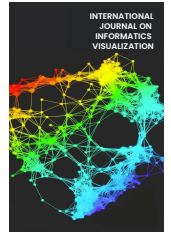




# INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION

journal homepage : [www.joiv.org/index.php/joiv](http://www.joiv.org/index.php/joiv)



## Autonomous Robot System Based on Room Nameplate Recognition Using YOLOv4 Method on Jetson Nano 2GB

Muhammad Pandu Dwi Cahyo<sup>a</sup>, Fitri Utamingrum<sup>b,\*</sup>

<sup>a</sup> Computer Engineering Study Program, Informatics Department, Faculty of Computer Science, Brawijaya University, Indonesia

<sup>b</sup> Computer Vision Research Groups Faculty of Computer Science, Brawijaya University, Indonesia

Corresponding author: \*f3\_ningrum@ub.ac.id

**Abstract**— The prediction of COVID-19 cases will continue to experience a surge, inseparable from the presence of a new variant of the coronavirus in the world. One of the best ways to prevent transmission of the virus is to avoid or limit contact with people showing symptoms of COVID-19 or any respiratory infection. The number of medical personnel infected when interacting with patients directly also needs to be an essential concern. Hence, an autonomous robot based on room nameplate recognition systems is a solution. It can be used as an intermediary medium for medical personnel with patients to reduce the intensity of direct contact primarily can be implemented in the hospital. It is expected to reduce the spread of the COVID-19 virus, especially among health workers. Each patient room in the hospital has its room nameplate to be used as a robot reference in navigating. This research aims to make a room nameplate recognition system using the YOLOv4 method on NVIDIA Jetson Nano 2GB that produces an output for 4-wheeled robot navigation control to move. This system is designed to detect rooms within a range of 1-3 meters using 5W and 10W power modes. The testing results based on recognition is obtained an average accuracy value of 95.34%. The system performance test results based on the power mode resulted in the best average computing time of 0.149 seconds. The average value of the accuracy of output integration with the system is 94.73%.

**Keywords**— Autonomous robot; recognition; nameplate; YOLOv4; NVIDIA Jetson Nano 2GB.

Manuscript received 5 Feb. 2022; revised 20 Feb. 2022; accepted 1 Mar. 2022. Date of publication 31 Mar. 2022.  
International Journal on Informatics Visualization is licensed under a Creative Commons Attribution-Share Alike 4.0 International License.



### I. INTRODUCTION

COVID-19 was first discovered in Wuhan (China), which subsequently migrated and spread worldwide. As a result of the massive spread of COVID-19, on March 11<sup>th</sup>, 2020, the World Health Organization (WHO) declared COVID-19 a global pandemic [1]. Globally, there are more than 218 million cases of COVID-19, with a death toll of 5,312,017 as of December 9<sup>th</sup>, 2021 [2]. Based on data from the world meters info site, 4,258,752 cases in Indonesia, with 4,109,675 patients recovered and 143,923 patients died [3].

One of the best ways to prevent transmission of the virus is to avoid or limit contact with people that show symptoms of COVID-19 or any respiratory infection [4]. As a health worker who deals directly with COVID-19 patients, this is very difficult to do, so to overcome the direct contact is to use robots as intermediary media [5]-[8]. Each patient room in the hospital must have a room nameplate. By utilizing a feature in the form of a pre-existing room nameplate, the room nameplate can be used as a robot reference in navigating.

Robots can function as liaisons for health workers when they want to give something to COVID-19 patients [9].

The previous research has proposed a method for the efficient detection of license plates [10]. The proposed license plate detection method uses three Faster-RCNN modules where each faster RCNN module uses a pre-trained CNN model, namely AlexNet, VGG16, and VGG19. Each Faster-RCNN module is trained independently, and their results are fused in the fusing layer. A publicly available dataset is used in experiments. The outcomes were outstanding where it can detect the exact location of license plates for 97 images. The accuracy of the proposed method is 97%. However, the system in this paper only detects through image input manually. It is not in real-time condition. The system also runs using a computer with qualified specifications, namely the Intel Xeon E5-1650 v4 CPU and 64 GB of memory. At the same time, when applied to robots, the main reason is portability, such as using a microcomputer like Jetson Nano.

A previous study investigated the performance analysis of the level of accuracy, precision, recall for heavy vehicle identification systems based on the YOLOv4 algorithm [11].

The crux of the model, an algorithmic computational mechanism, incorporates Mosaic Data augmentation and Transfer-learning techniques applied to avoid over-fitting and improve the optimal speed during training. A computer-vision algorithm is applied for testing a model in a real-time environment. This study compared the performance of the types of methods used, namely YOLO, YOLOv2, YOLOv3, YOLOv4, and Faster-RCNN. The highest test results were obtained by the YOLOv4 method, with an accuracy value of 96.54%.

