Dynamic time distribution system monitoring on traffic light using image processing and convolutional neural network method

M Rifai¹, R A Budiman², I Sutrisno¹, A Khumaidi³, V Y P Ardhana¹, H Rosika⁴, Tibyani⁵, M Setiyono⁶, F Muhammad⁷, M Rusmin⁸ and A Fahrizal¹¹

¹Politeknik Penerbangan Surabaya, Indonesia
²³⁴Politeknik Perkapalan Negeri Surabaya, Indonesia
⁵⁶Universitas Qamarul Huda Badaruddin, Indonesia
⁷Brawijaya University
⁸⁹Politeknik Pelayaran Sorong, Indonesia
¹⁰¹¹Universitas Muhammadiyah Sorong, Indonesia

E-mail: rifai.moch.smart@gmail.com

Abstract. The number of vehicles that continues to increase every time can cause congestion and increase the potential for accidents. One type of violation that often results in accidents is motorists who break through traffic lights when the red light is on. One factor that encourage motorists to not obey traffic light signs is due to the distribution of green time which tends to be fast on relatively dense tracks. This final project aims to get dynamic green light on traffic light according to the volume of the vehicle for each lane using convolutional neural network method and can detect traffic light violations in the form of breaking the stop line markings on zebra cross. The sensor used is camera. There are three cameras used in each line. The number of cars detected will be sent to microcontroller via laptop. The data on the number of cars is used as input for the convolutional neural network to produce output data that serves as reference for obtaining the length of time for the green light. If the car stops above or exceeds the stop line on zebra cross when the red light is on, the car will be detected as violation.

1. Introduction
One of the effects of the increasing population is that the number of vehicles on the road is also increasing. Based on data from the Central Statistics Agency in 2015, 2016, 2017 in a row the number of motor vehicles was 121,394,185, 129,281,079, 138,556,699 vehicles. This is not proportional to the volume of roads which tend to be fixed, this results in traffic congestion which results in congestion. Congestion can interfere with the mobility of the community.

The amount of time lost in wasted due to congestion has an impact on people's income. Congestion cannot be avoided when the hours go to work or go to school. That also applies when the hours come home from work or school. Congestion is also affected by sub-optimal traffic light settings and traffic violations by motorists. Traffic violations are caused by motorists who do not obey traffic signs. One of the traffic violations is that the driver stops on the zebra crossing on the traffic light which is used by pedestrians to cross the road. The Bandung City Transportation Agency released a type of violation that was often violated by motorists in the period 15 to 27 November 2017, stopping at the zebra
crossing on the traffic light was the most frequently committed violation of 326 violations (Mawardi, 2017).

The ideal condition (time) of traffic lights distribution is important which can reduce the density of vehicles by adjusting the distribution which can adjust according to the level of congestion at each time. So it is necessary to have a system of tools that can detect violations and can adjust the state (time) of traffic lights according to the density level of the vehicle. Research on getting dynamic time has been done several times. The research that has been done is in the form of analysis and design simulation using a microcontroller and PLC. Based on the above background, the authors took the research to create a dynamic time distribution system that is suitable for congested traffic conditions and detects violations when the vehicle's position crosses the crossing markings when the traffic light is red.

The author makes a design using the Convolutional Neural Network (CNN) based image processing method, so that the results of the research conducted can be known. Therefore, the authors raised a research entitled "Dynamic Time Distribution System Monitoring on Traffic Light Using Image Processing and Convolutional Neural Network Method".

2. Relevant Study

2.1. Related Work

The research entitled "Design of Dynamic Time Distribution System for Traffic Light Using Image Processing Method Based on Fuzzy Logic" has studied calculating traffic density using camera sensors and can control the time at traffic lights. In this study, we cannot distinguish between left and right paths. The counted vehicle is the total number of vehicles on the road. (Hidayat, 2018)

Subsequent research entitled "Design and Development of Dynamic Time Distribution Systems and Detection of Violations in Traffic Light Using Fuzzy Logic Based Image Processing Methods has conducted research on calculating the density of traffic lanes and detection of zebra cross violations using camera sensors and can control time at traffic lights. In this study, it cannot detect objects other than cars. Vehicle detection is based solely on color and cannot detect smaller objects. (Saputro, 2019)

In this study, referring to previous research developed where researchers used the Image Processing method and the Convolutional Neural Network method. In contrast to previous studies, in this study the method used was Convolutional Neural Network. This convolutional neural network method is able to detect objects based on the entered datasheet. Image Processing is used to determine the number of vehicles on the left lane (Hidayat, 2018).

