

Directional Sign Recognition for Raspberry Pi-Based Automatic Guided Vehicle Navigation

Florentinus Budi Setiawan¹, Rachmat Hidayat², Leonardus Heru Pratomo³, Slamet Riyadi⁴

^{1,2,3,4} Program Studi Teknik Elektro, Fakultas Teknik Universitas Katolik Soegijapranata, Jl. Pawiyatan Luhur Selatan IV No. 1 Semarang 50234 INDONESIA (tel.: 024-8441555; email: f.budi.s@unika.ac.id, rachmatha2508@gmail.com, tu.elektro@unika.ac.id)

[Received: 4 July 2022, Revised: 28 October 2022]
Corresponding Author: Florentinus Budi Setiawan

ABSTRACT — The development of modern times in robotics and mechanization technology has increased significantly in the past few decades due to their high efficiency in time and energy. In the goods mobilization system for companies' use, particularly the industrial and warehousing divisions, one of the robots that are used for transporting goods is an automatic guided vehicle (AGV). One of the old navigation methods in AGV is the use of a sensor to follow the line pattern on the detected object, namely the line on the floor. This method is rather ineffective because, gradually, these line pattern objects on the floor will fade caused by the effect of AGV wheels' frictional forces, causing the camera sensor can no longer detect them. Therefore, it is necessary to improve the AGV navigation method so that it can be a sustainable innovation. This navigation method used four image objects positioned in the area traversed by the AGV robot and the camera served as a forward-facing sensor so that the AGV could detect the pattern of image objects with the help of computer vision using the OpenCV software library. The pattern of the detected image object was processed by a program designed on the Raspberry Pi 4 Model B minicomputer. The test results prove that this method can detect image objects within the camera's field of view and successfully display the output of the image object. The system managed to recognize objects quite accurately, with parameters of 10–95 cm, and through several experiments. The analysis of the rotational speed of the front and rear wheels of the AGV was carried out using an oscilloscope and tachometer as a means of measuring wheel speed or rotation.

KEYWORDS — Robotics, AGV, Computer Vision, OpenCV, Raspberry Pi.

I. INTRODUCTION

In the automatic delivery system [1], the automatic guided vehicle (AGV) is an essential part. The AGV is an autonomous delivery vehicle that can perform certain movements by following a program and a guide route, such as moving forward, halting, and turning, as well as stopping at a programmed workstation. This vehicle is utilized to load or unload goods [2]. The AGV can be classified into two categories. The first category is the AGV with a guide path equipped with a reflective infrared transmitter and receiver to track the guide path, while the second is the AGV without a guide path driven by remote control with radio waves [3].

The AGV can utilize marks or patterns on the floor to navigate itself in certain environments [4]. This device can also make use of a magnet or laser as its vision for navigation methods, depending on the application. This method is frequently used in industrial sectors to convey materials or goods from one place to another at manufacturing facilities or warehouses [5]. The AGV can help increase productivity and reduce costs as it facilitates the factory or warehouse with automation. It has a wide range of applications, most of which are used to transport various materials, including pallets, rolls, racks, carts, and containers [6].

The industry has increased the use of robots for transportation and material processing [7]. This robotic system employs many innovative sensing technologies and control techniques to increase its versatility. This transport system relies on the accuracy and repeatability of the AGV that is used to convey materials and equipment around the workspace. The keys to achieving the requisite accuracy are precise and structured AGV calibration. Diverse approaches to AGV navigation have been extensively discussed in the literatures. Navigation approaches can be separated into two broad categories: those that perform calculations manually and those that rely on external positioning references. Manual

calculations primarily rely on accurate estimation of the AGV's travel distance and orientation [8].

A poorly calibrated and unregulated AGV can lead to serious positioning errors. Minor errors in the commanded rudder and the measured distance will accumulate, resulting in significant and imprecise positioning errors [9]. For an AGV that relies on an external navigation control setting resource, the ability to accurately direct the servo to the desired location may be affected by the quality of the vehicle calibration. This study presents a computer vision algorithm using a mathematical model for image processing [10] to obtain information from images or data.

Human vision in the presence of several different images can be directly analyzed according to their understanding. However, using computers for image analysis will result in more interpretation methods, that are overall more complex. Image data types were therefore used for object analysis. An image interpreted through a physical probability model and, finally, a computer vision algorithm can be formed. This study is a refinement of the detection system of directional signs [11]. The computer vision used in this study displayed output on serial monitors, or it could be said that only the results of the directional signs reading were shown. Hence, this study implemented directional sign readings using an autonomous vehicle which can be referred to as AGV [11]. During the implementation process of this method, computer vision algorithms could flexibly identify images accurately by utilizing four images used for navigation. The use of the region of interest (ROI) method in image processing can improve the obtained identification accuracy better than other methods. This paper is composed of four sections: Section I is the introduction, while Section II discusses the methodology. Section III presents the results and discussion, and Section IV contains conclusions.