Recognition image is researched in automation systems to detect and recognize logos on Google Street View [12]. This study proposes a framework consisting of a domain adaptation that is expected to reduce the loss function and represent important source features adopted by the target dataset. The study used several methods for testing, such as SSD, YOLOv2, YOLOv3, YOLOv4, and the process that the researcher proposed. Tests were carried out on the research dataset and the dataset from the public, namely BSVSO. The study's results using the BSVSO dataset showed that the YOLOv4 method got the second-highest average accuracy value of 68.55%. In contrast, when tested using the researcher's dataset, the YOLOv4 method got an average accuracy value of 92.89% when used for the detection and detection system.

In this research, we propose an autonomous system that can be implemented on a 4-wheel robot by utilizing a camera mounted on the front of the Jetson Nano-based robot, which was used to help detect and recognize the room's nameplate. The room nameplate identification process is carried out using the YOLOv4 algorithm because the resulting performance is excellent based on previous research applied to the detection system. The designed system can provide output in the form of navigation control for the robot to move forward and backward based on the plate recognition system output as a reference.

## II. MATERIAL AND METHOD

This study has five steps: data preparation, training of the YOLOv4 model, mechanical design, Graphical User Interface (GUI) design, and evaluation of the results. For the development of models, it is crucial to select the data augmentation technique, and the algorithms to be used in the model must also be chosen for the purpose. The model is created, followed by designing the robot hardware and the GUI system to run the previously developed model. After that, it is necessary to evaluate how accurate the model will be in actual situations. Fig. 1 shows a block diagram of the whole system.

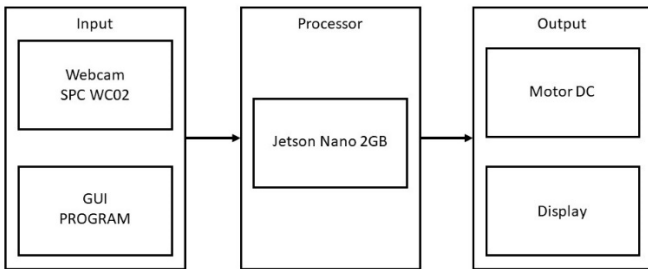


Fig. 1 Block Diagram of the Whole System

### A. Dataset

First, data preparation is preparing the dataset. It includes data augmentation and annotation [13]. In our research, we used our dataset by taking a photo of the room nameplate, which is made using laminated HVS paper with a size of 22cm x 10cm. All datasets were taken by positioning the nameplate on a small room (miniature) made using cardboard. The dataset consists of 5 classes, namely room A.1, room B.2, room C.3, room D.4, and BASE. Fig. 2 shows the example of each class dataset used in this study.



Fig. 2 Example of each class dataset

Roboflow framework [14] was used to shear, zoom, and mosaic data augmentation followed by changes in the brightness, color, and contrast using the Python program. These data augmentation techniques were applied to 80 raw images from each class resulting in a total dataset of 7250 images with 80% configuration for training data and 20% for validation data. In contrast, 50 images will be taken based on a predetermined distance for testing data. In the last step of preparing the data, we annotate all images according to the class using third-party tools called LabelImg [15]. Fig. 3 shows the flow process of data preparation used in this study.

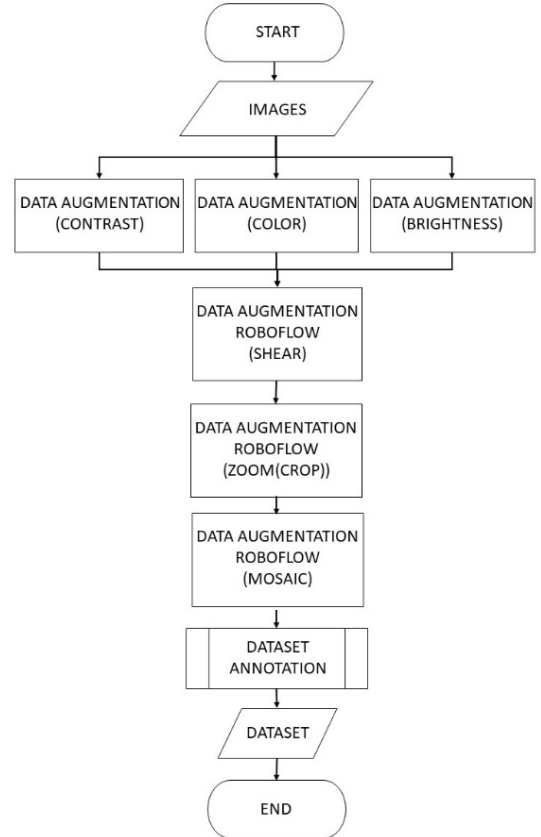


Fig. 3 Flow Chart of Data Preparation

### B. YOLOv4 Models

In this work, we investigate the suitability of the YOLOv4 algorithms for room nameplate recognition systems in autonomous robot environments. Fig. 4 shows the architecture of YOLOv4 using Darknet in general.

