
3	9	28	47	56
4	12	37	62	74

b

Figure 5. (a) Input Image and (b) Integral Image of Input Image

Value which had been calculated by using integral image would then be compared with the threshold value of specific features provided by AdaBoost. This should be completed to find potential features because not all features were relevant to use for specific object detection. AdaBoost combines potential features called weak classifier to become strong classifier. Weak classifier means less accurate or also irrelevant prediction [11]. Relevan and irrelevan features shown by Figures 6.

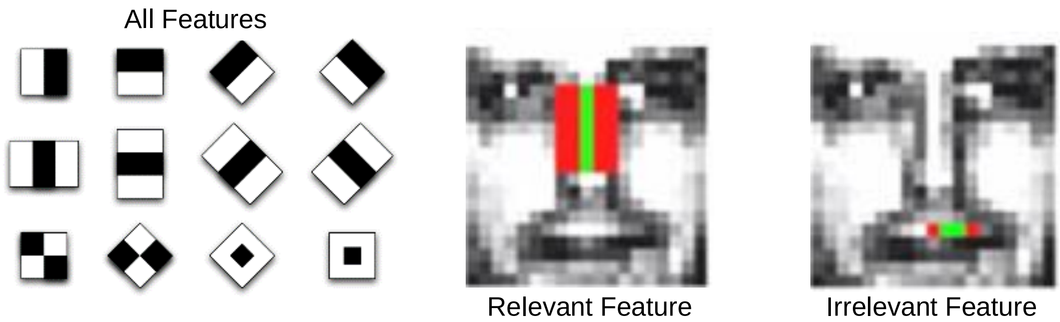


Figure 6. Examples of Relevant and Irrelevant Features

Strong classifier made by AdaBoost can detect object level by level on a cascade. Every sub-window was scanned for specific criteria on each step; furthermore, a sub-window containing positive object was used as a feed for the next level filtering with more specific criteria until obtaining a sub-window which was predicted as a car. On the other hand, a sub-windows not containing positive object was marked as background and separated by sub-window containing positive object as shown by Figure 7 [11].