II. PROPOSED METHODS

The method used in this research was the waterfall method. This method was used as a reference to ensure the success of the research project [12]. In the waterfall method, the entire device development process was divided into separate yet complementary phases. This model is the result of one phase operating as an input for the next phase in sequence. The most challenging part of any scientific investigation is not solving a particular problem but rather formulating questions that focus on essential aspects of the phenomenon under investigation [13]. The research stages of building AGV robots are shown in Figure 1.

A. DESIGN OF THE AGV MODEL

The initial step is to design the AGV model. As depicted in Figure 2, this device was manufactured through the 3D design process. This study discusses the design of a direction identifier employing computer vision as the AGV navigation [14]. This device utilized a Raspberry Pi 4 Model B minicomputer as its brain [15]. This minicomputer served as a central computer that commanded the AGV's movement on a predetermined path. Image processing was also carried out using a Raspberry Pi 4 model B minicomputer. Then, a high-torque MG996R digital servo motor was employed for the front-wheel steering [16]. The AGV sensor was the Raspberry Pi Camera V2, which is the new official camera board released by the Raspberry Pi. Meanwhile, for the acceleration on the front and rear wheels, the L928N driver was used as a DC motor driver module [17] and the rotational speed controller of that DC motor [18]. The desktop computer was used as a viewer or screen mirror when the AGV was operated. A rechargeable LiPo battery was installed as the AGV's power source, while the AGV chassis used a small 4WD remote control car chassis and an acrylic board as support for the AGV robot.

B. AGV SOFTWARE SETUP

The AGV design was carried out using 3D SketchUp software [19]. The algorithm was designed and implemented using Python 3.9.1 [20], while the computer vision library OpenCV [21] was used for image processing commands. After that, the AGV speed control and steering control algorithms were implemented in Python [22], while the Raspberry Pi 4 model B viewer board was run on a laptop using VNC Viewer software [23].

C. CONVERSION OF THE RGB COLOR VALUE TO THE HSV VALUE

The image caught by the camera is usually in the red, green, and blue (RGB) color space. This color space needs to be converted to another color space. The most popular approach in color segmentation techniques is the hue, saturation, and value (HSV) color model [24]. The RGB color model is mainly used in computer graphics applications. This color model is generally built on a Cartesian coordinate space that represents the x, y, and z axes. On the other hand, the HSV color model is built on the representation of hue, saturation, and value. Hue represents the original color itself; saturation indicates the purity of the color associated with the standard deviation around the dominant wavelength; meanwhile, the value represents the brightness of the color. In this case, it describes the lightness or darkness of the color ranging from 0 (dark) to 1 (light). Value is related to the amount of white [25].

In addition to the color-based technique, the shape-based technique was also used in the object detection stage. This

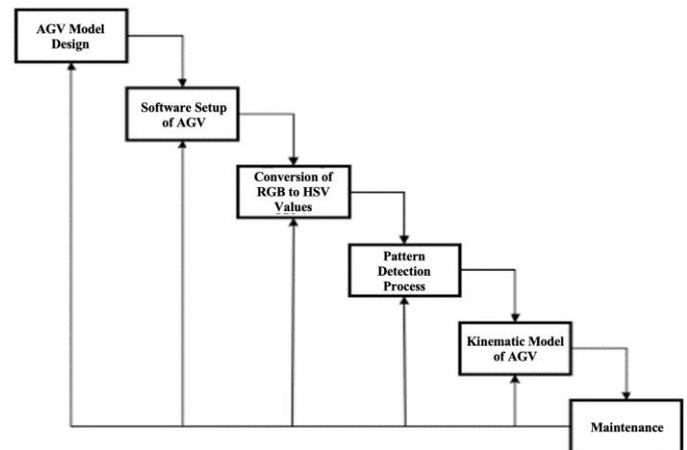


Figure 1. Research flowchart.