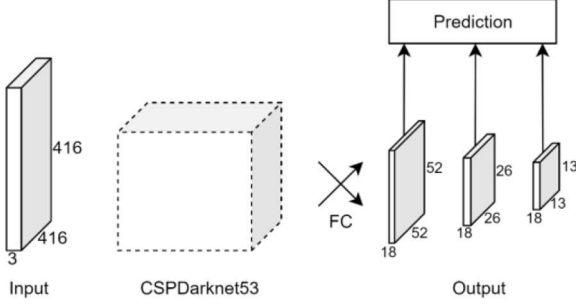


Fig. 4 YOLOv4 Architecture

This architecture uses the CSPDarknet53 CNN backbone architecture. This architecture contains 162 layers. Input is an image that has been resized according to the size set in the configuration. Next, the input layer is incorporated into the CSPDarknet53 backbone architecture. On the outside, the output is generated from the convolution process from the output of the CSPDarknet53 backbone architecture.

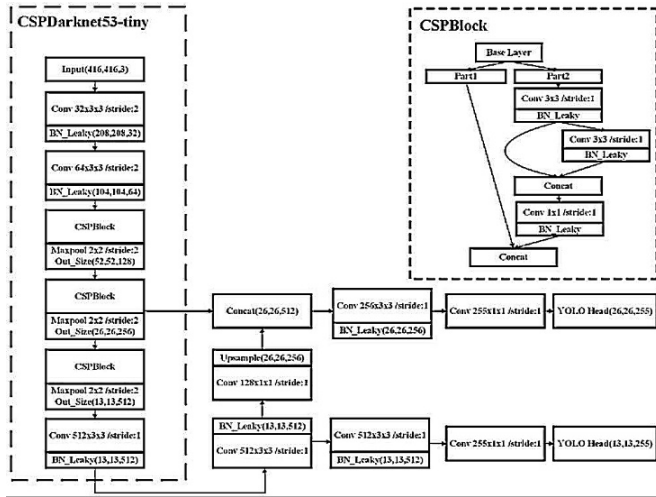


Fig. 5 CSPDarknet53-tiny structure

In this study, we used the CSPDarknet53-tiny architecture or a small version of CSPDarknet53 used in YOLOv4. Fig. 5 shows the layer structure of the CSPDarknet53-tiny. CSPDarknet53-tiny uses the CSPBlock module to divide it into two feature maps and then combine them. The CSPBlock module can improve the learning ability of the convolution network and improve computations which causes an increase in the accuracy of the YOLOv4-tiny method. The use of two different feature maps, namely 13x13 and 26x26 predicts the detection results [16]. In this study, YOLOv4-tiny was used to design a room nameplate recognition system. YOLOv4-tiny is designed based on YOLOv4 to increase detection speed. YOLOv4-tiny is very suitable for detection on small or mobile devices. Predictions made by YOLOv4-tiny are the same as YOLOv4 [16].

To train the model using the Darknet framework, we must set the configuration file first. Default by Darknet has

provided the configuration file with the extension .cfg YOLO to create a new configuration only by changing the value in this configuration file. This YOLOv4 configuration file contains the parameters used for the process during training and the structure of the YOLOv4 architecture. The parameter values that were changed in this configuration are shown in Table 1.

TABLE I  
PARAMETER VALUES IN THE CONFIGURATION FILE

Variable	Value
Width	416
Height	416
Max_batches	10000
Steps	8000, 9000
Filters	30
Classes	5

Furthermore, we can proceed with the training process. The inference process of the YOLOv4 network model uses NVIDIA Jetson Nano 2GB. The YOLO weights were converted to the TensorRT format. YOLOv4 uses an activation function called "Mish" while TensorRT does not support the activation function. So we need an activation function that can replace the "Mish" function using equation 1 [17].

$$f(x) = x * \tanh(\text{softplus}(x)) \quad (1)$$

where  $\text{softplus}(x) = \ln(1 + e^x)$ .

