IJCCS (Indonesian Journal of Computing and Cybernetics Systems)
Vol.14, No.3, July 2020, pp. 231~242
ISSN (print): 1978-1520, ISSN (online): 2460-7258
DOI: https://doi.org/10.22146/ijccs.54050

HOG Feature Extraction and KNN Classification for Detecting Vehicle in The Highway

Firnanda Al Islama Achyunda Putra*[1], Fitri Utaminingrum[2], Wayan Firdaus Mahmudy[3]
[1,2]Universitas Brawijaya , (0341) 577911, Indonesia
[3]Faculty of Computer Science, Universitas Brawijaya, Malang, Indonesia
e-mail: ***[1]firnandaaap@student.ub.ac.id**, [2]f3_ningrum@ub.ac.id, [3]wayanfm@ub.ac.id

Abstrak

Mobil otonom adalah kendaraan yang dapat membimbing dirinya sendiri tanpa campur tangan manusia. Berbagai jenis kendaraan tanpa kemudi sedang dikembangkan. Sistem masa depan di mana komputer mengambil alih seni mengemudi. Masalahnya adalah sebelum menjadi perhatian dalam mobil otonom untuk mendapatkan keamanan yang tinggi. Mobil otonom membutuhkan sistem peringatan dini untuk menghindari kecelakaan di depan mobil, terutama sistem yang dapat digunakan di lokasi jalan raya. Dalam tulisan ini, kami mengusulkan sistem deteksi kendaraan berbasis visi untuk mobil Otonom. Algoritme deteksi kami terdiri dari tiga komponen utama: ekstraksi fitur HOG, klasifikasi KNN, dan deteksi kendaraan. Ekstraksi fitur telah digunakan untuk mengenali objek seperti mobil. Dalam hal ini, kami menggunakan ekstraksi fitur HOG untuk mendeteksi sebagai mobil atau non-mobil. Kami menggunakan algoritma KNN untuk mengklasifikasikan. Klasifikasi KNN dalam penelitian sebelumnya memiliki hasil yang cukup baik. Mobil terdeteksi dengan mencocokkan data trining dengan data pengujian. Trining data dibuat dengan mengekstrak fitur HOG dari gambar 304 x 240 piksel. Sistem akan menghasilkan klasifikasi antara mobil atau non-mobil.

Kata kunci—*Histogram of Oriented Gradient (HOG), K-Nearest Neighbour (KNN), Deteksi Kendaraan*

Abstract

Autonomous car is a vehicle that can guide itself without human intervention. Various types of rudderless vehicles are being developed. Various countries have created many future systems where computers take over the art of driving. The problem is prior to being attentive in an autonomous car for obtaining high safety. The autonomous car needs an early warning system to avoid accidents in front of the car, especially the system can be used in the Highway location. In this paper, we propose a vision-based vehicle detection system for Autonomous car. Our detection algorithm consists of three main components: HOG feature extraction, KNN classifier, and vehicle detection. Feature extraction has been used to recognize object such as cars. In this case, we use HOG feature extraction to detect as a car or non-car. We use the KNN algorithm to classify. KNN Classification in previous studies had quite good results. Car detected by matching about trining data with testing data. Trining data created by extract HOG feature from image 304 x 240 pixels. The system will produce a classification between car or non-car.

Keywords— *Histogram of Oriented Gradient (HOG), K-Nearest Neighbour (KNN), Vehicle Detection*

Received Februari 4th,2020; Revised June 9th, 2020; Accepted July 30th, 2020

1. INTRODUCTION

The highway is the primary access to connect the road out of the city. Indonesia has a lot of highways. Base on data published from databooks in 2019, Indonesia has a 759 km highway on java island. The increasing length of the highway in Indonesia will be proportional to the number of accidents. Base on data published by PT Lintas Marga Sedaya, cipularang highway has the largest number of accidents in Indonesia. Total accidents in 2016 are 1,282 accidents, then in 2017 are 1,381 accidents and in 2018, there were 1,197 accidents. The number of accidents in the last three years is 3860 car. Base on the data, Dehradun (India) has an 11.574 accident between 2009 to 2014 [1]. The accident value in Indonesia or another country is still high.

Human behavior is a determining factor in safety traffic. It is very important that drivers are aware of the safety implications of their driving. The leading causes of accident on the highway are long distances and long periods. That makes the driver feel bored and feel sleepy. Referring to the current development of technology, we can minimize the risk for road users. We can give early notification when the vehicle in front of the driver. This system can be implemented on the autonomous car or warning sign.

Autonomous car uses various kinds of technologies. They can be built with GPS sensing knowledge to help with navigation. They may use sensors and other equipment to avoid collisions. Autonomous car has the ability to use a range of technology, where a vehicle displays information to the drivers in new and innovative ways. The autonomous car needs an early warning system to detect another vehicles in front of the car to avoid an accident. We created a detection system to develop a safety system in an autonomous car. This system can be used for an autonomous car without a driver or autonomous car still using the driver. Early warning systems use to increase safety to the drivers when in front of the car have another vehicles. Hence, the early warning system in the autonomous car is integrated between the buzzer and output system of the camera, when the vehicles detected, then the buzzer will be on.