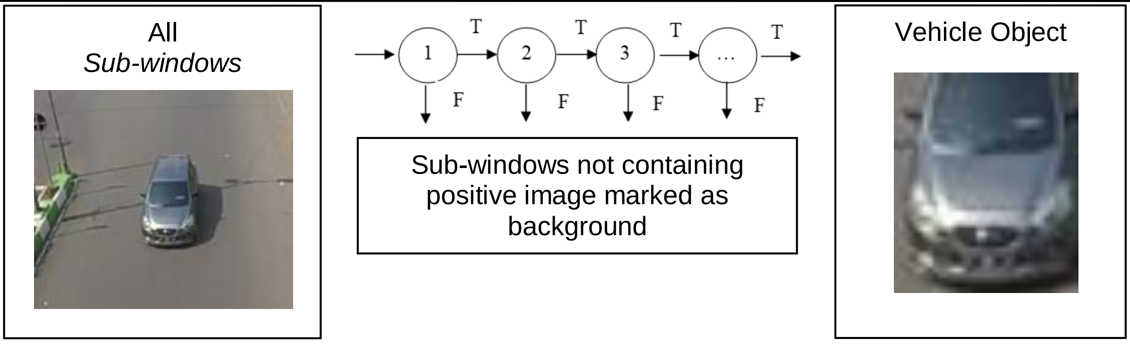


Figure 7. Cascade Object Detection Scheme

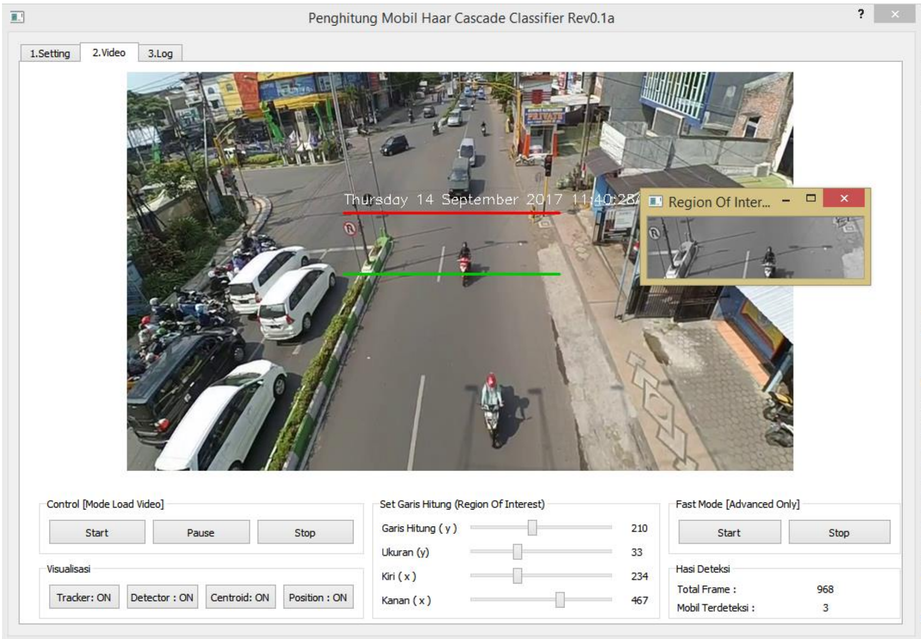


Figure 8. Region of Interest

2.2 Region of Interest

Region of interest is a specific region of an image that is considered as important in this study in obtaining data. Haar Cascade Classifier would not need to scan the entire image regions due to data collection process focused in specific image regions for object detection, object tracking and object counting, illustrated in Figure 8.

2.3 Object Tracking

Object tracking was used to obtain the specific position (x,y) of object inside the frame to be compared with list of previous positions of tracked objects; however, new positions or positions not including on the list of tracked object positions was added as a position (x,y) of a new object. If the new position was included in the list of positions of previous tracked objects, it would be used as a new position of a recognized object. The general process of object tracking is presented by the flowchart in Figure 9.

2.4 Object Counting

Every passing vehicle object inside ROI (Region of Interest) was tracked based on its position and would be compared with the list of tracked object positions. For a new position or position not including in the list of tracked objects, it was be added as a new object and should be counted. If the new position was included in the list of positions of previous tracked objects, it means the position had already been counted as a recognized vehicle. The general process of object counting will be illustrated by the flowchart in Figure 10.