The activation function "softplus" already supports the TensorRT model format and to convert from the YOLOv4 weight to the TensorRT model, a connecting model is needed, namely ONNX, so the first process is to change it in the ONNX format before being converted back into the TensorRT model format for the inference process from the YOLOv4 model to NVIDIA Jetson Nano 2GB.

### C. Mechanical Design

Mechanical design is the construction design and arrangement of mechanical components used in building a prototype tool [18]. In this study, researchers used a 4-wheeled robot with 4 DC motors, powerbank, battery, webcam, USB WiFi, jumper cable, L298N motor driver, switch, and a microcomputer NVIDIA Jetson Nano 2GB. Fig. 6 shows a wiring schematic, and Fig. 7 shows the visualization of hardware design.

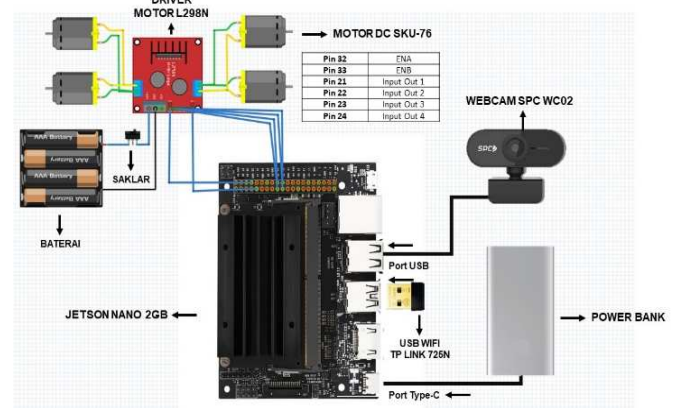


Fig. 6 Hardware Wiring Schematic

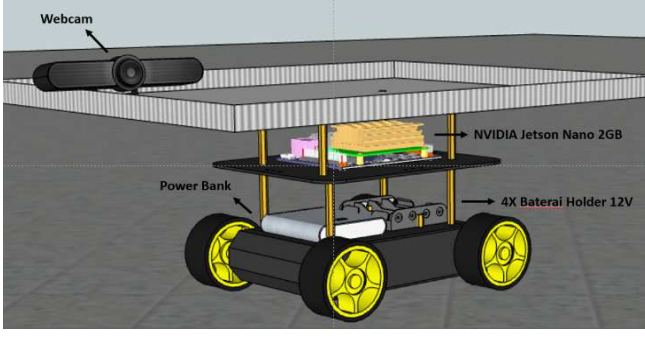


Fig. 7 Visualization of Hardware Design

On the 4-wheel robot board, a partition divides the robot into three parts at the bottom to install 4 DC motors connected to the L298N motor driver. In the middle, there is a power bank as a power source for the NVIDIA Jetson Nano 2GB and a battery as a power source for the DC motor and L298N motor driver. At the top, there is an NVIDIA Jetson Nano 2GB which is already connected to USB WiFi for VNC connection and a webcam camera. L298N motor driver port configuration connected to DC motor is shown in Table 2, and GPIO pin configuration on NVIDIA Jetson Nano 2GB connected to L298N motor driver are shown in Table 3.

TABLE II  
L298N MOTOR DRIVER PORT CONFIGURATION

DC Motor	L298N Motor Driver
Positive Pin on DC Motor 1 & 2	Pin OUT 1
Negative Pin on DC Motor 1 & 2	Pin OUT 2
Positive Pin on DC Motor 3 & 4	Pin OUT 3
Negative Pin on DC Motor 3 & 4	Pin OUT 4

TABLE III  
GPIO NVIDIA JETSON NANO 2GB PIN CONFIGURATION

NVIDIA Jetson Nano 2GB	L298N Motor Driver
Pin 32	ENA
Pin 33	ENB
Pin 21	IN1
Pin 22	IN2
Pin 23	IN3
Pin 24	IN4

The operating time that this system can work autonomously can be calculated using equation 2 [19] and equation 3 [20], where  $I$  = current,  $P$  = power, and  $V$  = voltage.

$$I = \frac{P}{V} \quad (2)$$

$$\text{Operating Time} = \frac{\text{Battery Capacity (in mAh)}}{\text{Load Current (in mA)}} \quad (3)$$

Based on the use of a powerbank in this research with a capacity of 10000mah with an output voltage of 5V, we can calculate in order to use the 5W power mode, the current is 1A (1000 mA), while for the 10W power mode the current is 2A (2000 mA). Based on that, an approximation operating time is about 10 hours for 5W power modes. Meanwhile, an approximation operating time for 10W power modes reaches 5 hours.

#### D. Graphical User Interface (GUI) Design

There are many types of user interfaces. A graphical user interface is a type of user interface that allows people to

interact with programs more than typing, such as computers [21].