3. Research Design

3.1. System Design

This chapter contains the research flow. For the discussion that is about the identification of problems, study of literature, system design, hardware and software design, analysis and discussion. Systematically the steps in the research are made in the form of flowcharts as shown in Figure 1 below.
3.2. Design

Dataset is an object that represents data and relations that have been created in storage. A dataset can be called a database, because it has the same structure. The dataset in this study is taken from several examples of several images that have been stored in the laptop storage space. And the dataset is used for object classification. the more input the dataset will make the higher the classification results and the heavier the laptop runs.

As in Figure 2 above, some examples of dataset images and the number of datasets used. However, later the dataset used will increase even more.

![System Design Diagram](image-url)
### Dataset Table

| Dataset Name | Dataset Image | Total |
|--------------|---------------|-------|
| Stuck        | ![Stuck Image](image1.png) | 200   |
| Middle       | ![Middle Image](image2.png)  | 200   |
| Slack        | ![Slack Image](image3.png)   | 200   |

*Figure 2. Dataset.*

Hardware placement designs such as PC placement and camera sensors are designed as best as possible in order to get an accurate and efficient system response. As planned as shown in Figure 3 and 4 below.

*Figure 3. Hardware Design System.*
3.3. CNN Architecture Design

This final project is focused on getting an object detection system with a Mini PC. In addition, the main objective of this study is to obtain a classification result from the CNN process. In the general CNN process, this process has 3 stages, namely pre-processing, processing, and classifying.

The pre-processing process consists of 2 processes, namely the formation of inputs and conversion of inputs to grayscale. The second process is processing which consists of image convolution, image dimension reduction, max pooling, training with neural networks, and softmax. For the last process is classifying which has 1 process, namely determining output.

As in Figure 5 with the flow of the feature extraction process, namely by using the convolution process which is carried out 2 times, with the subsampling layer only done once. And for the stages, the first stage is the grayscale image which is processed through a convolution kernel or can be called a 5 x 5 Gaussian kernel with 10 convolution features to produce 10 new images of the same size without reducing pixels, 320 x 240 pixels, then the convolution process is repeated for the second time with a 5 x 5 kernel in this convolution the feature map does not increase because each feature map is only convolved once so that the second feature map is 316 x 236 x 10. The next process is subsampling and ReLu with half reduction. The pixel size of the previous image, therefore, on this layer the feature map will be 158 x 118 x 10. After the feature extraction process is done, flattening is carried out, namely making the 2D (mxn) matrix become 1D (mx 1), this flattening process produces an input vector of 186,440 who will then go through the training process with several similar output vectors the shape. For the image in 3.10 with a car input and there are 4 output vectors that resemble, namely truck, car, tv monitor, chair, motorcycle. The training is carried out with the help of softmax. And the output turned out to be the correct one is car.
3.4. Image Processing Design

In image processing, there is an initial process called pre-processing which is used to obtain information about the original image that is ready to be processed for the next stage, which can be called processing so that the result is obtained.

Figure 6 is a general sketch of the process of the CNN method. Before entering the CNN stages, the quantity and quality of the dataset must be determined first. Because it will affect the results of the training. The pre-processing stage is the process of obtaining image data which is then converted from an RGB image to Grayscale. Whereas at the Processing stage, an image convolution process occurs which functions to improve the image by means of a filter using the kernel convolution so that the im-
image looks more detailed by reducing noise. Then proceed with dimension reduction to change the image pixel size from the convolution process in half. After obtaining an image of the next reduction process, the result of the reduction is trained by the neural network to obtain the weight. This weight can determine and produce the Output value on the system.

