Journal of Systems Engineering and Information Technology
Vol. 1 No. 1 (2022) Page 33 – 39 | e-ISSN: 2829-310X
Journal Homepage: www.jurnal.iaii.or.id/index.php/JOSEIT

Implementation Counting and Yolo Object Detection Methods for Identification Degree of Road Saturation

Rico Aditya Utama
Faculty Master of Electrical Engineering, Gunadarma University

Email corresponding author: ricoaditya59@gmail.com

Abstract

Identification of road conditions such as congestion at this time still requires manual efforts in getting results. This research will discuss how to identify the level of congestion by paying attention to the parameters of the degree of road saturation. The congestion parameters certainly need to be monitored, especially in crowded areas and big city business centers, such as on Sudirman Road, Central Jakarta. Many parameters to identify the level of road congestion, one of them is by observing the value of the degree of road saturation based on MKJI (Indonesian Road Capacity Manual). The purpose of this study is to propose a system that can calculate the value of the degree of saturation quickly and accurately using a camera. The method that is being proposed is to combine the Computer Vision with YOLO Object Detector techniques based on Deep Learning and the Object Counting method to get the value of traffic flow in the observed area. The results obtained by this system are good, this is supported by the error value obtained by the system around 3-4% for data photo and 10-11% for data video.

Keywords: Computer Vision, Object Detection, YOLO, Object Counting, Degree of Road Saturation.

1. Introduction

In big cities in Indonesia where the number of residents and motorists is large, such as Medan, Surabaya and Jakarta, congestion is always a source of traffic problems. Congestion is a situation of stalling or even stopping traffic flow caused by the number of vehicles exceeding the road capacity. Many parameters can be used as a reference to determine the value of congestion on a road, one of which is the degree of road saturation. The degree of road saturation is the ratio of traffic flow to road capacity. This research will discuss how to identify the level of congestion by paying attention to the parameters of the degree of road saturation.

The solution that has been carried out by the government especially in Jakarta to overcome congestion is the 3 in 1 program and the application of the odd-even vehicle plate crossing schedule rules as applied to the Jend road section. Sudirman. This 3 in 1 system emerged based on the Decree of the Governor of the Special Capital Region of Jakarta, No. 4104/2003 dated 23 December 2003 aims to reduce the density of traffic flow at certain hours based on the number of passengers in one vehicle. This system is considered ineffective because of the emergence of lively 3 in 1 jockeys. Another policy that has recently been implemented is the odd-even policy. This policy requires restrictions on vehicles (4-wheeled vehicles) that pass on certain segments based on the type of vehicle's license plate. If the date on that day is odd, then vehicles that may pass on that section must have odd numbered vehicles and vice versa.

In the last few years, there has been a lot of research on road traffic monitoring, especially using camera media. Based on research [1] in Morocco applying 3 methods, namely, segmentation, classification, and vehicle counting to detect and count the number of vehicles passing through. Y.Iwasaki [6] developed an algorithm to measure the queue length for improving the timing of traffic light, his approach is based on dividing each lane into blocks and detect the presence of vehicles in these blockes by using the Sobel operation.

This paper examines the same concept but with a different method approach and road conditions based on ground rules in Indonesia which commonly called Indonesian Road Capacity Manual especially in Jakarta.

Therefore we need a system that can produce values of degree of saturation on a highway accurately and automated. Technology that is very possible to design this system is a combination of the Deep Learning method (Using the YOLO algorithm: You Only Look Once) and Object counting to produce the value of traffic flow as one of the parameters to calculate the degree of saturation in a road under certain conditions has been observed.

Accepted by editor: 18-05-2020 | Final revision: 18-10-2020 | Online publication : 25-03-2022

2. Method

The methods used in this study are Object Detection, Vehicle Detection, Vehicle counting and Calculate "Degree of Road Saturation" Value.