In this work, the system was implemented in Python. We used a module library named "Tkinter" to create GUI programs. The GUI is designed to receive input in a nameplate target desired by the user. Each button in the GUI represents each class on the system, representing a function that will return each class name. The function will call the main program using the function's parameters of the value returned. Furthermore, when the main program function is executed successfully, the window will be terminated, and the system GUI will return to the idle position. Fig. 8 shows the visualization of the GUI program used in this system, and Fig. 9 shows the main program algorithm flowchart.

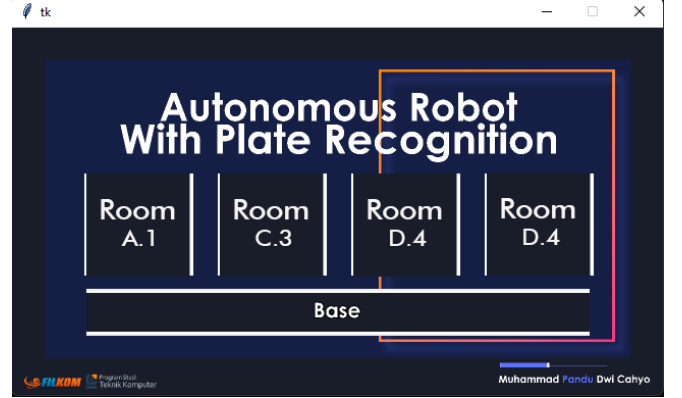


Fig. 8 Visualization of GUI Design

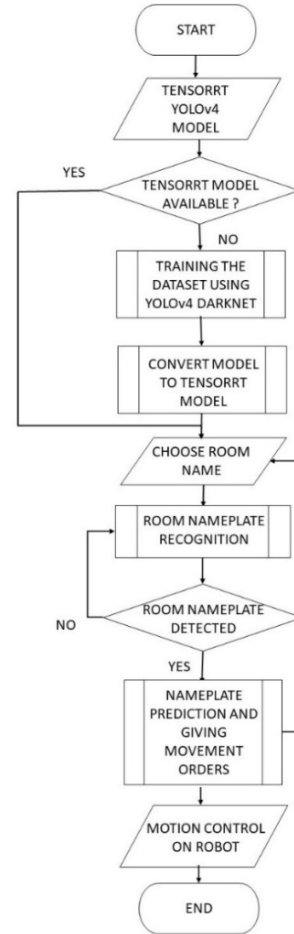


Fig. 9 Main program algorithm flowchart



### E. Evaluation

Accuracy is a quickly comprehensible paradigm that represents the ratios of examples that a classifier correctly recognizes the class of testing examples. To measure the accuracy of a classifier is also an easy way, and calculate according to the formula given in equation 4 [22], where TP denotes True Positive detection, the true positive is taken from the classification with the correct detection, and N denotes the number of data testing.

$$\text{Accuracy} = \text{TP} / \text{N} * 100 \quad (4)$$

### III. RESULTS AND DISCUSSION

This research initially compared the effect of confidence values and distance values on system accuracy. The test is intended to obtain the optimal value to be tested on the integration test of the room nameplate recognition system for motion control in autonomous robots. The results of confidence values test are shown in Table IV.

TABLE IV  
AVERAGE ACCURACY OF CONFIDENCE VALUES TEST RESULTS

Confidence Values	Detection Result			Accuracy (%)
	Correct Detected	Not Detected	False Detected	
0.6	47	0	3	94
0.7	46	2	2	92
0.8	46	3	1	92
0.9	44	5	1	88

We used test data taken at 3 meters for uniformity in this test. The highest accuracy value is when using a confidence value of 0.6, which gets an accuracy value of 94%. While the second-highest value was obtained when using a confidence values of 0.7 and 0.8, which is 92%. The lowest value is when using a confidence value of 0.9, which is 88%. Although the use of a confidence value of 0.6 gets the best accuracy value, the error rate obtained is the highest, reaching 6%. The use of a confidence value of 0.7 and 0.8 gets the same accuracy value. Still, the error rate of detection using a confidence value of 0.7 is more significant at 4% than only 2% when using a confidence value of 0.8. Because this study is intended to be an intermediary between health workers and patients so that the lowest possible error rate is needed, it can be concluded that the best confidence value used in this study is 0.8.

The results of the distance values test are shown in Table V. The results came out a success, given that the average value of the accuracy of the whole system is 95.34%.

