JURNAL INFOTEL
Informatics - Telecommunication - Electronics
Website: http://ejournal.st3telkom.ac.id/index.php/infotel
ISSN: 2085-3688; e-ISSN: 2460-0997

Implementation of line detection self-driving car using HSV method based on raspberry pi 4

Florentinus Budi Setiawan[1,*], Eric Pratama Putra[1], Leonardus Heru Pratomo[1], Slamet Riyadi[1]
[1]Departement of Electrical Engineering, Soegijapranata Catholic University
[1]Pawiyatan Luhur Street, No. 1, Semarang 50234, Indonesia
[*]Corresponding email: f.budi.s@unika.ac.id

Received 7 July 2022, Revised 21 August 2022, Accepted 1 September 2022

Abstract — With the development of technology, especially in industries, daily human activities can be carried out with artificial intelligence Robots. In this case, self-driving cars have several methods with GPS systems, radar, lidar, or cameras. In this study, a self-driving car system was designed on a navigation path model using a street mark detector with a camera as a vision sensor. This self-driving car system is used to move on a path on the detected line. This research uses the HSV method for line detection. This robot research uses OpenCV, which functions for image processing using dilate, morphologyEx, and Gaussian Blur processing methods. This research produces a fairly accurate level of path reading when compared to Robots that use the Line Follower Method. After several processing stages have been passed and the path has been read, this robot will move the motor following the path detected by the camera. Based on the hardware implementation through testing in the Autonomous Car laboratory with Computer Vision, it can work according to the algorithm. By using image processing with the HSV method, the resulting level of precision can reach the average value according to the direction of the self-driving car.

Keywords – HSV, image processing, raspberry pi 4, self-driving car

I. INTRODUCTION

The advantages of application in the field of technology in the modern era like today make every developed country proud in the world, especially in terms of scientific progress in transportation [1]. Transportation itself is an important part of human life. One of the most widely used private transportation is the car. Therefore, a technology was developed to drive a car automatically or a self-driving car [2]. A self-driving car is a vehicle that can move and recognize objects around it without human control. This vehicle uses various sensors such as radar, Light Detection and Range (LIDAR), a Global Positioning System (GPS), odometry, and cameras to detect objects in the vicinity [3]. Sensors generate information about things that the control system will process in self-driving cars to identify models of navigation paths and obstacles. In designing a self-driving car system that provides a model design that combines multiple sensors [4], this research will develop and manufacture a self-driving car system on an autonomous car prototype that can move along a track automatically using a camera as a sensor. Vision to detect road signs, so the car continues to run following the detected pattern [5].

The main goal of self-driving cars is to solve the functionality problem of self-driving cars, namely the navigation path design model. Planning the navigation path model is the main requirement so that that self-driving car can run in the right direction [6]. The application of self-driving cars is vision processing, and the lane model can be determined based on the object of road markings that will be traversed by this car. Therefore, street mark detection is the easiest solution to solve the lane model planning problem. All information is captured in real-time using camera sensors to detect road objects. Therefore, based on previous research on self-driving cars that provided a model design that combines several sensors [7] and research on self-driving cars using the Kinematic model [8], this study designed a self-driving car. The HSV algorithm system with the addition of a camera as a vision sensor to detect the street mark and a colour quantization feature method using thresholding clustering to overcome the shortcomings of this autonomous car prototype [9].

This study designed a self-driving car system using a prototype autonomic car that can move automatically. By following road markings detected using the camera as a vision sensor and using the Hough transforms method to adjust the colour in the object so that the image taken by the object in real-time can be using a Raspberry Pi 4 with the HSV filtering method.

II. Research Method

This section discusses the self-driving car systems and components, self-driving car hardware design, HSV method on a self-driving car, real-time object detection, and the Ackerman model of a self-driving car.