4. Result and Discussion
4.1. Sensor and Actuator Testing
4.1.1. Camera Sensor Testing
To proceed to the next stage, a sensor test is required to be installed in this system. The following are the results of sensor testing in this Final Project plan. In this section, contains about camera testing. For checking the camera can be seen directly from the indicators contained on the camera. There is a green LED indicator located in the front area. This indicator will light up when the program asks the camera to turn on. In the application in the system, the camera has a little time to start up at the beginning. In 1, the following is an example of an indicator when the camera is on and off. And in Table 1 the following is the time it takes for the camera to turn on.

| No | Webcam On | Webcam Off |
|----|-----------|------------|
| 1  | ![Webcam On](image1.png) | ![Webcam Off](image2.png) |
| 2  | ![Webcam On](image3.png) | ![Webcam Off](image4.png) |
| 3  | ![Webcam On](image5.png) | ![Webcam Off](image6.png) |

Table 1. Webcam Condition.

The next step is testing the webcam connection to the PC. The PC will be given a program that will read and display the video in real-time. If successful, the test of the camera connection is complete. Figure 7 shows the results of testing the connection between the camera and PC.
Figure 7. Testing Webcam to PC.

4.1.2. Pilot Lamp Testing.
To proceed to the next stage, it is necessary to test the actuator that will be installed in this system. The following are the results of actuator testing in this Final Project plan. In this section, contains about testing the Pilot Lamp. To check the Pilot Lamp, it can be seen directly from the flame of the lamp. The following is an example of an indicator when the Pilot Lamp turns on and off. In Table 2 the following is an image when the Pilot Lamp is On, and an image when the pilot lamp turns off.

| No | Pilot Lamp On | Pilot Lamp Off |
|----|---------------|----------------|
| 1  | ![Image](image1) | ![Image](image2) |
| 2  | ![Image](image3) | ![Image](image4) |
| 3  | ![Image](image5) | ![Image](image6) |
4.2. Training Result

![Figure 8](image.png)

Figure 8. Dataset Training Result.

Figure 8 is the result of training using the aldadelta optimizer. From the graph, it shows that the results obtained using the aldadelta optimizer have an accuracy of up to 1 using 30 epochs.

4.3. Result Image Processing and CNN

| No | Figure Detected | Monitoring |
|----|----------------|------------|
| 1  | ![Image 1](image1.png) | ![Monitoring 1](monitoring1.png) |
| 2  | ![Image 2](image2.png) | ![Monitoring 2](monitoring2.png) |
| 3  | ![Image 3](image3.png) | ![Monitoring 3](monitoring3.png) |

Table 3. Test Result.
4.4. Hardware
The mechanical design in this tool is made using wood as a base, while the road image uses a banner with a size of 110 cm x 110 cm. The intersection used is a 4-lane intersection, one of which is a directional route. There are 3 traffic light poles and there is 1 camera on each traffic pole that is used to detect the number of vehicles in 3 traffic lanes. Figure 9 is the prototype used in this final project. The prototype is based on the mechanical design in chapter 3.

![Figure 9. Hardware system.](image)

5. Conclusion
The conclusion of this study that this Convolutional Neural Network method can monitor density levels well. The existing dataset already has a good level of accuracy, reaching 100%. This level of accuracy can be seen from the results of training data that have been carried out using the Adadelta optimizer.

6. References
[1] Khumaidi A 2017 Welding defect classification based on Convolution Neural Network (CNN) and Gaussian Kernal. *International Seminar on Intelligent Technology and Its Application* pp 261-265
[2] Deshpande A 2016 *A Beginner's Guide to Understanding Convolutional Neural Network Part 2*
[3] Suartika I W E 2016 Klasifikasi citra menggunakan Convolutional Neural Network (CNN) pada Caltech 101. *Jurnal Teknik ITS* 5 A65-A69
[4] Limantoro S, Kristian Y and Purwanto D 2017 *Deteksi Pengendara Sepeda Motor Menggunakan Deep Convolutional Neural Networks* (Teknologi Informasi Sekolah Tinggi Teknik Surabaya)
[5] Lutfi M 2018 *Pendeteksi Objek Api dan Boneka pada Robot Pemadam Api Berkaki Menggunakan Video Processing Berbasis Mini PC* (Politeknik Perkapalan Negeri Surabaya)
[6] Rappler.idntimes.com 2018 This Able Enterprise Memberdayakan Penyandang Disabilitas. (Available: https://rappler.idntimes.com/sakinah-haniy/thisable-enterprise-memberdayakan-penyandang-disabilitas)