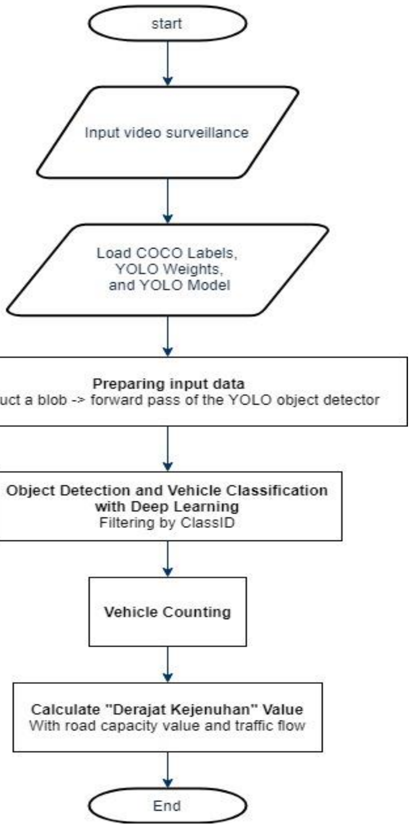

Figure 1. Flowchart system

2.1. Preparing Input Data

Before the frame is forwarded as input from a convolutional neural network from the YOLO deep learning model that has been loaded into the system, the frame must pass preprocessing data first, called the BLOB. In OpenCV, it provides a function to do image preprocessing for deep learning classification, namely cv2.dnn.blobFromImage.

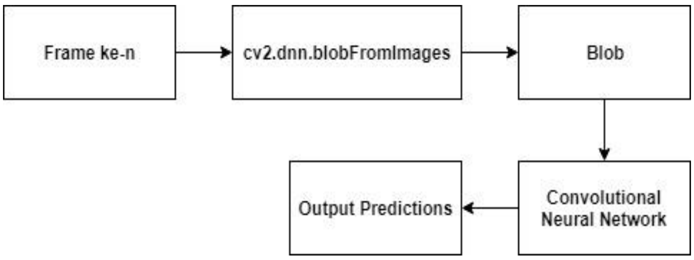

Figure 2. Blob processing

The function performs 2 processes for preprocessing data, namely:

1. Mean Subtractor, and
2. Scaling by some factor.

The result of this function, returns the value in the form of a list / array for one frame / image which will later become the convolutional neural network input of the YOLO deep learning model.

2.2. Object Detection and Vehicle Classification with Deep Learning

To do object detection YOLO uses a very different approach from the previous algorithm, which is applying a single neural network to the entire image. Input image will be resized to size 448 × 448 pixels then entered into a single convolutional network for prediction process. This network will divide the image into regions and then predict boundary boxes and probabilities, for each boundary area box weighed the probability to classify as objects or not. In this study the value of the confidence level limit is 0.5.