TABLE V  
AVERAGE ACCURACY OF DISTANCE VALUES TEST RESULTS

Distance (meter)	Detection Result			Accuracy (%)
	Correct Detected	Not Detected	False Detected	
1	48	1	1	96
2	49	0	1	98
3	46	3	1	92
Overall Accuracy				95,34

Tests were carried out at different distances. The factors that influence the good and bad of the test are the lighting and the angle of the nameplate on the picture frame.

The highest accuracy value based on the distance is at a distance of 2 meters, which is 98%. Meanwhile, the lowest value at a distance of 3 meters is 92%. The distance of 2 meters is the distance with the highest accuracy value because, at a distance of 2 meters, the size of the room nameplate on the picture frame is clearly and ideally visible. So it can be concluded that the best distance is 2 meters. A distance of 3 meters has the lowest accuracy value due to limitations in hardware, namely the quality of the webcam camera, where the webcam is not designed to capture images from a long distance so when shooting at a distance of 3 meters, the resulting image looks blurry. Fig. 10 shows when the system detects the room nameplate correctly.

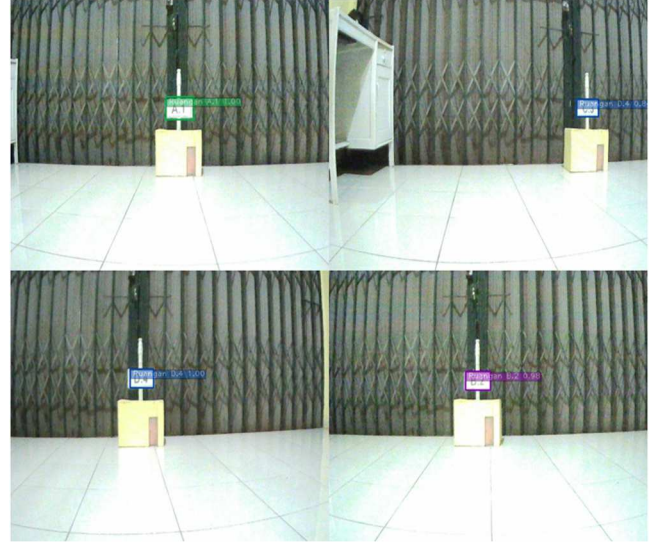


Fig. 10 Sample test results

The next test analyzes the effect of power mode on the NVIDIA Jetson Nano 2GB on frame per second (FPS) performance and computing time in running the room nameplate recognition system using the YOLOv4 method. The results of the power mode test are shown in Table VI. Fig. 11 shows the graph of computing time performance, and Fig. 12 shows the graph of FPS performance.

TABLE VI  
PERFORMANCE OF POWER MODE TEST RESULTS

Power Mode	FPS	Compute Time (s)
5 Watt	13,948	0,213
10 Watt	25,692	0,149

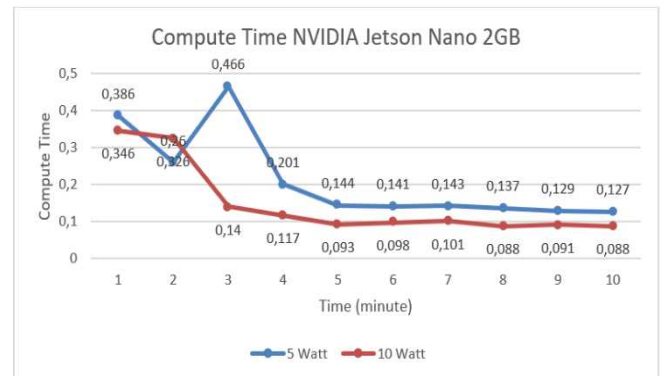


Fig. 11 Graph of computing time performance

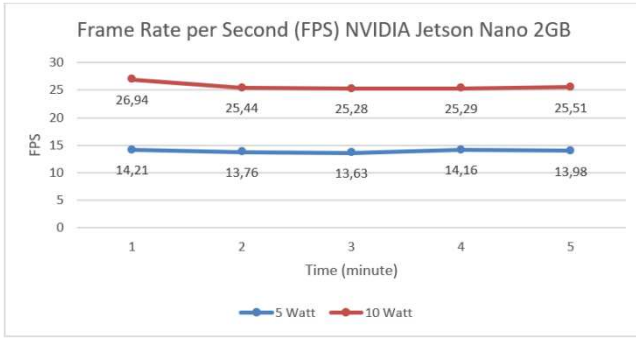


Fig. 12 Graph of FPS performance

The test shows a difference in performance when using different power modes. The best results are obtained using 10 W power mode, which produces an average FPS of 25.692 with a computation time of 0.149 seconds to process an image frame. Based on Fig. 11, it can be concluded that the computational time performance in the 4th image process will experience an acceleration of computing time in both power modes until it tends to stabilize. Long computing time in both power modes when processing the first image to the third image because the system tends to experience a decrease in performance due to the system just initializing many variables and calling a lot of functions to run the program. When the system is under control, the computational time will be stable at a specific value in both power modes. While in Fig. 12, it can be seen that the FPS performance produced by the NVIDIA Jetson Nano 2GB in running the room nameplate recognition program using the YOLOv4 algorithm on the TensorRT model makes stable performance for 5W and 10W power modes.

