Adaptive Traffic Light Controller Based on Congestion Detection Using Computer Vision

R G Putra¹*, W Pribadi¹, I Yuwono¹, D E J Sudirman¹, B Winarno¹

Politeknik Negeri Madiun, Indonesia

Email: gusta@pnm.ac.id¹, why.pribadi@pnm.ac.id¹, indarto@pnm.ac.id¹, dirvi@pnm.ac.id¹, basuki@pnm.ac.id¹

Abstract. The transportation sector plays an important role in realizing a smart city. The increase in the number of vehicles is currently not supported by an increase in road capacity. Traffic jams or congestion will occur in many places. Congestion will increase the accident rate, bad effect on economic growth, and increase gas emissions. Effective traffic management is necessary to reduce congestion levels and its side effects. A traffic light is one of traffic management methods. Traffic lights control the flow of traffic at road intersections, zebra crossings, and other traffic flow points. Conventional traffic lights work on a pre-programmed time sequence. This system is effective if the vehicle density is relatively constant. The density of vehicles from various directions fluctuates with time. To increase the effectiveness of using traffic light, an adaptive system is needed. In this study, a simple adaptive traffic light mechanism was developed based on congestion on the road using computer vision. Vehicle congestion is detected using the YOLOv3 object detection which detects the type of vehicle. The detection system used by YOLOv3 with pretrained weight COCO has a true positive value for motorbikes of 60%, cars (light vehicles) 93%, and trucks/buses (heavy vehicles) 100%. The processing speed of the Jetson Nano mini-PC with the OpenCV library on the GPU is 2 times faster than the process with the CPU.

1. Introduction

To support the realization of a smart city, the interconnection between fields is very important, including the transportation sector. The transportation sector plays a very important role to support regional progress. However, the increase in the number of vehicles on the road is not supported by an increase in road capacity. This causes many congestion points to occur. This congestion can increase the accident rate, affect economic growth, and increase gas emissions [1]. The implementation of effective traffic management is necessary to reduce the level of congestion and its side effects.

Traffic lights have an important role in the implementation of traffic management. Traffic lights control the flow of traffic installed at crossroads, zebra crossings, and other traffic flow points. Traffic regulation at road junctions is intended to regulate vehicle movement in each vehicle movement group. With this regulation, it is hoped that traffic can move regularly. Conventional traffic lights work sequentially based on a pre-programmed time sequence. This system is effective on roads with constant vehicle density at all times. The density of vehicles from various directions fluctuates. Therefore, to increase the effectiveness of using traffic lights, an adaptive system is needed.

There have been many studies conducted to increase traffic effectiveness. Among the research that has been done is monitoring and tracking area density based on the use of wifi / Bluetooth [2], adaptive traffic light based on the calendar [3], smart traffic management based on complex sensor
data [4], Traffic management based on mechanical and electronic devices [5], using wireless sensor network [6]. Most of the research uses specialized and complex sensors. The use of CCTV cameras will make the system simpler and can be easily implemented.

Research on monitoring using image processing has also been widely applied. Among them is a smart traffic light based on image processing [7], in this study, vehicle detection is based on moving vehicles. This will be an obstacle when the vehicle stops. Complex mechanisms are also carried out using artificial intelligence [8]. In this study, it is not explained how the data acquisition system from the sensor point and the use of complex mechanisms. In this study, a simple adaptive traffic light mechanism was developed based on image processing which is hoped to be easily applied in the field. The purpose of this study is to design a vehicle congestion monitoring system with image processing and determine the density of vehicles based on the reading of the number of vehicles to become a reference for adaptive traffic lights. Vehicle density is detected using YOLOv3 object detection based on the type of vehicle. The detection system used by YOLOv3 with pretrained weight COCO (Common Object in Context)