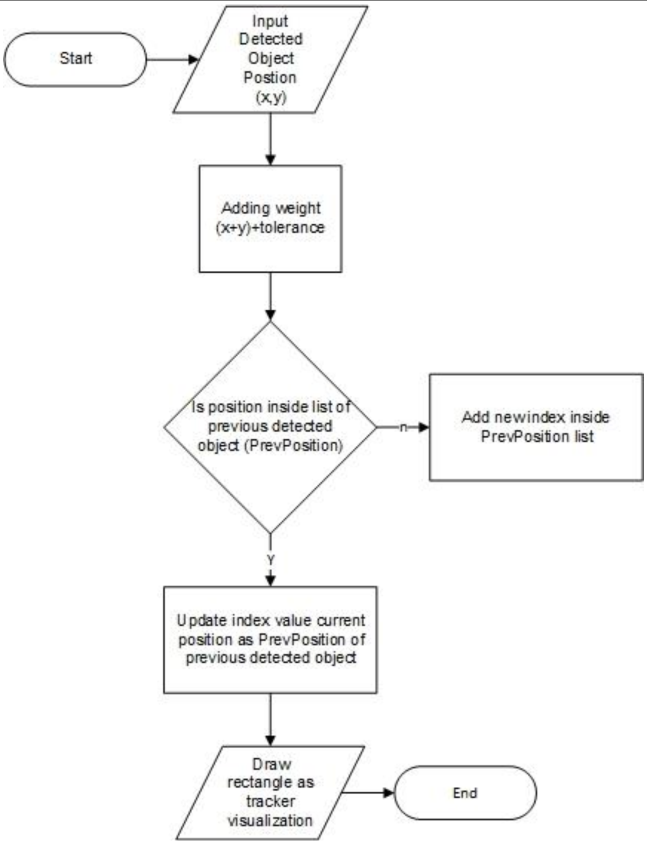


Figure 9. Flowchart of Object Tracking

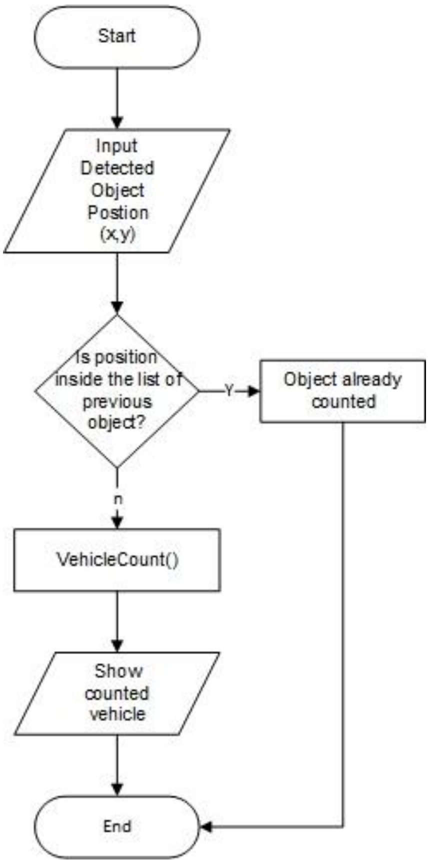
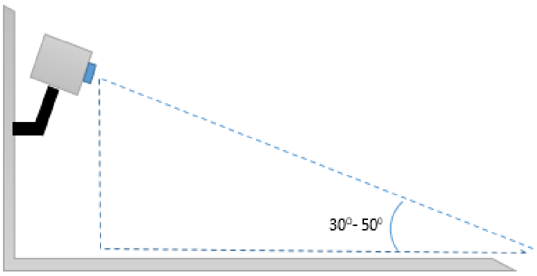


Figure 10. Flowchart of Object Counting

3. Results and Discussion

This section will introduce the result of the vehicle detection efforts. A total of 1000 vehicle images presented in Figure 11 and 200 negative images were used in the training procedure used OpenCV library. Positive samples were obtained by manually cropping and resizing to 30 x 30 pixels. Moreover, the vehicle area of the vehicle images was recorded with traffic surveillance camera depicted in Figure 11. The testing phase employed 8 videos as realistic data test of traffic surveillance video as shown in Table 1. Each video was taken using 30^o – 50^o camera position facing down the road as illustrated in Figure 12(a), and Figure 12(b) presents the video footage.

Figure 11. Positive Image Examples



a

b

Figure 12. (a) Camera Position and (b) Video Result of Camera Position Perseption

Table 1. Video Test Description

No	Video Test	Video Capture Time	Video Stabilizer	Shake Level	Car Total
1.	1.mp4	Afternoon	Tripods	Rare	41
2.	2.mp4	Evening	Hand	Low	71
3.	3.mp4	Morning	Tripods	Rare	50
4.	4.mp4	Evening	Hand	Low	69
5.	5.mp4	Afternoon	Tripods	Rare	41
6.	6.mp4	Evening	Hand	Low	86
7.	7.mp4	Afternoon	Tripods	Rare	60
8.	8.mp4	Afternoon	Tripods	Low	44

3.1 Detection Rate Testing

The detection rate testing measures the detection rate of classifier to obtain recall and precision value of classifier with different scale factor value to determine its value giving the best performance to the classifier.

Table 2. Detection Rate Testing

Video Test	Scale Factor	Actual Total Vehicle	TP	FP	FN	Recall $\left(\frac{TP}{TP+FN}\right)$	Precision $\left(\frac{TP}{TP+FP}\right)$
1.mp4	1.3	41	40	1	0	1	0.975
2.mp4	1.3	71	69	2	0	1	0.971
3.mp4	1.3	50	49	1	0	1	0.980
Average (Scale factor 1.3)						1	0.975
1.mp4	1.5	41	40	1	0	1	0.975
2.mp4	1.5	71	63	8	0	1	0.887
3.mp4	1.5	50	47	4	0	1	0.921
Average (Scale factor 1.5)						1	0.927
1.mp4	1.5	41	40	1	0	1	0.975
2.mp4	1.5	71	67	4	2	0.97	0.943
3.mp4	1.5	50	47	4	0	1	0.921
Average (Scale factor 1.7)						0.99	0.946
1.mp4	1.7	41	38	3	0	0	1
2.mp4	1.7	71	2	69	0	0	1
3.mp4	1.8	50	44	7	0	0	1
Average (Scale factor 1.9)						1	0.593

For data test as shown in Table 2, the classifier was tested using 3 samples of video test and 4 different values of scale factor. Each scale factor value resulted different detection rate. To archive high detection rate, the scale factor value should be determined giving the best performance to the classifier. According Table 2, the "1.3" of scale factor value giving the best performance for the classifier would be used in vehicle counting test section; furthermore, the average value of recall was set as "1", and the average precision value was "0.975".