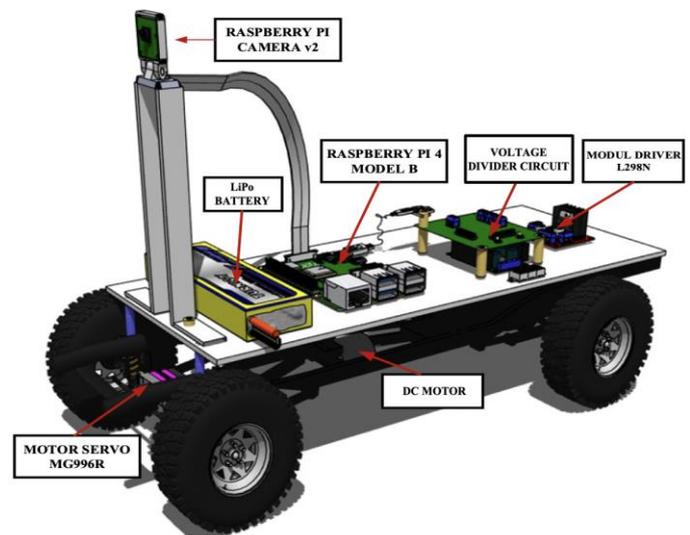


Figure 2. Design of the AGV 3D prototype from the side view.

detection technique utilized the common object in context (COCO) dataset. In fact, the shape-based technique is more challenging to implement due to the diverse shapes and appearances of detected signs. It occurs due to the difficulty in detecting images when they are too small and blurry as a result of the camera capturing them from afar under different conditions and lighting angles [26]. The algorithms using the HSV and RGB colors are explained in this section. A pixel is marked according to the HSV range. The boundaries for the HSV color space algorithm are described in (1).

$$(P_{hue_low} \geq 0,06) \text{ atau } (\leq 1) \text{ and } (P_{saturation} \geq 0,75) \text{ and } (P_{value} \geq 0,65). \quad (1)$$

Equation (1) shows that P_{hue_low} and P_{hue_high} represent the value of hue, $P_{saturation}$ represents the value of saturation, and P_{value} represents the value of value. On the contrary, color segmentation using the RGB algorithm utilizes different color boundaries. Besides, the achromatic composition was used to determine the ROI of candidates of directional sign images. The achromatic decomposition equation is shown in (2).

$$value(R, G, B) = \frac{(|B-R|+|G-B|+|R-G|)}{3E}. \quad (2)$$

In (2), values of $R, G,$ and B represent the brightness of the shadow, while E is the total value of the extracted achromatic color. Achromatic defines the brightness or darkness of a color in a black-and-white image. In its implementation, the value of

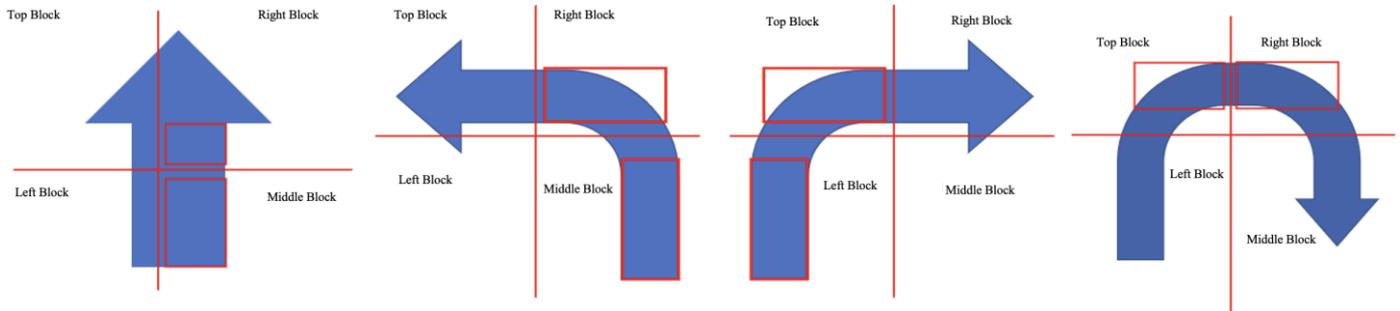


Figure 3. ROI marking.

E was set to 20. This value was selected as it yielded the best result in the RGB color segmentation. Based on the proposed algorithm, the image was converted into a binary image to separate the ROI area [27].

If the algorithm discovered pixels belonging to the direction pattern, they would be converted to white pixels, while black pixels represented the background. Then, the white pixels would be segmented using a bounding box, and the binary morphological process would eliminate the unwanted noise and combine some gaps that may arise due to an incomplete thresholding process [28].

D. PROCESS OF PROGRAM STAGE

In building this device, the program was divided into several subprograms: the library call, camera initialization, directional sign definition, directional sign identification, and AGV movement. The library call contained a list of libraries used in programming. If the library were not called, the program would not function properly. The camera initialization was used to set the camera’s resolution, brightness, rotation direction, and frame rate. The directional sign definition contained some preprocessing used to define the to-be-read images. In the directional sign identification, the direction reading was performed using the ROI method, while the AGV movement involved the servo and DC motors’ mechanics. These two motors were integrated according to the intended pattern. These subprograms were assembled into a single unit to become an arrow reading algorithm.