Research uses HOG feature extraction to detect an object [2][3]. HOG feature extraction is used to detect humans [4]–[7], and the camera will follow the person [8]. Before extracting the feature into a HOG, the image feature is converted to greyscale. in this research, feature extraction HOG has a pretty good performance to detect objects. Researches on the Advanced Driver Assistance System (ADAS) are developed quickly in recent years [9]–[12]. Driving systems should minimize driving difficulty under various driving conditions, thereby reducing traffic accidents. Vehicle detection is beneficial for the driver to reduce the number of traffic accidents. Our system uses one camera to retrieve data sets, which will later be useful as training data. By using only one camera, the system we made is quite simple but very useful. The system that we made classifies car or non-car with a distance that we specify.

Vehicle detection has contribution in many applications, such as automatic driving systems [13] and driver-assistance systems. Many researchers develop a vehicle detection method to obtain good performance, such as by combining HOG and SVM [14]. Research conducted by [14] uses overlapping techniques. In our proposed method capable for detecting the position of the vehicle in several places based on sliding window process. In this research, we use the KNN classification because in previous studies has a good accuracy [15]. Meanwhile, the performance of computation time could be improved by sklearn's tools. Hence, we conducted research by combining between HOG to obtain feature extraction value, KNN for classification Vehicle in the Highway.

In this paper, we purpose an algorithm to detect vehicles such as car or non-car. We use HOG feature extraction then classified by KNN Classification. The rest of the paper is organized as follows: In section II, the detailed information vehicle detection algorithm is described. Section III, the performance of our algorithm is presented, and section IV is the conclusion described.

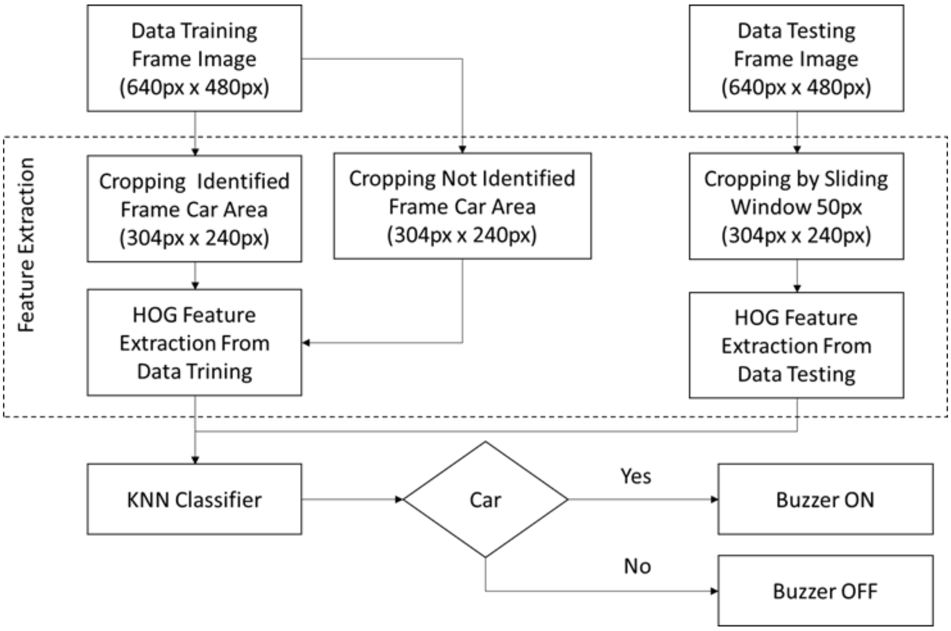


Figure 1 Block System

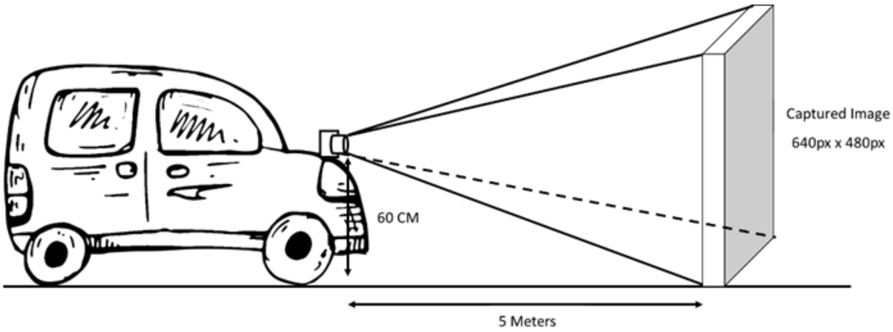


Figure 2 Capturing data to obtain Dataset.