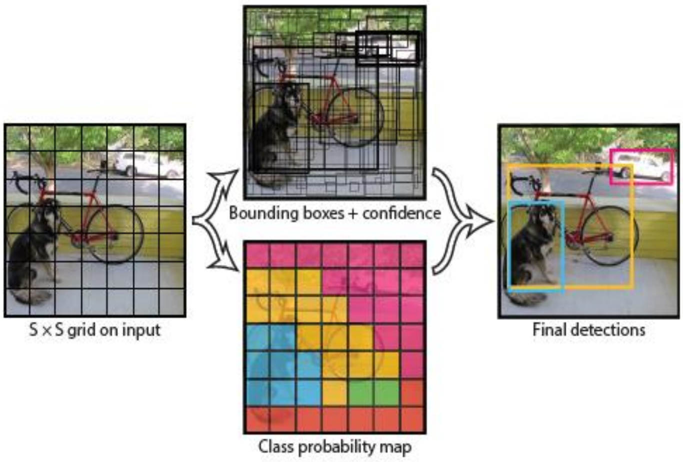

Figure 3. YOLO Ilustration

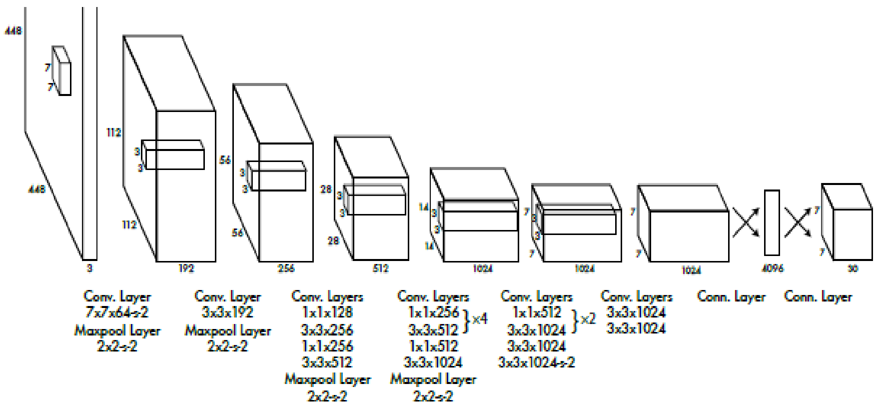

Figure 4. YOLO Architecture

In this study, the authors also use the OpenCV library as a platform to run YOLO. The OpenCV version used is OpenCV 3.4.4 written in Python version 3.6.7. the YOLO version used is YOLOv3 which has been trained using the COCO dataset. COCO dataset is a large-scale dataset library for object detection and segmentation. The COCO dataset used consists of 80 labels of objects such as bicycles, cars, planes, trucks, cats, traffic signs, chairs, tables, etc.

To do Vehicle detection is to filter classID from the COCO dataset label. ClassID which includes vehicles in the COCO dataset are classID 2 (Car), classID 3 (Motorbike), classID 5 (Bus), and classID 7 (Truck).

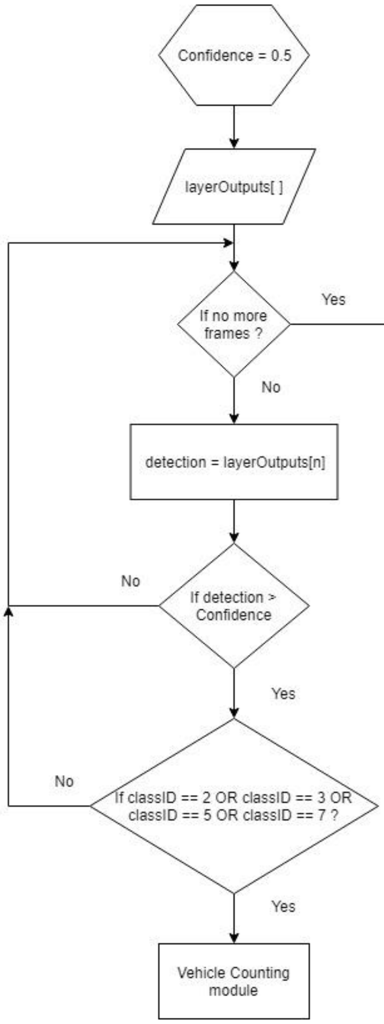

Figure 5. Object Detection and Vehicle Detection module

2.3. Vehicle Counting

To calculate the vehicle passing on a road segment being observed is done by drawing a line in the frame to be a reference to the centroid value on the vehicle object.

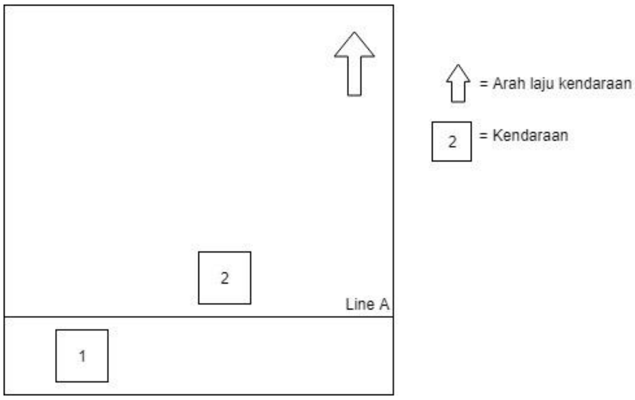

Figure 6. Vehicle counting Illustration

If the centroid in each vehicle box passes Line A, then the vehicle variable will be added by 1.

DOI: https://doi.org/10.29207/joseit.v1i1.1965

2.4. Understanding the degree of road saturation.

The degree of road saturation is defined as the ratio of traffic flow to road capacity. Often used as a parameter of road congestion / key factor determining the level of performance of a road segment (MKJI 1997). Based on the value of Road Traffic Degree (DS), the level of road service can be searched using the following table.

Table 1. The level of road conditions is based on QC

Level	Condition	Ratio (Q/C)
A	Free flow at high speed.	0.00 – 0.20
B	The current is stable, the speed needs to be limited by road conditions, the driver is free to choose the speed.	0.21 – 0.44
C	The flow is stable but the speed is limited by traffic conditions, the driver is limited in choosing the speed.	0.45 – 0.74
D	The current is approaching unstable, but can still be tolerated.	0.75 – 0.84
E	Volume approaches capacity, current is unstable	0.85 – 1.00
F	Traffic jams, low speed.	> 1.00

2.5. Calculate Degree of Road Saturation value

The last step is calculating the degree of saturation with the following equation.

$$DS = Q / C \qquad (1)$$

Which DS is Degree of road saturation (smp/hour). Q is Traffic flow (smp/hour). And C is Road Capacity (smp/hour).

The value of traffic flow (Q) is obtained from the final value of the variable vehicle after the counting process. While the value of road capacity (C) is obtained from the following formula.

$$C = Co * FCw * FCsp * FCsf * FCcs \qquad (2)$$

Which C is Road Capacity (smp/hour). Co is Base Capacity (smp/hour), FCw is Traffic width adjustment factor, FCsp is Road separator adjustment factor, FCsf is Side drag adjustment factor, and FCcs is City size adjustment factor.

3. Result and Discussion

For running the program, hardware specifications used in this research are :

a. CPU Intel i5 Gen 8th.
b. Memory RAM 4Gb.
c. Operating System : Ubuntu 18.04.2 LTS

The trial phase is divided into 2 scenarios, namely to capture photos and to capture video. Observation data from 2 scenarios are taken from the same place, namely Sudirman Road in Central Jakarta on July 21, 2019 at 17:17 WIB.

a. Calculate Degree of Road Saturation.
To calculate the value of the degree of road saturation that needs to be done first is to calculate the value of the capacity of the road in this study to take the sample road Jend. Sudirman Central Jakarta.

$$C = (25\ smp/menit * 1 * 1 * 0.97 * 1.04) * 4\ Lajur = 100{,}88\ smp/menit. \qquad (4)$$

If the value of the road capacity has been found the next step is to calculate the value of the degree of saturation, as follows.

DS = Traffic flow (per minute) / 100.88 smp/minute (5)

b. Test of photo capture.
The trial of capturing photos is done by taking photo data and representing one photo frame as one data per minute, then the photo data is input into the system. The results are as follows.

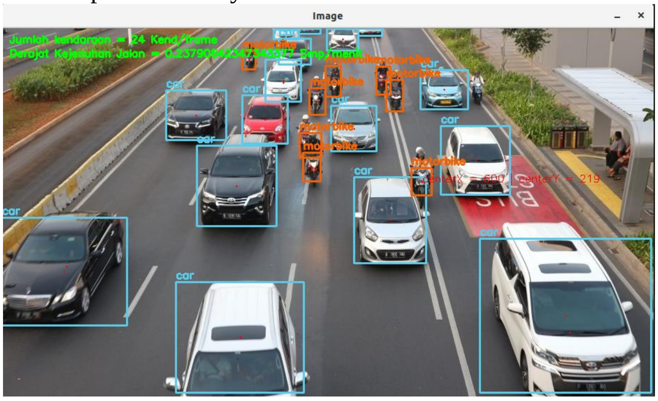

Figure 7. *The system results from photo capture data*

Table 2. Variable results of the photo data system

Variable Name	The results obtained in the system
Number of vehicles (Q)	24 Vehicles / Frame
Degree of road saturation (DS)	0.238

After the measurement results obtained by the system, it is necessary to validate the system results to find out how accurate the system has been made. Validation is done by comparing the results of the system with the results of manual calculations and will get the error value. The error value is calculated using the following formula.

$$Error = \frac{(Result\ manual - Result\ system)}{Result\ manual} \times 100\ \% \qquad (3)$$

Table 3. Validation of photo data results

Variable	The results system	Actual count	Error (%)
Q	24 Vehicles / Frame	25 Vehicles / Frame	4 %
DS	0.238	0.248	3 %

c. Test of video.
In testing the video capture is done by recording a video on the road segment Sudirman Central Jakarta then the video data is input into the system. The results are as follows.

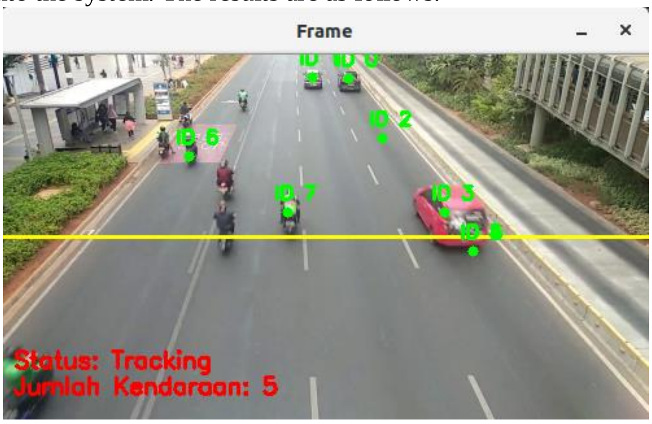

Figure 8. *System results from video data*

Table 4. Variable results of the video data system

Variable Name	The results obtained in the system
Number of vehicles (Q)	94 Vehicles / Minute
Degree of road saturation (DS)	0.015

Table 5. Validation of video data results

Variable	The results system	Actual count	Error (%)
Q	94 Vehicles / Minute	105 Vehicles / Minute	10.5
DS	0.015	0.017	11.4

4. Conclusion

YOLO is a deep learning object detection model that is suitable for use if we rely on a fast system with little importance to accuracy in its detection. Vehicle detection can be quickly detected in real time conditions.

For next research, to speed up the execution process and minimize errors, the image processing computation process should be done on the GPU (Graphic Processing Unit) not on the CPU as has been done in this study.

References

[1] Zakaria Moutaki, dkk. 2018. Real-Time System Based on Feature Extraction for Vehicle Detection and Classification. Transport and Telecommunication. 19(2): 93-102.
[2] Margit Betke, dkk. 2000. Real-time multiple vehicle detection and tracking from a moving vehicle. 12:69-83.
[3] Mrs. P. M. Daigavane, Dr. P. R. Bajaj. 2010. Real-time vehicle detection and counting method for unsupervised traffic video on highways.
[4] Uma Nagaraj, dkk. 2013. Traffic jam detection using image processing. 3(2):1087-1091.
[5] M. Sujatha, dkk. 2017. Traffic congestion monitoring using image processing and intimation of waiting time. 115(7):239-245.
[6] Yoichiro Iwasaki. 1998. An image processing system to measure vehicular queues and an adaptive traffic signal control by using the information of the queues. In Intelligent Transportation System, IEEE Conferenc. Pages 195-200.
[7] M. Fathy and M. Siyal. 1998. A window-based image processing technique for quantitative and qualitative analysis of road parameters. IEEE Transportation on Vehicular Technology. 47(4):1342-1349.
[8] Li Wei, Dai Hong-ying. 2016. Real-time road congestion detection based on image texture analysis. Procedia Engineering. 137:196-201.
[9] Huang, L., et al., 2014. Research on road congestion state detection based on machine vision. Journal of China Computer Systems, 35(1), 148-153.
[10] Wang, C., 2010. Research of urban traffic congestion detection based on video processing. Ph.D. thesis, Chongqing University, Chongqing, China.
[11] Yu Qiao, Zhongke Shi. 2012. Traffic Parameters Detection Using Edge and Texture. Procedia Engineering. 29:3858-3862.
[12] Redmon, Joseph., Santosh Divvala., Ross Girshick., Ali Farhadi. 2016. "You Only Look Once: Unified, Real-Time Object Detection".
[13] Direktorat Jendral Bina Marga. 1997. Manual Kapasitas Jalan Indonesia. Jakarta.
[14] https://devtrik.com/opencv/mengenal-opencv-open-source-computer-vision-library/.
[15] Dr. Adrian Rosebrock, "Deep Learning for Computer Vision with Python", September 2017.
[16] Jennie Kusumaningrum. 2011. Analisa Perbandingan Kinerja Jalan Jendral Sudirman Sebelum dan Sesudah Perpanjangan Waktu Three In One dan Penerapan Busway. Dinamika Teknik Sipil. 11(1) : 70-80.
[17] Wahban Al Okaishi,a dkk. 2019. Vehicular Queue Length Measurement Based On Edge Detection and Vehicle Feature Extraction. Journal of Theoretical and Applied Information Technology. 97(5) : 1595-1603.