E. PATTERN DETECTION PROCESS IN NAVIGATION IMAGES

The main objective of this research is to successfully identify the direction of the arrow or image object so that the AGV prototype can move by utilizing computer vision and the Raspberry Pi Camera V2 sensor. Images from real-time videos from the camera were converted into grayscale images to reduce the complexity of the color dimensions since the RGB color input was insufficient to detect objects in front of the camera. Then, noises in grayscale images must be reduced to achieve maximum accuracy. It is also necessary to determine the ROI in the noise reduction process so that the pattern is detected. Next, the image must be defined or converted to an HSV value.

The tuple values of one and zero were compared, then the pattern was detected using a look-up table which was part of the ROI recognition method. If the segment value was greater than the threshold value, the segment value was set to 1; otherwise, it was set to 0. The segmentation value must match the look-up table in order to determine the navigation direction on the AGV.

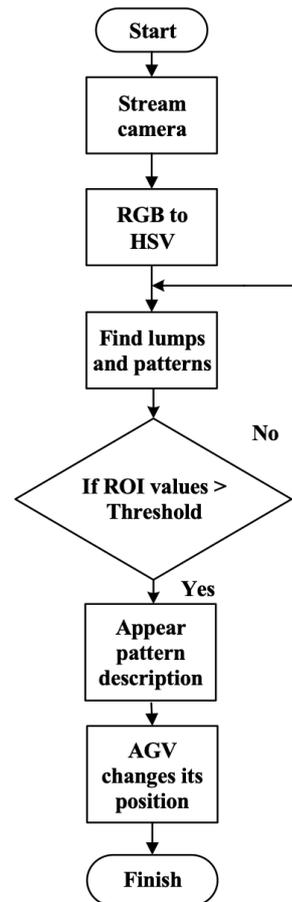


Figure 4. Flowchart of the image processing.

The AGV would move forward when a move forward image was detected, turn back when a turn back image was recognized, turn left when a turn left image was detected, and turn right when a turn right image was identified. The ROI marking for arrow direction detection using the ROI method is depicted in Figure 3. The flowchart of this image processing process is shown in Figure 4.

F. KINEMATICS MODEL OF AGV

The AGV prototype model is based on an excellent kinematic model for the slow-speed representation [29]; to design an AGV prototype is to define a kinematics model that regulates the mechanism of its movement. The kinematics model of vehicles was intended for real-time simulation purposes. Therefore, the computation time must be maintained as low as possible. Subsequently, the detailed modeling of differential drive robots, especially kinematics models will be discussed. The robot had four drive wheels that controlled the AGV’s movement. This AGV robot prototype considered a

differential drive kinematics robot in which wheel speed served as the AGV's motion controller. The front and rear wheels were connected by a metal axle or chassis. From this AGV model, the four-wheel configuration illustrates the AGV's degrees of freedom; the first allows the AGV to move forward and backward, while the second allows the AGV to turn, turn right, or turn left by setting the value of γ (in π rad). Global coordinates of the (X, Y) AGV are illustrated in Figure 5. The v_b parameter is the speed of the AGV's rear wheels (m/s); v_f is the speed of the AGV's front wheels (m/s); l_a and l_b represent the distance from the center of AGV's mass to R and P shafts (m); meanwhile, l_f is the distance from the AGV shaft to the Q point. The AGV kinematics is formulated as in (3) to (5).

$$\dot{X} = V \cdot \cos(\Psi + \beta) \quad (3)$$

$$\dot{Y} = V \cdot \sin(\Psi + \beta) \quad (4)$$

$$\dot{\Psi} = \frac{V \cdot \cos \beta}{l_a + l_b + l_f} \quad (5)$$

The β parameter represent the AGV's slip angle (in π rad), the direction of the AGV was determined by the shaft that was determined by the angle of γ , and the initial speed of the AGV was determined by the speed of rear wheels v_b , the value of slip angle β and the linear speed of the AGV are expressed in (6) and (7).

$$\beta = \tan^{-1} \left(\frac{l_f \cdot \tan \gamma}{l_a + l_b + l_f} \right) \quad (6)$$

$$V = \frac{v_b \cdot \cos \gamma + v_b}{2 \cos \beta} \quad (7)$$

The position of the AGV was then determined by applying the integration in (3) to (5) with respect to time t (s) as shown in (8) to (10).