A. Self-Driving Car Systems and Components

Fig. 1 illustrates the controller hardware scheme of a self-driving car. In this design, the hardware components used are Raspberry Pi 4 Model B as a microcontroller, Raspberry camera as a vision sensor, L298N motor driver as front-wheel drive, BTS7960 motor driver as rear-wheel drive, Arduino nano as driver liaison, 12 V ACCU 12 Ah. The device functions are as follows:

a) Raspberry Pi 4 Model B
The Raspberry Pi is a minicomputer developed in the UK by Raspberry Pi Foundation. Raspberry Pi 3 Model B, was released in February 2016. Broadcom BCM2837 SoC has a 64-bit quad-core ARM Cortex-A53 processor, running at higher speed up to 1.2 GHz, 1 GB of RAM, onboard Wi-Fi, Bluetooth up to 1.2 GHz, and USB boot capability [10]. When using the Raspberry Pi 4, there are several systems, one of which is a system for real-time programming performance to achieve stable and not slow results. Several companion components are needed, namely a microSD and a cooling fan (heat sink). To realize the scope of objects to be detected effectively and not too slowly.

b) Pi Camera
The Pi camera is a module that sends data to the Raspberry Pi 4 using an elastic cable to support sending programs. In addition, the camera module also has a high-speed connection capable of sending images and recording videos with high resolution [11].

c) Arduino Nano
Arduino Nano [12] is a small, easy-to-use board based on the ATmega328P architecture with an ATmega328 microcontroller with an operating voltage of 5 V. It uses 32 KB of flash memory, of which the bootloader uses 2 KB. The Arduino Nano processor clock speed is 16 MHz, which serves to send the program to the BTS7960 driver.

d) Driver Motor L298N
The L298N motor driver is a component or driver module for DC motors. The L298N driver is a speed regulator and can control the direction of rotation of the DC motor [13]. Therefore, the L298N motor driver is perfect for robot design applications and is ideal for connecting to a microcontroller or Raspberry Pi 4 Model B.

e) This DC motor driver can produce up to 43 A of current by functioning as a front-wheel-drive DC motor. The DC source voltage can be supplied is between 5.5 V - 27 V DC, while the input voltage level is between 3.3 V - 5 V DC [14]. This motor driver uses a complete H-bridge circuit with BTS7960 IC overheating and overcurrent protection.

Fig. 1 is a hardware design scheme that will be used.

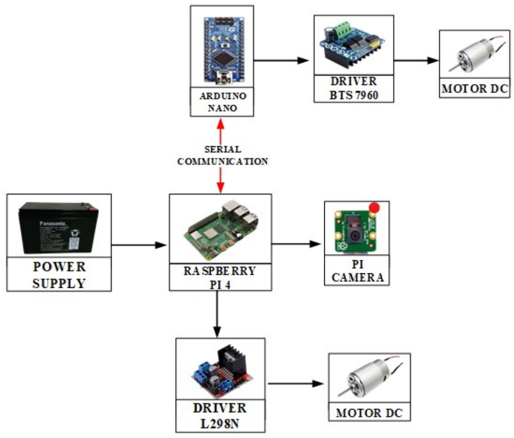


Fig. 1. Hardware schematic.

B. Self-Driving Car Hardware Design

The hardware used in this self-driving car is a prototype autonomous car with a drive system, namely a DC motor for speed control. The sensor used in this self-driving car is the Raspberry Pi camera. At the same time, the controller used is a Raspberry Pi 4 Model B and an Arduino nano which functions as a program. As for the motor driver, L298N and BTS7960 drivers are used, which function as DC motor regulators. Fig. 2 is a prototype of an autonomous car.

Self-driving is designed to detect the location of road marking objects through the results of the image processing taken by the Raspberry camera. The camera is operated as an image receiver by the Raspberry Pi 4 Model B minicomputer.

The coordinates processing results of detecting road markings from the image are sent to the Raspberry Pi 4 through communication between the camera and the Raspberry Pi 4. In addition, detecting object result of the street mark are forwarded to run the dc motor as the steering wheel and the movement of the self-driving car [15]. The image processing

Fig. 2. Prototype autonomous car.

process for detecting road signs and the working system relationship between hardware, can be seen in figure below.