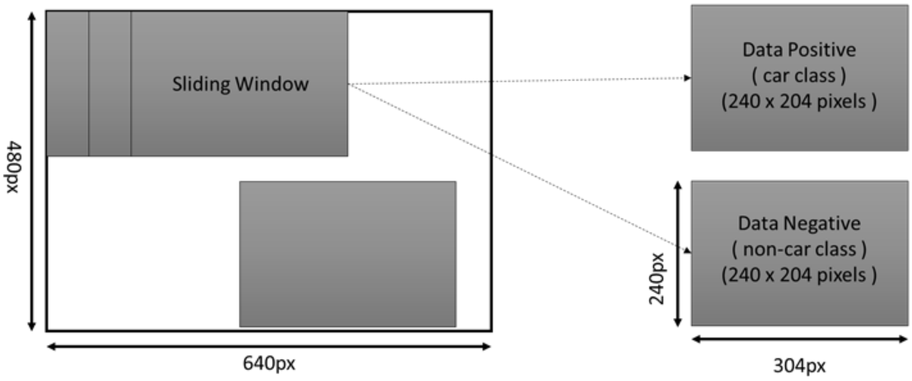


Figure 3 Dataset Cropping

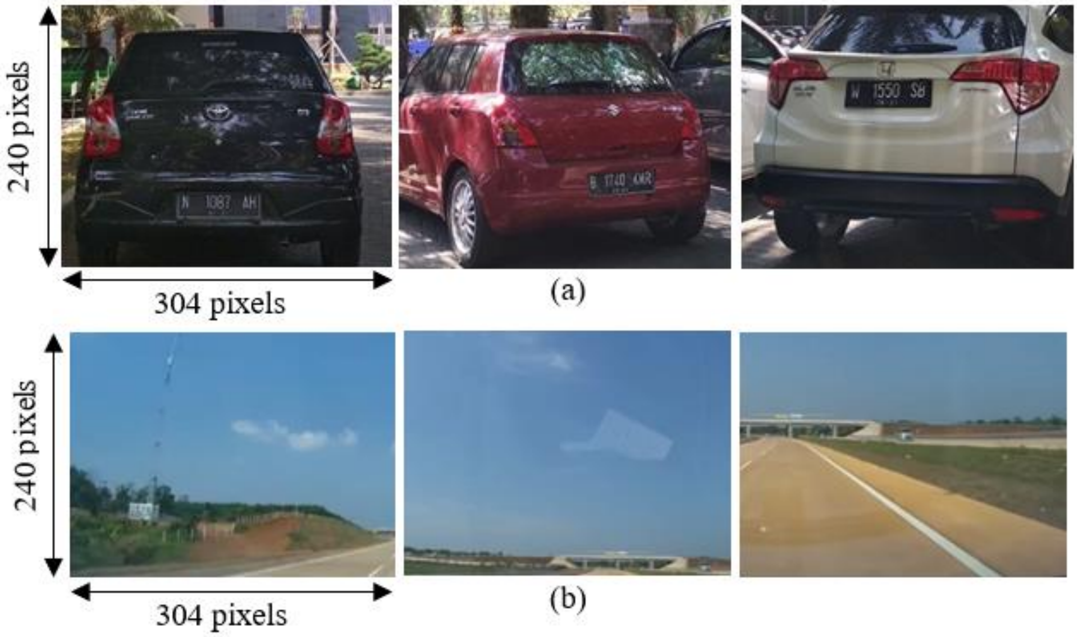


Figure 4 Data acquisition (a) training for car class and (b) non-car class

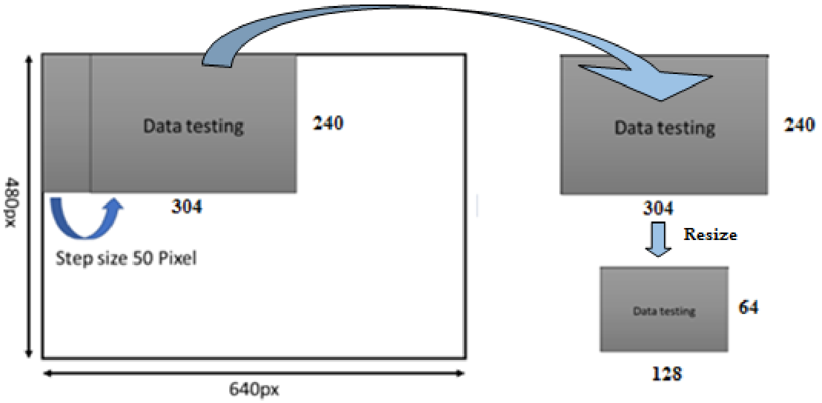


Figure 5 Process testing data

2. METHODS

This research is divided into three important parts, shown in Figure. 1. First, is data set acquisition, second is HOG feature extraction and the last is classified with KNN classification. HOG feature extraction has many of features. HOG feature extraction has 3780 features for one frame [14]. HOG feature extraction, it will be more detailed in feature extraction. Referring to previous research that has been conducted by L. Mao, the object detection process uses overlapping techniques. Over lapping technique requires a long time, because it requires many classifications in one frame and just only can be able to detect a single obstacle in each frames. Meanwhile, in our proposed method capable for detecting the position of the vehicle in several places based on sliding window process. It has a good accuracy and the performance of computation time could be improved by sklearn's tools, so that it can be implemented in real