3.2 Vehicle Counting Test

This section will be testing section for counting car as a final result of following functions such as detecting, tracking and counting.

Table 3. Vehicle Counting Result

No	Video Test	TP	FP	FN	Recall	Precision
1	1.mp4	40	1	0	1	0.975
2	2.mp4	69	2	0	1	0.971
3	3.mp4	49	1	0	1	0.980
4	4.mp4	67	1	3	0.957	0.985
5	5.mp4	41	0	1	0.976	1
6	6.mp4	82	4	4	0.953	0.953
7	7.mp4	59	1	0	1	0.983
8	8.mp4	43	1	0	1	0.977
Average					0.985	0.978

As informed in Table 3, Haar Cascade Classifier was tested using 8 traffic videos and resulted "1" as maximum value for precision and recall, having the average recall set as "0.986" and average precision set as "0.978".

4. Conclusion

This research designs a classification system to determine object as specific type of vehicle. Haar Cascade Classifier proposed by Paul Viola and Michael Jones is used to determine object as car and counted the number of passing vehicles on the specific road using traffic videos as input. The detection rate of this system is affected by the scale factor value, different scale factor value providing varied detection rates. In obtaining high detection rate, the scale factor value giving the best performance to classifier should be determined. In the future, providing accurate and robust vehicle detection system will still be a challenging task in the field of intelligent transportation surveillance systems. The researchers will conduct further research for comparing background subtraction and Haar Cascade Classifier method or combining background subtraction and Haar Cascade Classifier to detect more specific vehicle types.

5. Notation

s(x,y)	: Representation of Integral Image (x,y)
i(x,y)	: Current Representation of Image (x,y)
Positive Image	: Recognized Object
Negative Image	: Not Recognized Object or Background Image
TP	: True Positive or Detected Vehicle
FP	: False Positive or Undetected Vehicle
FN	: False Negative or Missed Classification/Missed Detection

References

[1] Ministry of Transportation, "*Transportation Statistics*," Vol. I, 2016.
[2] Central Bureau of Jakarta Province, "*Transportation Statistics of Jakarta 2015*," 2015.
[3] N. Redhantika, “Traffic Density in Malang City," Universitas Merdeka Malang, Pp. 1–10, 2014.
[4] R. Anwar, “*Measuring Passenger Car Unit in Banjarmasin Municipality*," Vol. 1, No. 1, Pp. 22–27, 2000.
[5] Fajar Mit Cahyana, “*Design Counting Program on the Number of Vehicles on Uni-directional Traffic Using C++ Programming Language with OpenCV Database*," Universitas Brawijaya, 2014.
[6] A. Helmi, “*Application of Traffic Density Level Detection Based on the Number of Passing Vehicles with OpenCV*,” 2015.
[7] C.-J. Lee, “Obstacle Detection and Avoidance Via Cascade Classifier for Wheeled Mobile Robot,” Int. Conf. Mach. Learn. Cybern., Pp. 5, 2015.
[8] P. Viola and M. M. J. Jones, “*Robust Real-time Face Detection*,” Int. J. Comput. Vis., Vol. 57, No. 2, Pp. 137–154, 2004.
[9] M. Syarif, P. Studi, T. Informatika, F. I. Komputer, U. Dian, and N. Semarang, “*Blink Detection with Haar Cascade Classifier and Contour for Password Login*,” Techno.com, Vol. 14, No. 4, Pp. 242–249, 2015.
[10] A. Mordvintsev, “*OpenCV-Python Tutorials Documentation Release Beta*,” 2017.
[11] J. Howse, "OpenCV Computer Vision with Python," 2013.