$$X = \int_0^t V \cdot \cos(\Psi + \beta) dt, \quad (8)$$

$$Y = \int_0^t V \cdot \sin(\Psi + \beta) dt, \quad (9)$$

$$\Psi = \int_0^t \frac{V \cdot \cos \beta}{l_a + l_b + l_f} (\tan \gamma) dt. \quad (10)$$

The values of l_a , l_b , l_f , v_b , $\cos \beta$, $\cos \gamma$, and $\tan \gamma$ parameters were regarded as constants, therefore

$$X = \frac{V \cdot \cos \beta}{l_a + l_b + l_f} \cdot (\tan \gamma) \cdot \left[\sin \left(\frac{V \cdot \cos \beta}{l_a + l_b + l_f} \cdot (\tan \gamma) \cdot t + \beta \right) - \sin \beta \right] \quad (11)$$

$$Y = \frac{V \cdot \cos \beta}{l_a + l_b + l_f} \cdot (\tan \gamma) \cdot \left[\cos \beta - \cos \left(\frac{V \cdot \cos \beta}{l_a + l_b + l_f} \cdot (\tan \gamma) \cdot t + \beta \right) \right] \quad (12)$$

$$\Psi = \frac{V \cdot \cos \beta}{l_a + l_b + l_f} \cdot (\tan \gamma) \cdot t. \quad (13)$$

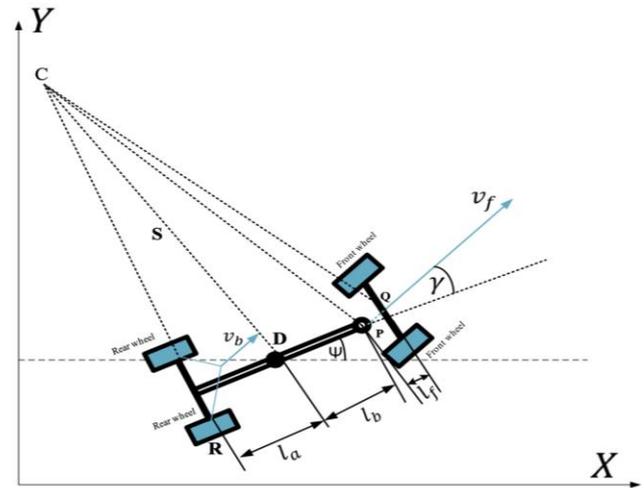


Figure 5. Framework of the AGV model.

The X , Y , and Ψ parameters are global coordinat (X, Y) of the AGV and AGV's turning angle at that time, respectively [30].

III. RESULT AND DISCUSSION

A. IMAGE PROCESSING TESTING

At this stage, testing was carried out when the camera caught the pattern of the detected object. If the detected pattern was correct, a red square box separated the object identified as a direction pattern from other objects which are not. Then, a descriptive text was generated and displayed; it gave the pattern-identified object a name. Four images were used as AGV navigation inputs. Figure 6 depicts results of the directional sign identification for the AGV navigation. Based on Figure 6(a) to Figure 6(d), it is evident that when the arrow direction is detected by the AGV robot camera, the system detects the pattern immediately and displays the results on the monitor screen. When the arrow direction as in Figure 6(a) was detected, the text "Move Straight" was generated, then the AGV would run straight. When the arrow direction as in Figure 6 (b) was detected, the text "Turn Back" was generated, then the AGV would subsequently maneuver back and forth. Furthermore, when the arrow direction as in Figure 6(c) was detected, "Turn Left" was generated, resulting in the AGV maneuvering to the left. Finally, when the arrow direction as in Figure 6(d) was detected, "Turn Right" was generated, and the AGV would then maneuver to the right. However, the arrow direction could not be detected when it was reversed or rotated. The results of testing the detection of marks and wheel rotations on the AGV will be displayed in the next discussion topic.

The results of the directional detection test for the AGV navigation with the distance parameter from multiple test trials are presented in Table I. At a distance of 10 cm to 80 cm, the percentage of detection success was 100%, with a total of three trials. At a distance of 90 cm and 95 cm, the percentage of success was 33.3%, with a total of three trials. The following equation was used to calculate the total percentage of success.