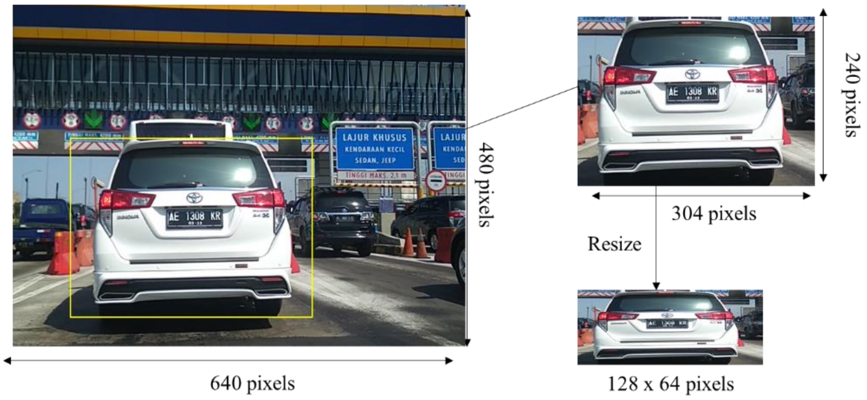


Figure 6 Preprocessing cropping by sliding windows and resizing

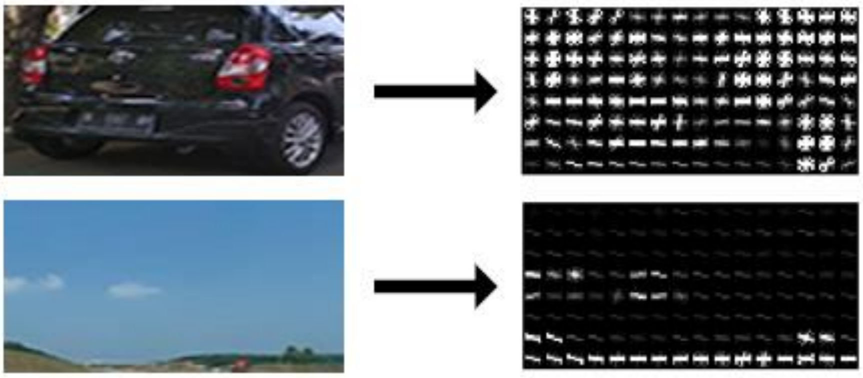


Figure 7 Visual HOG feature Extraction

time conditions. Sliding window use for cropping data and matching between training data and data testing. Cropping images of the dataset, as shown in Figure. 3.

In this paper, we used KNN Classification tools, that has been improved speed performance by sklearn's tools in Python. KNN Classification uses sklearn's tools, only takes 0.4 second. Besides of that, to increase the speed performance, we resize the 240x304 window data to become 64x128. So, by resizing the size of data window will be increased the speed of KNN performance. For real-time implementation; we used 24 fps video, so we can process 5 frames per seconds. The process of testing data, as shown in Figure. 5.

2.1 Dataset Acquisition

Dataset taken by camera video with resolution 640 x 480 pixels. We convert the video to an image for training data. The number of data sets is 200 images, 100 is a car image, and 100 is a non-car image. Car dataset is taken from several directions like right, left, or center. From 100 data, we took 33 from the left, 33 from the right, and 34 from the middle. Types of cars tested in this study are city cars excluding trucks. For more clearly, is shown in Figure. 4(a). The number of non-car data sets is 100 images. We use mobile phone xiaomi redmi 4 camera. Our camera placed on the fix position in front of the car. The high camera position is 60 cm that was measured from the street. The distance between the car to captured data is 5 meters.

The distance between the camera and the captured data can be illustrated in Figure. 2. Data is divided into two data, namely car and non-cars. The data set is taken by cropping a section into pixels with a size of 304 x 240 pixels. We crop with a size of 304 x 240 pixels because at that pixel size, the car can really be fully identified. For more clearly, is shown in Figure. 2 and Figure. 4. We created a trining data by cropping the image to the same size. We use 50 step sizes in sliding windows because when comparing data testing, it doesn't take too long. Cropped images produce 30 pictures in 1 Frame. From the truncated image, we divide the data into two classes, namely car class and non-car class.

2. 2 HOG Feature Extraction

Histogram of Oriented Gradient (HOG) provides excellent performance relative to other existing features. The basic hypothesis is that local object appearance and shape can often be characterized rather well by the distribution of local intensity gradient or edge directions, even without precise knowledge of the corresponding gradient or edge positions [16][17]. The aim of HOG is to describe an image with a local-oriented gradient histogram. These histograms represent occurrences of specific gradient orientation in local parts of the image [18]. HOG feature extraction consists of several important steps.

2. 2.1 Preprocessing

We have an original image captured from the camera. Image size captured from the camera is 640 x 480 pixels. We have selected a patch of size 304 × 240 pixels for calculating our HOG feature descriptor. The spot is cropped from the original image more clearly, as shown in Figure. 3. We resize the patch selected for calculating the HOG feature value. HOG needs a specific size to calculate the feature value. For getting the 3860 feature value, we must resize the image to 126 x 64 pixels. Figure 6 explains the process of pre-processing.

2. 2.2 Calculate the gradient image

The HOG Descriptor calculated by the horizontal and vertical gradient. This method needs one feature kernel, and feature kernel can use a Sobel operation to count x gradient from image (Gx) and (Gy).

$$Gx = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \tag{1}$$

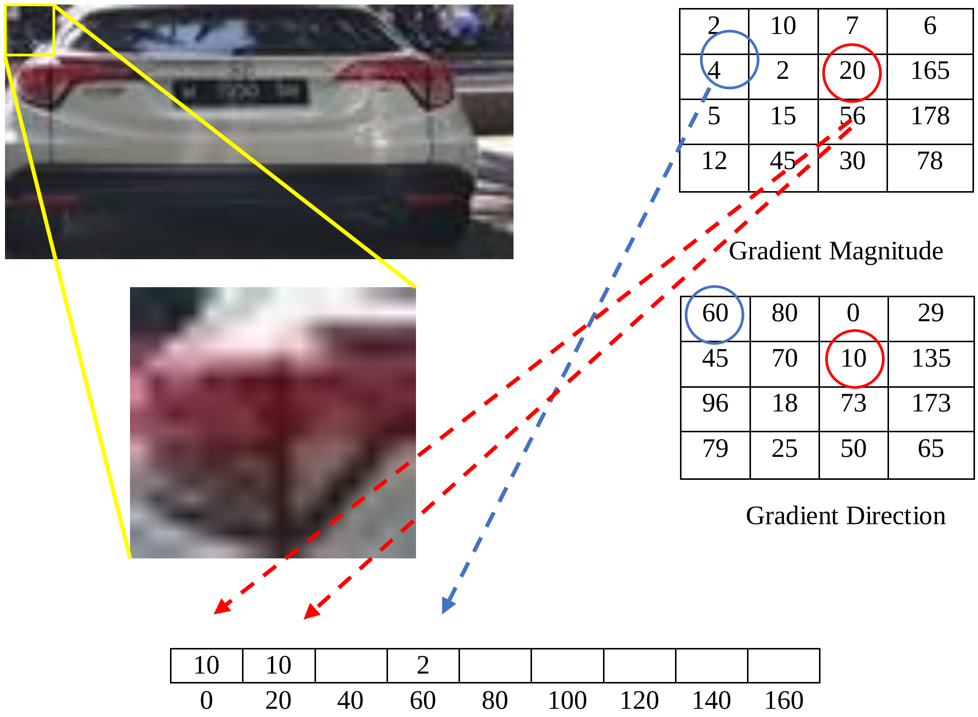
$$Gy = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \tag{2}$$

G_x *= Sobel Horizontal matrics operation*
G_y *= Sobel Vertical matrics operation*

Gradient direction indicates the direction in which the most significant image changes. We calculated from the sum of the square root of the Gradient x (Gx) and (Gy). Gradient direction value can be calculated by Eq. 3 [19] .

$$G = \sqrt{Gx^2 + Gy^2} \tag{3}$$

G = Gradient Direction
G_x *= Sobel Horizontal matrics operation*
G_y *= Sobel Vertical matrics operation*



2	10	7	6
4	2	20	165
5	15	56	178
12	45	30	78

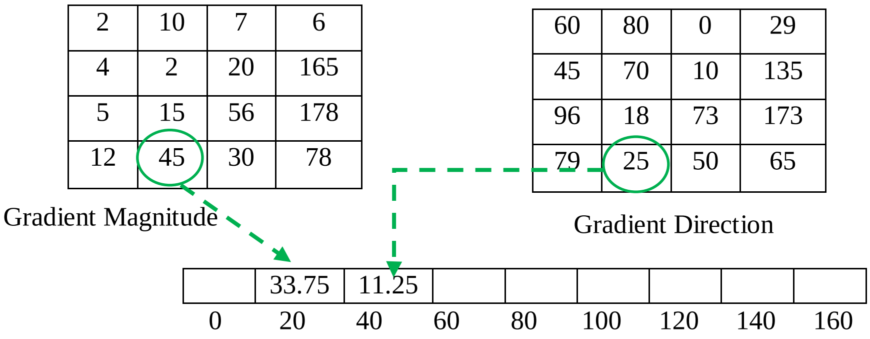
Gradient Magnitude

60	80	0	29
45	70	10	135
96	18	73	173
79	25	50	65

Gradient Direction

0	20	40	60	80	100	120	140	160
10	10		2					

Figure 8 HOG features. Each block consists of a grid of spatial cells.



2	10	7	6
4	2	20	165
5	15	56	178
12	45	30	78

Gradient Magnitude

60	80	0	29
45	70	10	135
96	18	73	173
79	25	50	65

Gradient Direction

0	20	40	60	80	100	120	140	160
	33.75	11.25						

Figure 9 Vote value

Gradient magnitude contains information about how much change has occurred. The value of Gradient magnitude can be calculated with Eq. 4 [19].

$$R_0 = arctan\frac{Gy}{Gx} \tag{4}$$

R_0 = *Gradient Magnitude*
G_x = *Sobel Horizontal matrics operation*
G_y = *Sobel Vertical matrics operation*

2. 2.3 Calculate Histogram of Gradient in 8x8 cells

Image divided to 8x8 pixel on 1 cell. Figure. 5 explain about 192 pixel value (8 x 8 x 3). Block has two values, the gradient value, and the magnitude value. The gradient at the pixel encircled using red has an angle of 10 degrees and a magnitude of 20. Since 10 degrees is halfway between 0 and 20, the vote by the pixel splits evenly into the two bins. For more clearly look at Figure. 7. We have several information that must be solved related to the problem. How about the value is not middle. For example, look at Figure. 8. The range of H is 20. We use Eq. 5 [19] to calculated vote value for fill the bin value.

$$Vote = \frac{\Delta R_{\mathrm{i}}}{H} * G_{\mathrm{i}} \ and \ \frac{1 - \Delta R_{\mathrm{i}}}{H} * G_{\mathrm{i}} \tag{5}$$

R_i = Orientation Gradient Value
H = Range of angle value
G_i = Gradient Value
ΔR_i = Upper limit of the signed gradient

2. 2.4 16x16 Block Normalization

In the previous step, we created a histogram based on the gradient of the image. Gradients of an image are sensitive to overall lighting. We would like to “normalize” the histogram, so they are not affected by lighting variations. You can work it out yourself to see that normalizing [256,128,64] will result in [0.87, 0.43, 0.22], which is the same as the normalized version of the original RGB vector. You can see that normalizing a vector removes the scale Eq. 6 and 7 [19].

$$L^1 - norm : v' = \frac{v}{\sqrt{||v||^2 + \mathcal{E}}} \tag{6}$$

$$L_2 - norm : v' = \frac{v}{\sqrt{||v'||_2^2 + \mathcal{E}^2}} \tag{7}$$

v = Blog features that contain a histogram
$\mathcal{E}$ = 0,1
L–norm = Normalization

2. 2.5 16x16 Block Normalization

In Each detection, a window is divided into cells of size 8 x 8 pixels, and each group of 2 x 2 cells are integrated into a block in a sliding fashion, so blocks overlap with each other. In this case, The HOG feature extraction need to have an aspect ratio of 1:2. For example, they can be 105 X 210 pixels, 128 X 256 pixels, or 1000 X 2000 pixels but not 102 X 203 pixels. The gradient magnitudes vote into the orientation histogram as shown Figure. 7. Each block contains a concatenated vector of all its cells. In other words, each block is represented by a 36-0 feature vector that is normalized to an L2 unit length . Each 64 x 128 detection window is represented by 7 x 15 blocks, giving a total of 3780 features per detection window.

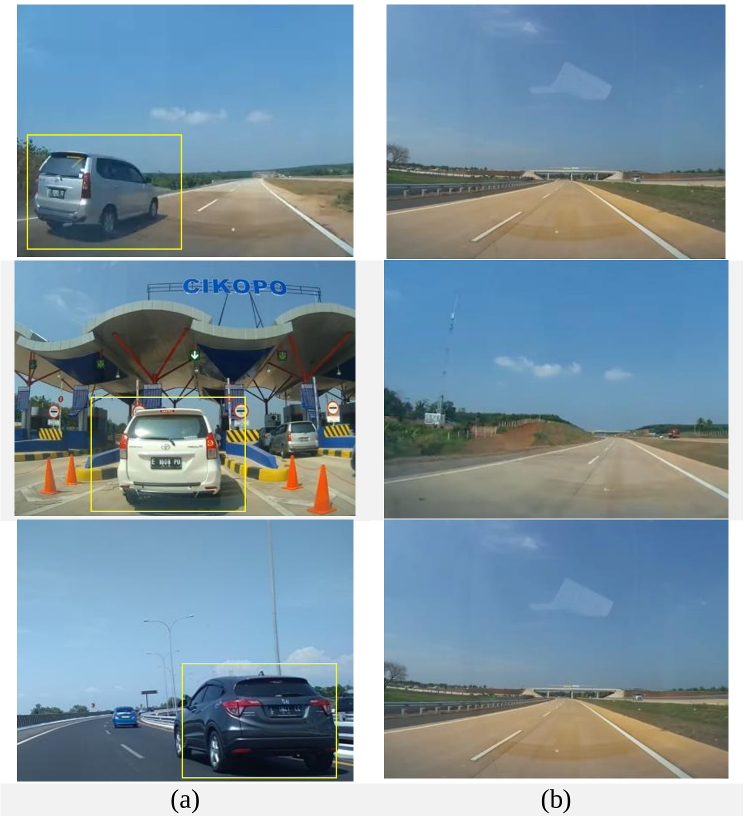


Figure 10 visual result (a) is a car detected and (b) car not detected

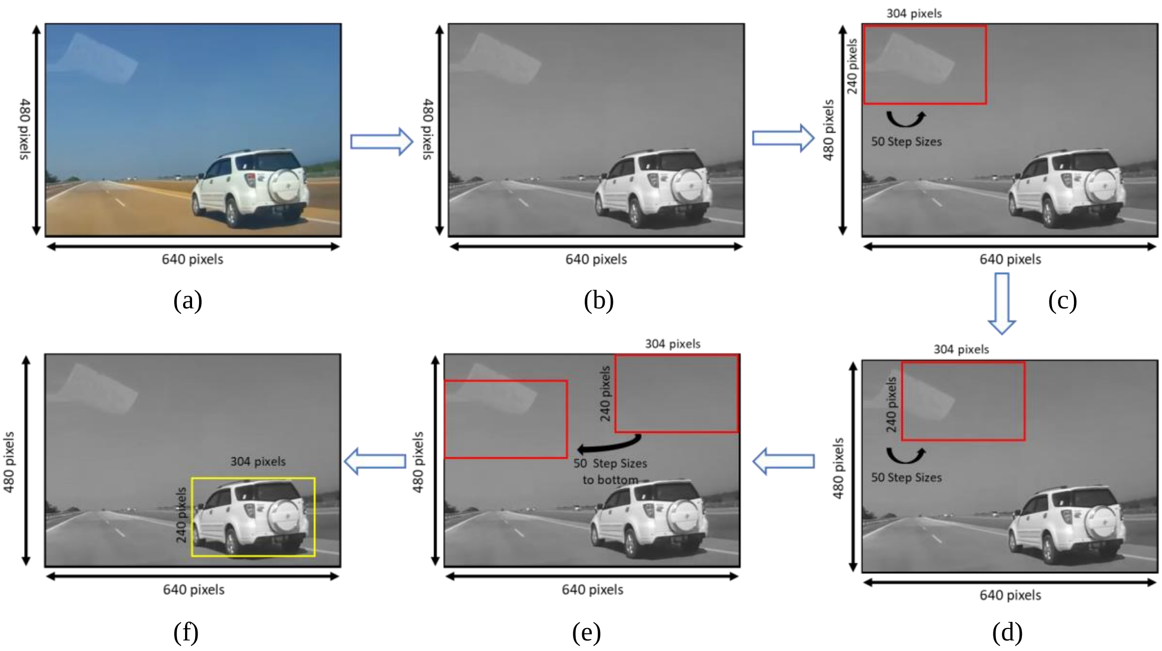
2. 3 KNN Classification

In our work, K-Nearest Neighbours (KNN) algorithms are used for classification about car or non-car. K-Nearest Neighbours (KNN) is a very simple algorithm, easy to implement supervised machine learning algorithm that can be used to solve both classification problems. Euclidean distance can be calculated by Eq. (8).

$$ED = \sqrt{\sum_{i=1}^{k} (x_i - y_i)^2} \tag{8}$$

ED = *Euclidean Distance*
x_i = *Test Data*
y_i = *Training Data*
k = *Number of Neighbors*

The k-value has to be chosen carefully. If the k-value is too small, it will be too sensitive to noise. If the k-value is too large, the neighborhood may also include other classes [20]. ED is a weighted feature value. When the ED is large, it will be a very far neighbor from a small ED value. The result of classification is a large value will not be one class with a small value, because the distance is very far. We determine k = 1,4 to find the nearest neighbor due to this number is the best number after the experiment base on accuration.



(a) (b) (c)

(f) (e) (d)

Figure 11 Visual Analysis bounding matching process

3. RESULTS AND DISCUSSION

Evaluate the performance of the purposed methode, we use different location about testing data with training data. Training data is taken from 2 different types. Training data we collect includes parking areas and highways. The proposed algorithm is evaluated on a group of test images captured from several images. The camera is mobile phone xiaomi redmi 4 equal when we take a set of data. Camera mount fix in front of a car.

3. 1. Visual Analysis

The camera takes a video, and the system captures an image with 640x480 resolution. The camera is placed in front of the car. The captured image produces the RGB color scale. RGB color has three-channel colors there are R-channel, G-channel, and B-channel. Our system only processes the one-channel of color space. So, the RGB color space must be converted to grayscale. The system takes data with a 304x240 window size from the grayscale image. The window will be moved to the right and down to take data. In every steps of window movement is 50-pixels as shown in Figure 11. In this process will be obtained 30 data with the size 304x240 pixels. So, we can detect the position of the vehicle in several places based on the 30 data's that were obtained.

In this session, we will discuss about visual analysis. Image detected as a car then it will be bounded by yellow color. Size of the bonding is same as when we crop the data set at the beginning. Size of bounding is 304 x 240 pixels. We choose that number of ratio because car will be easyly to detect. Figure. 9. are presented about visual analysis. Car detected presented in Figure 10 (a). Car not detected present on Figure 10 (b). Yellow bounding box present about car detected.

3. 2. Accuracy

We compare between the actual condition and the result system to analysis the performance of the proposed method. We took 50 random test data, where 25 data were car and

25 data were non-car. Accuracy can be calculated based on visual output with the output obtained from the system.

Table 1. The accuracy of true car images

Total Data Training	Actual Condition		Detection Result		Accuration Rate %
	Car	Non Car	Car	Non Car	
K1	25	-	21	4	84%
	-	25	5	20	80%
K4	25	-	22	3	88%
	-	25	4	21	84%
Rate					84%

3. 3. Comparison result with the previous methode

In previous research by [21] is using HOG feature extraction and uses a Support Vector Machine (SVM) classification.

Table 2. Comparison with previous

Methode		Accuration Comparation
Feature Extraction	Classification	
HOG	SVM	83%
HOG	KNN	84%

4. CONCLUSIONS

We use single camera for detection car or non-car. We use HOG feature Extraction and K-Nearest Neighbors algorithm to classified car or non-car. Figure. 6 show some representative detection results. The bounding boxes superimposed on the original image indicate the final detection. Figure. 6 shows detection results on simply structured roads. This is the easiest traffic scenario for any vision-based on-road vehicle detection system. Our system worked very well under this scenario. The experimental results show this system can achieve high target reliability detection. Accuration rate in our system is 84%. Other methode for detecting car or non-car is by [21] the accuration in this paper is 83%.