After getting the most optimal confidence value, distance, and power mode. The next step is to use all these variables to test the system integration on the autonomous robot's output in motion control/navigation. The results of the integration test are shown in Table VII.

TABLE VII  
AVERAGE ACCURACY OF INTEGRATION SYSTEM TEST RESULTS

Start Position	Target Position	Actual Position	Result
BASE	BASE	BASE	Correct
BASE	Room A.1	Room A.1	Correct
BASE	Room B.2	Room B.2	Correct
BASE	Room C.3	Room C.3	Correct
BASE	Room D.4	Room D.4	Correct
Room A.1	BASE	BASE	Correct
Room A.1	Room A.1	Room A.1	Correct
Room A.1	Room B.2	Room B.2	Correct
Room A.1	Room C.3	Room C.3	Correct
Room A.1	Room D.4	Room D.4	Correct
Room B.2	BASE	BASE	Correct
Room B.2	Room B.2	Room B.2	Correct
Room B.2	Room C.3	Room C.3	Correct
Room B.2	Room D.4	Room D.4	Correct
Room C.3	BASE	Room B.2	Incorrect
Room C.3	Room C.3	Room C.3	Correct
Room C.3	Room D.4	Room D.4	Correct
Room D.4	BASE	BASE	Correct
Room D.4	Room D.4	Room D.4	Correct
Accuracy of Integration System (%)			94,73

The robot can only move backward when it chooses to return to BASE, while for other rooms, it can only move sequentially in the sequence BASE - Room A.1 - Room B.2 - Room C.3 - Room D.4. If the robot is in room C.3 and wants to go to room A.1, it must first move back to BASE before moving to room A.1. According to Table VII, it can be seen that there is 1 test that is not accurate because the robot stops moving in room B.2 while the target room that the robot wants to go to is BASE. This happens because the robot detects the nameplate of the B.2 room as a BASE class. The accuracy of system integration to hardware obtained is 94.73% by testing every possible combination of movements that the robot can carry out.

#### IV. CONCLUSION

In the current work, we have developed a room nameplate recognition system using the YOLOv4 algorithm based on computer vision. We mounted a camera placed in front of the 4-wheeled robot is aimed toward the room nameplate. Later, the captured image will be detected using the YOLOv4 method to recognize the class of the room nameplate. When the system recognizes the object matching with the target, the system will control the motion of the robot to move forward, backward, or stop according to the object's class. In this study, we tested the effect of confidence and distance values on the average accuracy of the room nameplate recognition system. Testing the impact of power mode uses the NVIDIA Jetson Nano 2GB on FPS performance and computing time in running the system. The test shows that using a confidence value of 0.8, a distance of 2 meters, and a power mode of 10W will produce the best performance. The value of this variable will be used in testing the system integration. The comprehensive test on integrating the room nameplate recognition system for controlling the DC motor obtained an accuracy value of 94.73%. Based on the results of research and discussion, it can be concluded from this study that the room nameplate recognition using the YOLOv4 method can be implemented into the autonomous robot on Jetson Nano 2GB.

In addition, we suggest adding a speaker or buzzer as the resulting output so that there is a sign if the robot has reached the target and adding variations in robot motion in navigating and applying the system to other types of robots.

#### ACKNOWLEDGMENT

We thank the Faculty of Computer Science, University of Brawijaya, for giving us the opportunity and support to carry out this research.

#### REFERENCES

- [1] WHO, (2020). Retrieved from WHO Director-General's opening remarks at the media briefing on COVID-19 - March 11th 2020. [online] Available at: <https://www.who.int/dg/speeches/detail/who-director-general-s-opening-remarks-at-the-media-briefing-on-covid-19---11-march-2020>.
- [2] WHO Coronavirus (COVID-19) Dashboard, (2022). [online] Available at: <https://covid19.who.int/>.
- [3] Worldometers, (2022). COVID-19 Coronavirus Pandemic. [online] Available : <https://www.worldometers.info/coronavirus/>.
- [4] J. Howard, A. Huang, Z. Li, Z. Tufekci, V. Zdimal, H.V.D. Westhuizen, A.V Delftm A. Price, L. Fridman, L.H. Tang, V. Tang, G.L. Watson, C.E. Bax, R. Shaikh, F. Questier, D. Hernandez, L.F Chu, C.M. Ramirez, A.W. Rimoin. "An evidence review of face masks