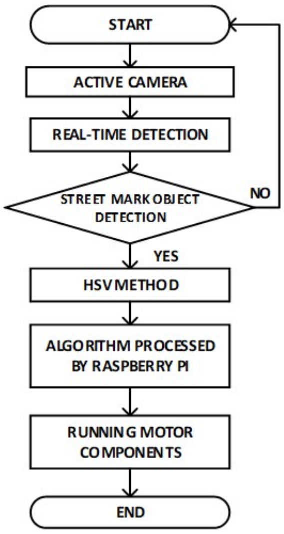


Fig. 3. Flowchart system of self-driving car.

The prototype autonomous car in this design uses a 12 V, 12 Ah battery, which a voltage source used to supply power to the Raspberry Pi 4 and DC motor.

C. HSV Method on Self-Driving Car

HSV (Hue, Saturation, Value) is one of the methods used to apply this self-driving car. HSV functions in colour selection because the HSV method approaches the visual way of the human eye in distinguishing colours. HSV is the result of the transformation of the RGB colour form (Red, Green, Blue), and the light intensity or colour brightness value varies from 0 to 100 [16].

The HSV colour method is beneficial in detecting street marks by segmenting based on the lowest colour range to the highest colour on street mark objects. HSV has three main characteristics, namely Hue, Saturation, and Value which have different values. The Hue value is changed from 0 to 255, with the value 0 being red. Saturation describes the intensity of the colour that includes values 0 to 255. If the saturation value is less, it will show a grey colour. The deal consists of values 0 to 255, with 0 being the dark colour and the value 255 being the red colour. The range of the HSV colour space used in this study is shown in Fig. 4.

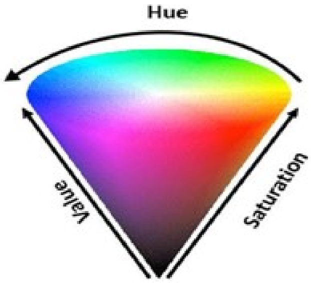


Fig. 4. The range of the HSV colour.

The HSV space method has the following formula, with the notation shown in Table 1.

Table 1. Notations of the HSV Space Method

Notation	Definition
R'	Red value
G'	Green value
B'	Blue value
C_{max}	Maximum contrast value
C_{min}	Minimum contrast value
H	Hue
S	Saturation
V	Value
$\triangle$	Determination of C_{max} and C_{min}

$$R' = \frac{R'}{255}, G' = \frac{G'}{255}, B' = \frac{B'}{255} \tag{1}$$

Eq. (1) shows the RGB colour composition or the initial input of the colour combination to be used

$$C_{max} = max(R', G', B') \tag{2}$$

$$C_{min} = min(R', G', B') \tag{3}$$

Eqs. (2) and (3) are the determination of the maximum contrast and minimum contrast in the RGB colour composition used.

$$\triangle = C_{max} - C_{min} \tag{4}$$

Eq. (4) is the result between maximum contrast and minimal contrast. Meanwhile, (5) is an equation that produces hue values using the R, G, and B colour values, Minimum Contrast, and Maximum Contrast.

Thus, Hue's calculation becomes:

$$H = \begin{cases} 0^0 & ; \triangle = 0, \\ 60^0 \times \left(\frac{G'-B'}{\triangle} \mod 6 \right) & ; C_{max} = R', \\ 60^0 \times \left(\frac{B'-R'}{\triangle} + 2 \right) & ; C_{max} = G', \\ 60^0 \times \left(\frac{B'-G'}{\triangle} + 4 \right) & ; C_{max} = B' \end{cases} \tag{5}$$

Eq. (5) produces hue values using the R, G, and B colour values, Minimum Contrast, and Maximum Contrast.

The Saturation calculation is shown in (6):

$$f(x;\lambda,k)=\begin{cases}0 & ;C_{max}=0,\\ 0 & ;C_{max}\neq 0\end{cases} \tag{6}$$

(6) is the value used to produce the saturation value by using subtraction and division of the minimum and maximum contrast values
And the Value calculation

$$ContrastValue=\frac{C_{max}}{C_{min}} \tag{7}$$

(7) shows the value by determining the maximum or minimum contrast value.

If the object's colour composition is within the specified HSV value range, the object will be white as assumed to match the range value. Meanwhile, items with a colour composition outside the range value will change colour to black. Fig. 5 is the threshold process for changing the HSV colour range.