[7] Sena S 2017 Pengenalan Deep Learning Part 7: Convolutional Neural Network. (Available: https://medium.com/@samuelensa/pengenalan-deep-learning-part-7-convolutional-neural-network-cnn-b003b477de94)

[8] Sidharta H 2017 Introduction to Open (CV Ilmu Komunikasi Binus Malang)

[9] Soebhakti H 2009 TPA81 Thermopile Array. AVR Application Note pp 1-6

[10] Stanford University 2016 An Introduction to Convolutional Neural (University Stanford).

[11] Jamii’in M A, Hu J, Marhaban M H, Sutrisno I and Mariun N B 2016 Quasi ARX neural network based adaptive predictive control for nonlinear systems. IEEJ Trans Elec Electron Eng 11 pp 83-90

[12] Jamii’in M A, Sutrisno I and Hu J 2012 Lyapunov learning algorithm for Quasi-ARX neural network to identification of nonlinear dynamical system. IEEE International Conference on Systems, Man, and Cybernetics (SMC) (Seoul, 2012) pp 3147-3152 doi: 10.1109/ICSMC.2012.6378275

[13] Jamii’in M A, Sutrisno I and Hu J 2013 Deep searching for parameter estimation of the linear time invariant (LTI) system by using Quasi-ARX neural network. The 2013 International Joint Conference on Neural Networks (IJCNN) (Dallas-TX, 2013) pp 1-5 doi: 10.1109/IJCNN.2013.6707091

[14] Jamii’in M A, Sutrisno I and Hu J 2015 Maximum power tracking control for a wind energy conversion system based on a quasi ARX neural network model. IEEJ Trans Elec Electron Eng 10 pp 368-375

[15] Jamii’in M A, Sutrisno I, Hu J, Mariun N B and Marhaban M H 2014 An adaptive predictive control based on a quasi-ARX neural network model. 13th International Conference on Control Automation Robotics & Vision (ICARCV) (Singapore, 2014) pp 253-258 doi: 10.1109/ICARCV.2014.7064314

[16] Sutrisno I, Jamii’in M A and Hu J 2014 Modified fuzzy adaptive controller applied to nonlinear systems modeled under quasi-ARX neural network. Artif Life Robotics 19 pp 22–26 https://doi.org/10.1007/s10015-013-0137-6

[17] Sutrisno I, Jamii’in M A and Hu J 2014 An improved elman neural network controller based on quasi ARX neural network for nonlinear systems. IEEJ Trans Elec Electron Eng 9 pp 494-501

[18] Sutrisno I, Che C and Hu J 2014 An improved adaptive switching control based on quasi-ARX neural network for nonlinear systems. Artif Life Robotics 19 pp 347–353

[19] Sutrisno I, Jamii’in M A and Hu J 2012 Neural predictive controller of nonlinear systems based on quasi-ARX neural network. 18th International Conference on Automation and Computing (ICAC) (Loughborough, 2012) pp 1-6

[20] Widodo R B et al. Grasping and attached mode in human computer interaction in the study of mouse substitution. Journal of Physics: Conference Series 1196 1 012034

[21] Sutrisno I, Jamii’in M A, Hu J and Marhaban M H 2015 Self-organizing quasi-linear ARX RBFN modeling for identification and control of nonlinear systems. 54th annual conference of the society of instrument and control engineers of Japan (SICE). IEEE

[22] Jamii’in M A, Sutrisno I and Hu J 2014 Nonlinear adaptive control for wind energy conversion systems based on quasi-ARX neural networks model. Proceedings of the International Multi-Conference of Engineers and Computer Scientists (March 12-14, 2014) (Hong Kong)

[23] Sutrisno I 2009 Pemrograman Komputer dengan Software Matlab (Surabaya: ITS Press)