- against COVID-19", PNAS 2021 Vol. 118 No. 4 e2014564118. <https://doi.org/10.1073/pnas.2014564118>
- [5] Bogue R. Robots in a contagious world. *Ind Rob* 2020;47:642–73. <https://doi.org/10.1108/IR-05-2020-0101>.
- [6] Yang G-Z, Nelson BJ, Murphy RR, Choset H, Christensen H, Collins SH, et al. Combating COVID-19-The role of robotics in managing public health and infectious diseases. *Sci Robot* 2020;5. <https://doi.org/10.1126/scirobotics.abb5589>.
- [7] Khaleghi A, Mohammadi MR, Jahromi GP, Zarafshan H. New ways to manage pandemics: using technologies in the era of COVID-19, a narrative review. *Iran J Psychiatry* 2020;15:236–42.
- [8] Sharma D, Nawab AZB, Alam M. Integrating M-Health with IoMT to counter COVID-19. *Stud Comput Intell* 2021;923:373–96. [https://doi.org/10.1007/978-981-15-8534-0\\_20](https://doi.org/10.1007/978-981-15-8534-0_20).
- [9] Khan ZH, Siddique A, Lee CW. Robotics utilization for healthcare digitization in global COVID-19 management. *Int J Environ Res Public Health* 2020;17. <https://doi.org/10.3390/ijerph17113819>.
- [10] Omar, N., Abdulazeez. A.M., Sengur, A., Al-Ali, S.G.S. (2020). Fused faster RCNNs for efficient detection of the license plates. In *Indonesian Journal of Electrical Engineering and Computer Science (IJECS)* Vol. 19, No. 2, August 2020, pp. 974–982.
- [11] Sowmya, V., Radha, R. (2021). Heavy-Vehicle Detection Based on YOLOv4 featuring Data Augmentation and Transfer-Learning Techniques. *J. Phys.: Conf. Ser.* 1911 012029.
- [12] Yohannes, E., Shih, T.K., Lin, C.Y., Enkhbat, A., Hong, C.Y., Utaminingrum, F.. (2021). Domain Adaptation Deep Attention Network for Automatic Logo Detection and Recognition in Google Street View. *IEEE Access*, 10.1109/ACCESS.2021.3098713.
- [13] Y.L.X, Wang, Z. Maio, W.C, Tan. Data Augmentation for ML-driven Data Preparation and Integration. *VLDB Endowment*, Vol. 14, No. 12 ISSN 2150-8097. doi:10.14778/3476311.3476403
- [14] Roboflow, 2021. Roboflow Annotate. [online] Available at: <https://roboflow.com/annotate>.
- [15] Tzatalin. LabelImg. Git code (2015). [online] Available at: <https://github.com/tzatalin/labelImg>.
- [16] Jiang, Z., Zhao, L., Li, S., Jia, Y., 2020. Real-time object detection method based on improved YOLOv4-tiny.
- [17] D. Misra. (2020). Mish: A Self Regularized Non-Monotonic Activation Function. arXiv:1908.08681v3.
- [18] J., Nasir, A.A. Ramli, Michael. (2019). Design of Door Security System Based on Face Recognition with Arduino. *JOIV* Vol.3, No. 2 e-ISSN : 2549-9904, ISSN : 2549-9610.
- [19] All About Circuits. Ohm's Law - How Voltage, Current, and Resistance Relate. [online] Available at: <https://www.allaboutcircuits.com/textbook/direct-current/chpt-2/voltage-current-resistance-relate>.
- [20] Digikey. Battery Life Calculator. [online] Available at: <https://www.digikey.com/en/resources/conversion-calculators/conversion-calculator-battery-life>.
- [21] H.N.M. Shah, M.Z.A. Rashid, Z. Kamis, M.S.N. Aras, N.M. Ali, F. Wasbari, M.N.F.B.A. Bakar. (2018). Design and Develop an Autonomous UAV Airship for Indoor Surveillance and Monitoring Applications. *JOIV* Vol.2, No. 1, e-ISSN : 2549-9904, ISSN : 2549-9610.
- [22] A.A. Fauzi, F. Utaminingrum, F. Ramdani. (2020). Road surface classification based on LBP and GLCM features using kNN classifier. *Bulletin of Electrical Engineering and Informatics*, Vol. 9, No. 4, pp. 1446–1453