$$\frac{\text{Total of distance success}}{\text{Total of distance trial}} \times 100\% = \text{percentage of success} \quad (14)$$

$$\frac{20}{27} \times 100\% = 81.4 \%$$

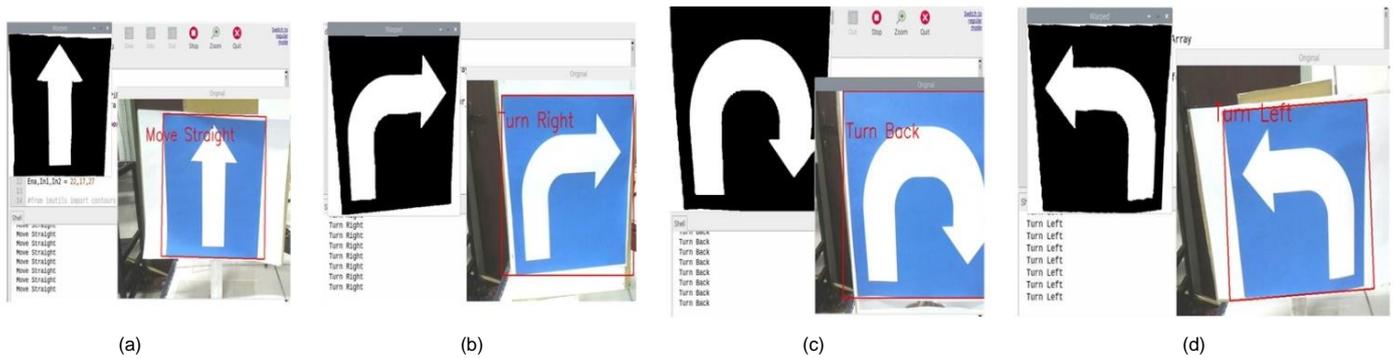


Figure 6. Image test results on the navigation with different size, (a) move straight, (b) turn right, (c) turn back, and (d) turn left.

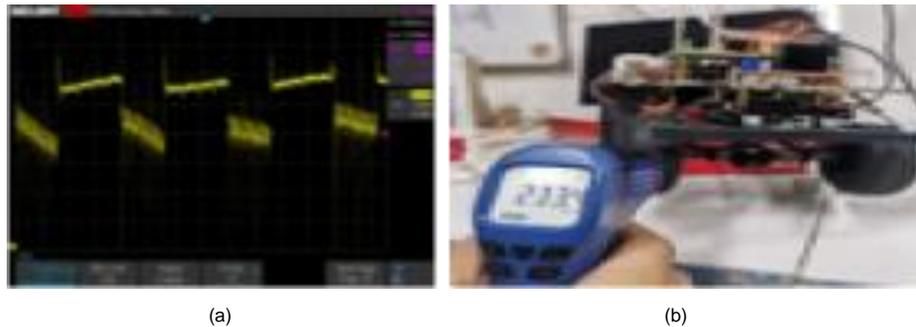


Figure 7. Measurement results using an oscilloscope and tachometer, (a) PWM duty cycle of 60%, (b) speed result of PWM duty cycle of 60%.

TABLE I
TESTING OF NAVIGATION DIRECTION DETECTION

No.	Distance	Experiment		
		1	2	3
1	10 cm	Success	Success	Success
2	25 cm	Success	Success	Success
3	65 cm	Success	Success	Success
4	70 cm	Success	Success	Success
5	75 cm	Success	Success	Success
6	80 cm	Success	Success	Success
7	85 cm	Success	Success	Fail
8	90 cm	Success	Fail	Fail
9	95 cm	Success	Fail	Fail

Total of distance success is the total of success in detecting objects from experiments that were carried out, Total of distance trial is the total of trial that was done. Using (14), the obtained percentage of success of the detected object was 81.4%.

B. AGV SPEED MEASUREMENT RESULTS

At this stage, the rear wheel rotation of the AGV was measured with pulse width modulation (PWM) measurement parameters using an oscilloscope and a digital tachometer. Measurements were made by setting the PWM duty cycle from 25% to 95%. The rotational speed of the AGV wheel was influenced by the size of the PWM duty cycle. The initial measurement was done by changing the PWM duty cycle to 60%. Figure 7(a) depicts the PWM signal displayed on the oscilloscope, whereas Figure 7(b) depicts the resulting speed of 233.4 rpm.

IV. CONCLUSION

This study has successfully implemented an image processing algorithm using the ROI method to detect four patterns on objects in the form of directional sign images. The detection was carried out using the AGV prototype, with a total

success percentage of 81.4%. The stages of the AGV navigation process worked well, and image processing that utilized computer vision was able to read commands well. The test parameter was the distance between the image object and the camera. In addition, the DC motor and L298N driver module on the AGV exhibited outstanding performance in regulating torque, speed, and PWM duty cycle. This AGV robot can serve as a reference for further robot development. Further research can utilize other image processing methods, such as pattern recognition methods and sign traffic recognition methods.

CONFLICT OF INTEREST

The authors declare no conflict of interest in this research.

AUTHOR CONTRIBUTION

Conceptualization, Florentinus Budi Setiawan and Rachmat Hidayat; methodology, Leonardus Heru Pratomo; system design, Florentinus Budi Setyawan and Rachmat Hidayat; software, Rachmat Hidayat; analysis, Rachmat Hidayat; writing—original draft preparation, Florentinus Budi Setyawan; writing—review and editing, Florentinus Budi Setyawan, Leonardus Heru Pratomo, and Slamet Riyadi.

REFERENCES

- [1] J.H. Chou, "Automatic Guided Vehicle," *Proc. 1995 Int. IEEE/IAS Conf. Ind. Autom. Control Emerg. Technol.*, 1995, pp. 241-245, doi: 10.1109/iacet.1995.527570.
- [2] E.M. Ngandu, N. Luwes, and K. Kusakana, "Navigation System for an Automatic Guided Vehicle," *J. Phys. Conf. Ser.*, Vol. 1577, pp. 1-11, 2020, doi: 10.1088/1742-6596/1577/1/012030.
- [3] S. Legowik, R.V Bostelman, T. Hong, dan E.R. Messina, "Guideline for Automatic Guided Vehicle Calibration," National Institute of Standards and Technology, Gaithersburg, MD, USA, Interagency/Internal Report, NISTIR 8168, Mar. 2017., doi: 10.6028/NIST.IR.8168.
- [4] X. Zhang and S. Xu, "Research on Image Processing Technology of Computer Vision Algorithm," *2020 Int. Conf. Comput. Vision, Image Deep Learn. (CVIDL)*, 2020, pp. 122-124, doi: 10.1109/CVIDL51233.2020.00030.

- [5] R. Bostelman, T. Hong, and G. Cheok, "Navigation Performance Evaluation for Automatic Guided Vehicles," *IEEE Conf. Technol. Pract. Robot Appl. (TePRA)*, 2015, pp. 1-6, doi: 10.1109/TePRA.2015.7219684.
- [6] M. Antony *et al.*, "Design and Implementation of Automatic Guided Vehicle for Hospital Application," *2020 5th Int. Conf. Commun., Electron. Syst. (ICCES)*, 2020, pp. 1-6, doi: 10.1109/ICCES48766.2020.09137867.
- [7] M.N. Tamara *et al.*, "Electronics System Design for Low Cost AGV Type Forklift," *2018 Int. Conf. Appl. Sci., Technol. (iCAST)*, 2018, pp. 464-469, doi: 10.1109/iCAST1.2018.8751559.
- [8] X. Zhou, T. Chen, and Y. Zhang, "Research on Intelligent AGV Control System," *Proc. 2018 Chinese Autom. Congr. (CAC)*, 2019, pp. 58-61, doi: 10.1109/CAC.2018.8623384.
- [9] J. Lee, C.H. Hyun, and M. Park, "A Vision-Based Automated Guided Vehicle System with Marker Recognition for Indoor Use," *Sensors*, Vol. 13, No. 8, pp. 10052-10073, Aug. 2013, doi: 10.3390/s130810052.
- [10] J.V. Stone, "Computer Vision: What Is The Object?," *Artif. Intell. Simul., Behav.*, 1993, pp. 199-208.
- [11] D. Priyanka *et al.*, "Traffic Light and Sign Detection for Autonomous Land Vehicle Using Raspberry Pi," *2017 Int. Conf. Invent. Comput. Inform. (ICICI)*, 2018, pp. 160-164, doi: 10.1109/ICICI.2017.8365328.
- [12] W. Nugraha, M. Syarif, and W.S. Dharmawan, "Penerapan Metode SDLC Waterfall dalam Sistem Informasi Inventori Barang Berbasis Desktop," *JUSIM (J. Sist. Inf. Musirawas)*, Vol. 3, No. 1, pp. 22-28, May 2018, doi: 10.32767/jusim.v3i1.246.
- [13] C. Gao and G. Hembroff, "Implications of Modified Waterfall Model to the Roles and Education of Health IT Professionals," *2012 IEEE Neww. Oper. Manage. Symp.*, 2012, pp. 1368-1369, doi: 10.1109/NOMS.2012.6212076.
- [14] A. Chakraborty, "Image Processing and Image Pattern Recognition a Programming Tutorial," *2018 First Int. Conf. Artif. Intell. Ind.*, 2018, pp. 122-123, doi: 10.1109/AI4I.2018.8665702.
- [15] R. Helbet, V. Monda, A.C. Bechet, and P. Bechet, "Low Cost System for Terrestrial Trunked Radio Signals Monitoring Based on Software Defined Radio Technology and Raspberry Pi 4," *2020 Int. Conf., Expo. Elect., Power Eng. (EPE)*, 2020, pp. 438-441, doi: 10.1109/EPE50722.2020.9305536.
- [16] TowerPro, "MG996R High Torque Metal Gear Dual Ball Bearing Servo," MG996R datasheet, Aug. 2015.
- [17] B. Hamzi *et al.*, "Implementation of Data Driven Control System," *Electron.*, Vol. 8, No. 2, pp. 1801-1804, 2021.
- [18] S.S. Babu and A. Sukesh, "Current Programmed Controlled DC-DC Converter for Emulating the Road Load in Six Phase Induction Motor Drive in Electric Vehicle," *2020 Int. Conf. Power Electron., Renew. Energy Appl. (PEREA)*, 2020, pp. 1-6, doi: 10.1109/PEREA51218.2020.9339779.
- [19] A.A. Putri and T. Aditya, "3D Modelling and Visualization of Drinking Water Supply System Using 3D GIS," *2017 7th Int. Annu. Eng. Sem. (InAES)*, 2017, pp. 1-6, doi: 10.1109/INAES.2017.8068574.
- [20] F. Liu and Z. Yang, "Design of VMware vSphere Automatic Operation and Maintenance System Based on Python," *2018 Int. Conf. Adv. Mechatron. Syst. (ICAMechS)*, 2018, pp. 283-286, doi: 10.1109/ICAMechS.2018.8506789.
- [21] X. Farhodov *et al.*, "Faster RCNN Detection Based OpenCV CSRT Tracker Using Drone Data," *Int. Conf. Inf. Sci., Commun. Technol. (ICISCT)*, 2019, pp. 1-3, doi: 10.1109/ICISCT47635.2019.9012043.
- [22] A. Ohno, S. Matsumoto, M. Ohshita, and K. Kaida, "A Learning Support System of C Programming Language for Novices as a Platform for Learning Analytics," *2020 9th Int. Congr. Adv. Appl. Inform. (IIAI-AAI)*, 2020, pp. 270-273, doi: 10.1109/IIAI-AAI50415.2020.00060.
- [23] H.N. Chi, P.J. Lee, and C.L. Lo, "Visual Tracking Cleaner - A Robot Implements on the Whiteboard," *2020 Int. Conf. Syst. Sci., Eng. (ICSSE)*, 2020, pp. 1-4, doi: 10.1109/ICSSE50014.2020.9219288.
- [24] N.M. Ali, N.K.A. Md Rashid, and Y.M. Mustafah, "Performance Comparison between RGB and HSV Color Segmentations for Road Signs Detection," *Appl. Mech. Mater.*, Vol. 393, pp. 550-555, Sep. 2013, doi: 10.4028/www.scientific.net/AMM.393.550.
- [25] G. Kumar, P.P. Sarthi, P. Ranjan, and R. Rajesh, "Performance of K-Means Based Satellite Image Clustering in RGB and HSV Color Space," *2016 Int. Conf. Recent Trends Inf. Technol. (ICRTIT)*, 2016, pp. 1-5, doi: 10.1109/ICRTIT.2016.7569523.
- [26] A. Ajmal, C. Hollitt, M. Freaan, and H. Al-Sahaf, "A Comparison of RGB and HSV Colour Spaces for Visual Attention Models," *2018 Int. Conf. Image, Vis. Comput. New Zeal. (IVCNZ)*, 2018, pp. 1-6, doi: 10.1109/IVCNZ.2018.8634752.
- [27] S. Sudha, K.B. Jayanthi, C. Rajasekaran, and T. Sunder, "Segmentation of RoI in Medical Images Using CNN- A Comparative Study," *TENCON 2019 - 2019 IEEE Reg. 10 Conf. (TENCON)*, 2019, pp. 767-771, doi: 10.1109/TENCON.2019.8929648.
- [28] C. Shi-Gang *et al.*, "Study on Segmentation of Lettuce Image Based on Morphological Reorganization and Watershed Algorithm," *2018 Chin. Control, Decis. Conf. (CCDC)*, 2018, pp. 6595-6597, doi: 10.1109/CCDC.2018.8408290.
- [29] A.J. Weinstein and K.L. Moore, "Pose Estimation of Ackerman Steering Vehicles for Outdoors Autonomous Navigation," *2010 IEEE Int. Conf. Ind. Technol.*, 2010, pp. 579-584, doi: 10.1109/ICIT.2010.5472738.
- [30] N. Alfiany *et al.*, "Kinematics and Simulation Model of Autonomous Indonesian 'Bekak' Robot," *2020 IEEE Reg. 10 Symp. (TENSYPMP)*, 2020, pp. 1692-1695, doi: 10.1109/TENSYPMP50017.2020.9230782.