REFERENCES

[1] S. Kumar and D. Toshniwal, "A data mining framework to analyze road accident data," *J. Big Data*, vol. 2, no. 1, 2015.

[2] D. Konstantinidis, T. Stathaki, V. Argyriou, and N. Grammalidis, "Building Detection Using Enhanced HOG-LBP Features and Region Refinement Processes," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 10, no. 3, pp. 888–905, 2017.

[3] S. Qi, J. Ma, J. Lin, Y. Li, and J. Tian, "Unsupervised Ship Detection Based on Saliency and S-HOG Descriptor From Optical Satellite Images," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 7, pp. 1451–1455, 2015.

[4] M. Bilal, A. Khan, M. U. K. Khan, and C. M. Kyung, "A Low-Complexity Pedestrian Detection Framework for Smart Video Surveillance Systems," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 10, pp. 2260–2273, 2017.

[5] N. D. Nguyen, D. H. Bui, and X. T. Tran, "A Novel Hardware Architecture for Human Detection using HOG-SVM Co-Optimization," *Proc. - APCCAS 2019 2019 IEEE Asia Pacific Conf. Circuits Syst. Innov. CAS Towar. Sustain. Energy Technol. Disrupt.*, pp.

33–36, 2019.

[6] M. Bilal and M. S. Hanif, “Benchmark revision for HOG-SVM pedestrian detector through reinvigorated training and evaluation methodologies,” *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 3, pp. 1277–1287, 2020.

[7] M. J. Flores Calero, M. Aldas, J. Lazaro, A. Gardel, N. Onofa, and B. Quinga, “Pedestrian detection under partial occlusion by using logic inference, HOG and SVM,” *IEEE Lat. Am. Trans.*, vol. 17, no. 9, pp. 1552–1559, 2019.

[8] F. Utaminingrum *et al.*, “Human guide tracking using combined histogram of oriented gradient and entropy difference minimization algorithm for camera follower,” *J. Telecommun. Electron. Comput. Eng.*, vol. 9, no. 4, pp. 49–54, 2017.

[9] K. P. Divakarla, A. Emadi, and S. Razavi, “A cognitive advanced driver assistance systems architecture for autonomous-capable electrified vehicles,” *IEEE Trans. Transp. Electrif.*, vol. 5, no. 1, pp. 48–58, 2019.

[10] M. Hasenjager, M. Heckmann, and H. Wersing, “A Survey of Personalization for Advanced Driver Assistance Systems,” *IEEE Trans. Intell. Veh.*, vol. 5, no. 2, pp. 335–344, 2020.

[11] J. Chen, M. Bhat, S. Jiang, and D. Zhao, “Advanced driver assistance strategies for a single-vehicle overtaking a platoon on the two-lane two-way road,” *IEEE Access*, vol. 8, pp. 77285–77297, 2020.

[12] A. Bisoffi, F. Biral, M. Da Lio, and L. Zaccarian, “Longitudinal Jerk Estimation of Driver Intentions for Advanced Driver Assistance Systems,” *IEEE/ASME Trans. Mechatronics*, vol. 22, no. 4, pp. 1531–1541, 2017.

[13] Z. Sun, G. Bebis, and R. Miller, “On-road vehicle detection: A review,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 5, pp. 694–711, 2006.

[14] L. Mao, M. Xie, Y. Huang, and Y. Zhang, “Preceding vehicle detection using histograms of oriented gradients,” *2010 Int. Conf. Commun. Circuits Syst. ICCCAS 2010 - Proc.*, pp. 354–358, 2010.

[15] K. R. Skoien, M. O. Alver, and J. A. Alfredsen, “A computer vision approach for detection and quantification of feed particles in marine fish farms,” *2014 IEEE Int. Conf. Image Process. ICIP 2014*, pp. 1648–1652, 2014.

[16] X. Cao, C. Wu, P. Yan, and X. Li, “Linear SVM classification using boosting HOG features for vehicle detection in low-altitude airborne videos,” in *2011 18th IEEE International Conference on Image Processing*, 2011, pp. 2421–2424.

[17] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” *Proc. - 2005 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognition, CVPR 2005*, vol. I, pp. 886–893, 2005.

[18] M. Cheon, W. Lee, C. Yoon, and M. Park, “Vision-Based Vehicle Detection System With Consideration of the Detecting Location,” *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 3, pp. 1243–1252, 2012.

[19] Y. Li and G. Su, “Simplified histograms of oriented gradient features extraction algorithm for the hardware implementation,” *Proc. - 2015 Int. Conf. Comput. Commun. Syst. ICCCS 2015*, pp. 192–195, 2016.

[20] P. A. Raktrakulthum and C. Netramai, “Vehicle classification in congested traffic based on 3D point cloud using SVM and KNN,” *2017 9th Int. Conf. Inf. Technol. Electr. Eng. ICITEE 2017*, vol. 2018-Janua, pp. 1–6, 2017.

[21] S. Bougharriou, F. Hamdaoui, and A. Mtibaa, “Linear SVM classifier based HOG car detection,” *2017 18th Int. Conf. Sci. Tech. Autom. Control Comput. Eng. STA 2017 - Proc.*, vol. 2018-Janua, pp. 241–245, 2018.