2. Research Method

The block diagram of the system in this study and the devices used are shown in Figure 1. The device consists of an IP camera as an image sensor, a hub/switch to connect to the network, and an image processing unit using nVidia Jetson Nano. The system process flow is shown in Figure 2. The camera takes a traffic picture at a traffic-light point when it turns red. A Region of Interest (ROI) is selected, which is an area to be observed so that it is not affected by others. Furthermore, it is processed to obtain the number of existing vehicles consisting of motorbikes, cars, trucks, and buses. Processing is carried out by providing a value for each type of vehicle. the sum of the vehicles multiplied by the value is then calculated. These results were then used as a reference time setting at traffic lights. The vehicle weight values are shown in Table 1. MC means motorbike, LV means light vehicle and HV means heavy vehicle.

| No | Type    | Values |
|----|---------|--------|
| 1  | Motorbike (MC) | 0.5    |
| 2  | Car (LV)    | 1      |
| 3  | Truck (HV)  | 1.3    |
| 4  | Bus (HV)    | 1.3    |

Fig. 1. System block diagram.

Table 1. The vehicle weight values
The vehicle load calculation for each point based on the congestion of the vehicle is shown in equation 1. Where $B_n = point \ load\ n$, $J = number\ of\ vehicles\ of\ each\ type$, $w = value\ of\ loading$, $k = type\ of\ vehicle$. To determine the length of time the green light at traffic lights is done adaptively based on the accumulation of vehicles at that point. The calculation can be done based on the following Pseudocode.

a. $t_{green} = \frac{\text{total\ weight}}{\text{normal\ weight}}$

b. if $t_{green} < \min_{green}$:
   $t_{green} = \min_{green}$

c. elif $t_{green} > \max_{green}$:
   $t_{green} = \max_{green}$

$t_{green}$ is the counting period of the green light, total_weight is the calculation result based on equation 1, normal_weight is the value of the vehicle load during normal conditions, min_green is the minimum green time calculation period, max_green is the maximum green time calculation period. Minimum green is used to limit the minimum time to remain tolerant of normal values. Maximum green is also used to limit the length of green time is not excessive. Tests performed include testing of True Positive, False Negative, and time Processing in frames per second (fps). FPS is obtained based on the speed of the tool processing one image frame.

3. Result and Analysis

Testing is carried out in the field by taking a road image and then applying the object detection method. The test location was carried out at one of the intersections in Caruban District, Madiun Regency. Data were collected randomly during the day. The hardware prototype is shown in Figure 3 below. The equipment used is nVidia jetson nano, power supply, and a switch/hub that is connected to a camera that has been installed at the intersection point.
3.1. False Positive and False Negative Test Results

The results of image capture from the camera are shown in Figures 4 and 5 below. The image shows the results of image capture processed by selecting the Region of Interest (RoI) with the area in the red line. Objects outside the region of interest do not need to be detected because they are outside the road area. After the ROI process, object detection is carried out using YOLOv3. The results of the detection are then processed by weighting and equations as described. The results of this process are then used as a reference for determining the green light time.

![Fig. 4. Detection result 1](image1)

![Fig. 5. Detection result 2](image2)

The results of the object detection test are shown in Table 2 and the following graph in Figure 6. The detection method uses Yolov3 where the weight used is the weight that has been trained in the COCO (common object in context) scheme. The detection of objects for motorbikes is used to detect people because the results of detecting motorbikes with the COCO scheme are far from expectations. From the detection results in several trials, it was found that the true positive test results of motorbikes were 60%, cars or light vehicles (LV) was 93%, and heavy vehicles (HV) consisting of buses and trucks were 100%. Truck and Bus detection has a value of 100% for both of them, but the detection for each is not so good. False positives for the vehicle have a value of 0.5%. A small False Positive value shows excellent results and is suitable for this application. The True Positive value still needs to be improved again by retraining the weight value of YoloV3 COCO.
Table 2. False Positive and False Negative Test Results

| No. | Vehicle       | True Positive (%) | False Positive (%) |
|-----|---------------|-------------------|-------------------|
| 1.  | Motorbike     | ≈60 %             | ≈0.5 %           |
| 2.  | Car (LV)      | ≈93 %             | ≈0.5 %           |
| 3.  | Truck/Bus (HV)| ≈100 %            | ≈0.5 %           |