Fig. 5. Thresholding process based on HSV color range.

The image above shows the thresholding process in maintaining the original object image with the modified image using the HSV color range method.

D. Real-Time Object Detection

Video-based camera sensors are used in real-time to detect road signs. Real-time detection will be processed if the frame meets the criteria for the appropriate range value at calibration time. For example, in self-driving car research, calibration is needed to determine the HSV colour value used as input in measuring road marking objects in real-time video testing [17]. Fig. 6 is a Flowchart of object detection in real-time.

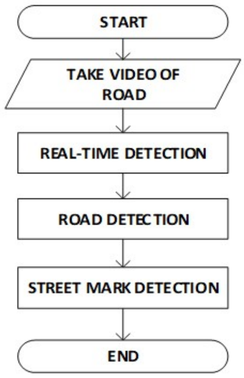


Fig. 6. Real-time detection flowchart.

The flowchart above shows how computer vision works. First, the program runs, then the camera for detecting road markings will be active and display a video stream for detecting roads that will be traversed by autonomous cars.

E. Ackerman Model on a Self-Driving Car

The Autonomous Car prototype assumes the kinematic model that there is no slip angle between the front and rear wheels are the same therefore, the kinematic model applies to low-speed applications [18]. The front and rear wheels are connected by a metal frame that allows the autonomous car to move forward and backwards simultaneously [19], [20]. Therefore, in this autonomous car turning motion, it will be possible to form an angle or coordinate point. The following is the equation of the kinematic model of the autonomous car, where the notation is shown in Table 2.

Table 2. Notations of the Kinematic Model

Notation	Definition
$\dot{X}$	X coordinate
$\dot{Y}$	Y coordinate
v_b	Rear wheels
v_f	Front wheels
l_a, l_b	Length from the center of mass AGV to axis P
l_f	Length from AGV axis to point Q
β	Slip angle of AGV
dt	Change over time
$\dot{\psi}$	Change in angle at the center of the axis
γ	Gamma angle Y

$$\dot{X}=V\cos(\psi+\beta) \tag{8}$$

$$\dot{Y}=V\sin(\psi+\beta) \tag{9}$$

$$\dot{\psi}=V\cos(\beta)\frac{\tan\delta_f}{L} \tag{10}$$

Here, $\beta(\pi rad)$ represents the slip angle of the AGV, the direction of the AGV is determined by the axis determined by the gamma angle γ. The initirear wheel determines the initial speed of the autonomous car. Speed, the slip β angle value and the linear speed of the autonomous car are written as the following equation.

$$\beta = \tan^{-1}\left(\frac{l_f \tan\gamma}{l_a + l_b + l_f}\right) \quad (11)$$

$$V = \frac{v_b \cos(\gamma) + v_b}{2\cos(\beta)} \quad (12)$$

The AGV position is then determined by applying the integration in (8) – (10) for time $t(s)$.

$$X = \int_0^t V\cos(\psi + \beta)dt \quad (13)$$

$$Y = \int_0^t V\sin(\psi + \beta)dt \quad (14)$$

$$Z = \int_0^t V\cos(\psi + \beta)dt \quad (15)$$

Fig. 7 shows the Ackerman frame model of the self-driving car.

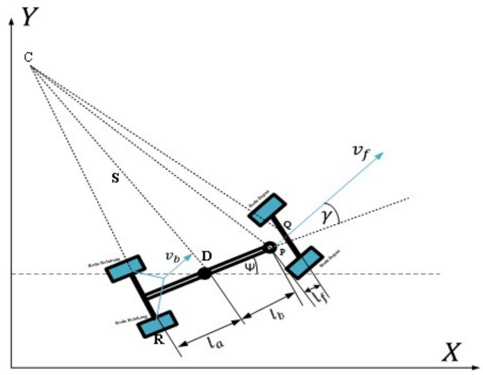


Fig. 7. Frame Ackerman autonomous car.

III. Result and Discussion

This test is carried out to determine the function of the prototype autonomous car, which has an explanation of the stability and ability of the self-driving car system to detect street marks. That way, we can see how fast the real-time system detects street-mark objects. With a real-time system, self-driving cars can calculate the detection distance based on the detection area.

A. Self-Driving Car Speed Test Based on Turning Angle

To determine the ability of the prototype autonomous car to move according to a predetermined path. So it is necessary to test the speed and turning angle. This test aims to determine the rate of the autonomous car linearly. The linear speed itself is influenced by the rotation of the DC motor used in the prototype autonomous car. So test is carried out with a tachometer using the robot running on a predetermined path, and the rotational speed of the wheels is calculated using a tachometer. The measurement results obtained the average speed value in units of RPM (revolutions per minute). The results of the rotation speed measurement can be seen in Table 3.

Table 3. Rotation Speed

No.	PWM duty cycle (%)	Rotation Speed (RPM)	Autonomous Car Speed Condition
1	30	125.5	Slow
2	40	210.8	Slow
3	50	274.8	Moderate
4	60	293.5	Moderate
5	70	418.9	Fast

It can be seen in Table 3 that the speed test is measured at a duty cycle of 30% to 70%. At a duty cycle of 30% to 40%, the motor in an autonomous car will run slowly in the range of 125 rpm to 210 rpm. 50% to 60% duty cycle, the autonomous car motor will rotate 274 rpm to 293 rpm. While 70% duty cycle, the autonomous car motor will move up 418 rpm. If the duty cycle increases high, the motor will rotate quite fast according to the duty cycle used. It can be seen the wheels rotational speed when the autonomous car is running on the track. This test influences how we set up its PWM duty cycle.

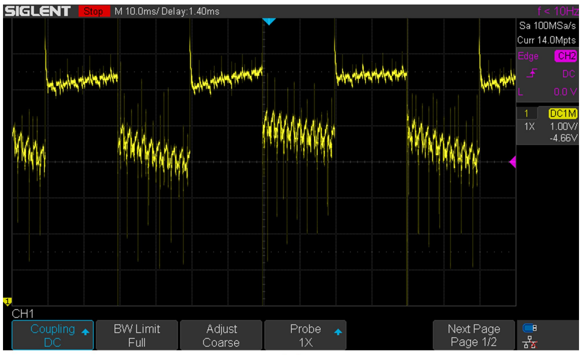
For testing, the autonomous car will be strengthened by the data from the output signal. These results are obtained through the image of the oscilloscope measurement results. The image from the oscilloscope will later show the results of the PWM signal measurement, as shown in Table 3.

Based on the test results, it can be analyzed that the autonomous car can turn at a maximum angle. The PWM value determines whether an autonomous car can run slow or fast. The first measurement result with a PWM duty cycle of 30%, shown in Fig. 8 (a), is the result of the signal on the oscilloscope. Fig. 8 (b) shows the resulting speed of 125.5 RPM. The second measurement result with a PWM duty cycle of 70%, shown in Fig. 9 (a), is the result of the signal on the oscilloscope. Fig. 9 (b) shows the resulting speed of 418.5 RPM.

B. HSV Color test on Self-Driving Car

HSV testing is carried out to check the processing of street mark objects on a predetermined path. In this test, the thresholding method is used, namely measuring the maximum and minimum limits. In addition, this test is carried out to distinguish the white and black colours found in the HSV method, which have a maximum or minimum value.

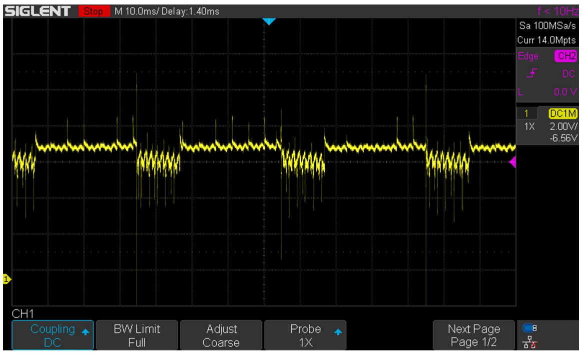
To find out the maximum and minimum limits of the HSV method, we do a test by distinguishing the filter from black objects and white objects. This test is done by activating the program contained in the Raspberry Pi 4 Model B. Fig. 10, Fig. 11, Fig. 12 and Table 4, Table 5, Table 6, represent the testing of the HSV method.



(a)

(b)

Fig. 8. (a) Signal duty cycle PWM 30%, (b) Acceleration output 125.5 RPM.



(a)

(b)

Fig. 9. (a) Signal duty cycle PWM 70%, (b) Acceleration output 418.5 RPM.

Fig. 10. Hue threshold test.

Table 4. Hue Threshold Test

Hue Min	Hue Max	White Object	Black Object
155	179	Detected	Not Detected

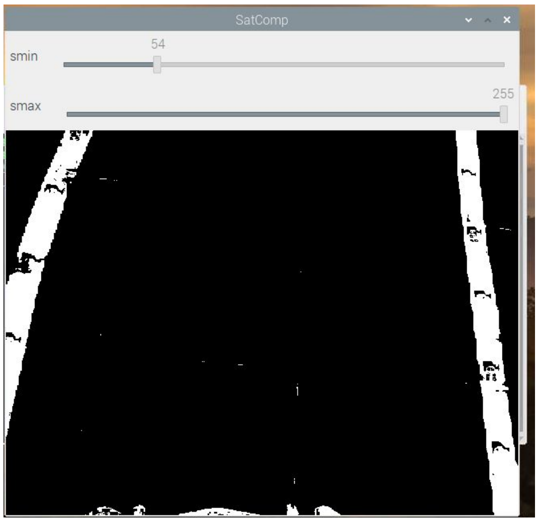


Fig. 11. Saturation threshold test.

Table 5. Saturation Threshold Test

Saturation Min	Saturation Max	White Object	Black Object
54	255	Detected	Not Detected

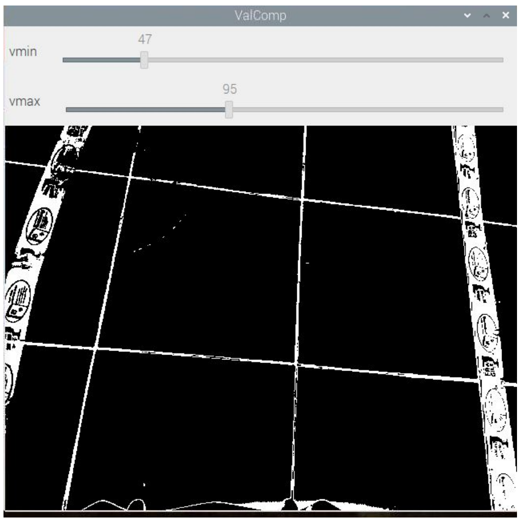


Fig. 12. Value threshold test.

Table 6. Value Threshold Test.

Value Min	Value Max	White Object	Black Object
47	95	Detected	Not Detected

Based on Fig. 10, Fig. 11, and Fig. 12, the thresholding calibration value is determined by finding the minimum and maximum values of the HSV (Hue, Saturation, Value) method, which aims to detect white and black objects under street mark object detection.

C. Final Result of Street Mark Detection with HSV Method

In this last test, we can see that the autonomous car moves following the street mark object that has been processed based on the HSV method in real-time so that tracking appears on every path the camera has detected. The emergence of this tracking path makes the autonomous car run stably, following the path that has been seen. In addition, the tracking line functions to determine a turning angle for the autonomous car so that the self-driving car system can function properly according to the algorithm that has been created.

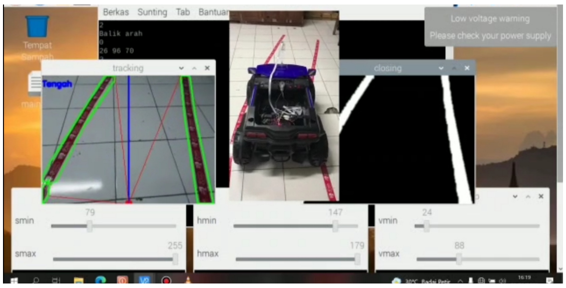


Fig. 13. Self-driving car test.

Based on Fig. 13, the test has been carried out indoors and outdoors. It is better to test this self-driving car indoors to get maximum results, based on tests carried out indoors, there is no interference with the intensity of sunlight. On the other hand, in outdoor testing, the sun's power is very high, so the performance of the HSV method is not optimal. After the text edit has been completed, the paper is ready for the template. Duplicate the template file by using the Save As command, and use the naming convention prescribed by your conference for the name of your paper. In this newly created file, highlight all of the contents and import your prepared text file. You are now ready to style your paper; use the scroll down window on the left of the MS Word Formatting toolbar.

IV. Conclusion

Based on the testing and analysis that has been carried out. The autonomous car prototype has been successfully designed. According to the specifications of the self-driving car system with vision image processing performance using the HSV method as a street mark object detector, thus making autonomous cars run following street mark objects. The threshold used from this calibration result has a minimum and maximum value range that makes this HSV method able to process a road mark object. From this test, the resulting value range is hue min = 155, hue max = 179, saturation min = 54, saturation max = 255, and value min = 47, value max = 95, with the result that HSV detects street marks well. In this process, the Raspberry Pi processes Yahoo by sending data to the Raspberry Camera. So that the Real-Time function on the visual image is active. Which makes the BTS7960 driver and L298N driver run correctly—using a DC motor to adjust the PWM duty cycle to the turning angle on the self-driving car track to determine the speed and torque of a DC motor when an autonomous car is running. The maximum value of the turning angle is proportional to the speed of the autonomous car. The faster the autonomous car runs, the smaller the turning angle and vice versa. A program that has been made and researched is more accurate and has almost no street mark error. In the future, this research will be used for industrial needs, which will then be applied to Autonomous cars in the industry.

Acknowledgment

The author would like to thank Soegijapranata Catholic University for supporting facilities and places to conduct research in the laboratory. Hopefully, this tool can be helpful for technological progress in the world. Also, the author would like to thank the University President for his support.

References

[1] M. V. G. Aziz, A. S. Prihatmanto, and H. Hindersah, "Implementation of lane detection algorithm for self-driving car on toll road cipularang using Python language," in *2017 4th International Conference on Electric Vehicular Technology (ICEVT)*, 2017, pp. 144–148. doi: 10.1109/ICEVT.2017.8323550.

[2] Z. M. U. Din, W. Razzaq, U. Arif, W. Ahmad, and W. Muhammad, "Real time Ackerman steering angle control for self-driving car autonomous navigation," in *2019 4th International Conference on Emerging Trends in Engineering, Sciences and Technology (ICEEST)*, 2019, pp. 1–4. doi: 10.1109/ICEEST48626.2019.8981710.

[3] R. Run and G. Sun, "The cost-effective GPS guided autonomous vehicle - a feasibility study based on self-balancing scooter," in *2017 International Conference on Applied System Innovation (ICASI)*, 2017, pp. 778–780. doi: 10.1109/ICASI.2017.7988546.

[4] H. Cho, Y.-W. Seo, B. V. K. V. Kumar, and R. R. Rajkumar, "A multi-sensor fusion system for moving object detection and tracking in urban driving environments," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 1836–1843. doi: 10.1109/ICRA.2014.6907100.

[5] S. N. K. Rangan, V. G. Yalla, D. Bacchet, and I. Domi, "Improved localization using visual features and maps for autonomous cars," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, 2018, pp. 623–629. doi: 10.1109/IVS.2018.8500540.

[6] J. Newman, Z. Sun, and D.-J. Lee, "Self-driving cars: a platform for learning and research," in *2020 Intermountain Engineering, Technology and Computing (IETC)*, 2020, pp. 1–5. doi: 10.1109/IETC47856.2020.9249142.

[7] A. A. Mahersatillah, Z. Zainuddin, and Y. Yusran, ”Unstructured road detection and steering assist based on HSV color space segmentation for autonomous car,” in *2020 3rd International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, 2020, pp. 688–693. doi: 10.1109/ISRITI51436.2020.9315452.

[8] Y. Chen, S. Chen, T. Zhang, S. Zhang, and N. Zheng, ”Autonomous vehicle testing and validation platform: integrated simulation system with hardware in the loop,” in *2018 IEEE Intelligent Vehicles Symposium (IV)*, 2018, pp. 949–956. doi: 10.1109/IVS.2018.8500461.

[9] E. Oruklu, D. Pesty, J. Neveux, and J. Guebey, ”Real-time traffic sign detection and recognition for in-car driver assistance systems,” in *2012 IEEE 55th International Midwest Symposium on Circuits and Systems (MWSCAS)*, 2012, pp. 976–979. doi: 10.1109/MWSCAS.2012.6292185.

[10] K. B. Swain, S. Dash, and S. S. Gouda, ”Raspberry pi based integrated autonomous vehicle using LabVIEW,” in *2017 Third International Conference on Sensing, Signal Processing and Security (ICSSS)*, 2017, pp. 69–73. doi: 10.1109/SSPS.2017.8071567.

[11] T. Sorwar, S. B. Azad, S. R. Hussain, and A. I. Mahmood, ”Real-time Vehicle monitoring for traffic surveillance and adaptive change detection using raspberry pi camera module,” in *2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC)*, 2017, pp. 481–484. doi: 10.1109/R10-HTC.2017.8289003.

[12] A. R. Hutauruk, J. Pardede, P. Aritonang, R. F. Saragih, and A. Sagala, ”Implementation of wireless sensor network as fire detector using arduino nano,” in *2019 International Conference of Computer Science and Information Technology (ICoSNIKOM)*, 2019, pp. 1–4. doi: 10.1109/ICoSNIKOM48755.2019.9111537.

[13] L. Chen, J. Zhang, and Y. Wang, ”Wireless car control system based on arduino uno R3,” in *2018 2nd IEEE Advanced Information Management,Communicates,Electronic and Automation Control Conference (IMCEC)*, 2018, pp. 1783–1787. doi: 10.1109/IMCEC.2018.8469286.

[14] P. Bhowmik, P. K. Rout, J. M. Guerrero, and A. Abusorrah, ”Vector measurement-based virtual inertia emulation technique for real-time transient frequency regulation in microgrids,” *IEEE Transactions on Power Electronics*, vol. 36, no. 6, pp. 6685–6698, 2021. doi: 10.1109/TPEL.2020.3034699.

[15] S. Chaudhari, P. Shendge, and S. Phadke, ”Disturbance observer based model following controller for four wheel steering vehicles,” in *2019 Second International Conference on Advanced Computational and Communication Paradigms (ICACCP)*, 2019, pp. 1–6. doi: 10.1109/ICACCP.2019.8883014.

[16] A. Aqthobilrobbany, A. N. Handayani, D. Lestari, Muladi, R. A. Asmara and O. Fukuda, ”HSV Based Robot Boat Navigation System,” 2020 International Conference on Computer Engineering, Network, and Intelligent Multimedia (CENIM), 2020, pp. 269–273. doi: 10.1109/CENIM51130.2020.9297915.

[17] S. Kolkur, D. Kalbande, P. Shimpi, C. Bapat, and J. Jatakia, “Human skin detection using RGB, HSV, and YCbCr color models,” in *Advances in Intelligent Systems Research*, vol. 137, pp. 324–332, 2017. doi: 10.2991/iccasp-16.2017.51.

[18] Y. Lin, H. Zhao, C. Ye, and H. Ding, ”A computationally efficient and robust kinematic calibration model for industrial robots with kinematic parallelogram,” in *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2017, pp. 1334–1339. doi: 10.1109/ROBIO.2017.8324602.

[19] F. B. Setiawan, O. J. Aldo Wijaya, L. H. Pratomo, and S. Riyadi, “Sistem navigasi automated guided vehicle berbasis computer vision dan implementasi pada raspberry pi,” *J. Rekayasa Elektr.*, vol. 17, no. 1, pp. 7–14, 2021. doi: 10.17529/jre.v17i1.18087.

[20] F. B. Setiawan, Y. Yovie, C. Wibowo, L. H. Pratomo, and S. Riyadi, “Perancangan automated guided vehicle menggunakan penggerak motor DC dan motor servo berbasis raspberry pi 4,” *J. Rekayasa Elektr.*, vol. 18, no. 2, pp. 94–101, 2022. doi: 10.17529/jre.v18i2.25863.