Fig. 6. Object detection test result

3.2. Processing Speed Test Results

Carrying out the detection process using YOLOv3 requires adequate computing devices. TOLOv3 is a method that uses deep learning to recognize objects. YOLOv3 which runs natively on the darknet platform. The new OpenCV series includes deep learning facilities from various platforms including TensorFlow and Darknet. This study using the test specifications as shown in Table 3 below. The hardware used is nVidia Jetson Nano. Tests were carried out by utilizing the CPU and GPU variations of the Jetson Nano.

Table 3. Testing Specifications

| No. | Specification          | Value                                      |
|-----|------------------------|--------------------------------------------|
| 1.  | Image dimension        | 640 x 300 pixel                            |
| 2.  | Yolov3 Size            | 320 x 320                                  |
| 3.  | Processing platform    | OpenCV DNN                                  |
| 4.  | Processor Jetson Nano (CPU) | Quad-core Arm A57 processor @ 1.43 GHz   |
| 5.  | Graphics Processing Unit (GPU) | 128-core Maxwell GPU                       |
| 6.  | Memory (RAM)           | System Memory – 4GB 64-bit LPDDR4 @ 25.6 GB/s |

The results of testing the processing speed for an image frame are shown in Table 4 and Figure 7. From the experimental results, the processing speed using the CPU is 0.5, and using the GPU reads 1 fps. From these results, it can be concluded that the processing speed with the specifications as mentioned, GPU usage processes two times faster than using the CPU.

Table 4. Processing Speed Test Results

| No. | Processing unit | Speed (fps) |
|-----|-----------------|-------------|
| 1.  | CPU             | 0.5         |
| 2.  | GPU             | 1           |
4. Conclusion

The conclusion of this research is the detection system using YOLOv3 with pretrained weight COCO gets a true positive value for motorcycles 60%, Cars (LV) 93%, and Trucks / Buses (HV) 100%. The processing speed of the nVidia Jetson Nano method with the OpenCV library on the GPU is 2 times faster than the process with the CPU. By using this method, there is insufficient FPS for real-time detection.

References

[1] de Souza A M, Brennand C A R L, Yokoyama R S, Donato E A, Madeira E R M and Villas L A 2017 Traffic management systems: A classification, review, challenges, and future perspectives Int. J. Distrib. Sens. Networks
[2] Fernández-Ares A, Mora A M, Arenas M G, Garcia-Sanchez P, Romero G, Rivas V, Castillo P A and Merelo J J 2017 Studying real traffic and mobility scenarios for a Smart City using a new monitoring and tracking system Futur. Gener. Comput. Syst.
[3] Yousef A, Shatnawi A and Latayfeh M 2019 Intelligent traffic light scheduling technique using calendar-based history information Futur. Gener. Comput. Syst.
[4] Finogeev A, Finogeev A, Fionova L, Lyapin A and Lychagin K A 2019 Intelligent monitoring system for smart road environment J. Ind. Inf. Integr.
[5] Rath M 2018 Smart Traffic Management System for Traffic Control using Automated Mechanical and Electronic Devices IOP Conference Series: Materials Science and Engineering
[6] Tubaishat M, Shang Y and Shi H 2007 Adaptive traffic light control with wireless sensor networks 2007 4th Annual IEEE Consumer Communications and Networking Conference, CCNC 2007
[7] Razavi M, Hamidkhani M and Sadeghi R 2019 Smart Traffic Light Scheduling in Smart City Using Image and Video Processing Proceedings of 3rd International Conference on Internet of Things and Applications, IoT 2019
[8] Wei H, Yao H, Zheng G and Li Z 2018 IntelliLight: A reinforcement learning approach for intelligent traffic light control Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining