

Design and Implementation of Computer Vision Based Autonomous Vehicle Prototype

Emre DANDIL^{1*}, Bilal ARAL²

^{1,2}Department of Computer Engineering, Faculty of Engineering, Bilecik Seyh Edebali University, Bilecik, Turkey

¹<https://orcid.org/0000-0001-6559-1399>

²<https://orcid.org/0000-0002-8756-5850>

*Corresponding author: emre.dandil@bilecik.edu.tr

Research Article

Article History:

Received: 12.12.2021

Accepted: 14.01.2022

Published online: 23.02.2022

Keywords:

Autonomous vehicles
Computer vision
Raspberry Pi
Cascade classifier
Traffic sign recognition
Lane tracking

ABSTRACT

In recent years, since automotive makers, IT providers, commercial electronic chip manufacturers have entered a rapid investment race for autonomous vehicles (AVs), they have started to be used in many infrastructures. AVs are automobiles that can drive themselves using the automatic control systems while cruising by sensing the road, traffic flow and surroundings without the need for a driver. AVs can detect objects around them using technologies such as radar, lidar, GPS, audiometry and computer vision. In this study, the design and implementation of a computer vision-based autonomous vehicle prototype is proposed. The developed prototype can perform lane tracking and traffic sign control by processing images obtained from the camera, and can detect objects around it with the distance sensor. In the AV prototype, the Raspberry Pi 3B+ module is used to process the images and control the motors, and the cascade classifier is used to recognize the traffic signs. In the performed tests, traffic signs are recognized in 7 different scenarios and the performances are compared. According to the results, the accuracy rate is 94,8% in the tests performed with only one traffic sign. As a result, the autonomous vehicle prototype developed in this study can successfully recognize traffic signs and move on the determined route in different scenarios.

Bilgisayarla Görme Destekli Otonom Araç Prototipinin Tasarımı ve Uygulaması

Araştırma Makalesi

Makale Tarihiçesi:

Geliş tarihi: 12.12.2021

Kabul tarihi: 14.01.2022

Online Yayınlanma: 23.02.2022

Anahtar Kelimeler:

Otonom araçlar
Bilgisayarla görme
Raspberry Pi
Kaskad sınıflandırıcı
Trafik işareti tanıma
Şerit takibi

ÖZET

Son yıllarda, otomotiv üreticileri, bilgi teknolojileri sağlayıcıları, ticari elektronik çip üreticileri otonom araçlar (OA) için bir yatırım yarışına girdiklerinden, OA birçok altyapıda kullanılmaya başlamıştır. OA seyir halindeyken, otomatik kontrol sistemlerini kullanarak sürücüye ihtiyaç duymadan yolu, trafik akışını ve çevreyi algılayarak kendi kendine gidebilen otomobillerdir. OA, radar, lidar, GPS, odyometri ve bilgisayarla görme gibi teknolojileri kullanarak etrafındaki nesnelere algılayabilir. Bu çalışmada, bilgisayarla görme tabanlı bir otonom araç prototipinin tasarımı ve uygulaması önerilmektedir. Geliştirilen prototip, kameradan elde edilen görüntüleri işleyerek şerit takibi ve trafik işareti kontrolü gerçekleştirebilmektedir ve mesafe sensörü ile etrafındaki nesnelere algılayabilmektedir. Geliştirilen OA prototipinde görüntüleri işlemek ve motorları kontrol etmek için Raspberry Pi 3B+ modülü ve trafik işaretlerini tanımak için kaskad sınıflandırıcı kullanılmıştır. Yapılan testlerde trafik işaretleri 7 farklı senaryoda tanınmış ve performansları karşılaştırılmıştır. Sonuçlara göre tek trafik işareti ile yapılan testlerde doğruluk oranı %94,8'dir. Sonuç olarak, bu çalışmada geliştirilen otonom araç prototipi, trafik işaretlerini başarıyla tanıyabilmektedir ve farklı senaryolarda belirlenen rotada hareket edebilmektedir.

To Cite: Dandil E., Aral B. Design and Implementation of Computer Vision Based Autonomous Vehicle Prototype. Osmaniye Korkut Ata Üniversitesi Fen Bilimleri Enstitüsü Dergisi 2022; 5(Özel sayı): 50-65.

Introduction

Autonomous vehicle (AV) technology has made rapid progress, especially in recent years, thanks to its advanced hardware and software systems. AVs are capable of fully automatic driving without any human intervention. In these technologies, solutions are sought for problems such as reducing the number of traffic accidents, increasing the mobility of the elderly and disabled, reducing gas emissions into the atmosphere and using infrastructures more effectively (Fagnant and Kockelman, 2015; Van Brummelen et al., 2018). It can be said that one of the important reasons in the development of AV technology is the very high rate of driver-related traffic accidents due to reasons such as fatigue, distraction, and excited driving (Van Brummelen et al., 2018). In addition, it provides a number of benefits such as safety, time efficiency, reducing traffic congestion, efficient use of resources, environmental impacts and increased mobility (Haboucha et al., 2017).

In addition to saving time for drivers to monitor the road continuously, AVs can also provide more effective use of time in activities such as sleeping, eating, relaxing or working while driving. More efficient parking is also possible with these vehicles at the same time. In addition, since a regular traffic flow will be optimized with AVs, it is predicted that with the reduction of traffic congestion, the amount of CO₂ released and fuel consumption will also decrease (Haboucha et al., 2017). Thanks to AVs, it is expected that the total vehicle fleets will decrease significantly, as a result of paving the way for children, the elderly and the disabled to travel by vehicle (Burns, 2013; Anderson et al., 2014; Meyer et al., 2017). Thus, it is clear that AVs will not only transform transportation, but also cause major changes in cities. Today, many vehicle companies are now investing in AVs. It is predicted that AVs will also significantly change the transportation market worldwide (Wiseman, 2021).

In terms of driver assistance systems, AVs are equipped with six different levels as Level 0, Level 1, Level 2, Level 3, Level 4 and Level 5 depending on whether there is any automatic system or fully autonomous driving systems (Gouda et al., 2021; NHTSA, 2021). While there is no autonomous system at Level 0, Level 5 represents vehicles with driverless fully automatic equipment. In recent years, companies such as Google, Tesla, Nissan and Volvo have been producing assistive driver technologies and additional equipment, as well as producing autonomous vehicles. In addition to the rapid growth of AV technologies from 2020, it is estimated that between 2022 and 2025, AVs will be seen in the markets (Fagnant and Kockelman, 2015).

Evaluating the status of AVs today, it was a major milestone for AVs when DARPA (Defense Advanced Research Projects Agency) launched a Grand Challenge in 2004. The goal was for AVs to be able to travel a route of 150 miles. After success for this challenge, the same team performed improvements so that AVs can deal with traffic issues. Especially since 2014, after Google's self-driving cars have come a long way successfully (Anthony, 2014), many vehicle manufacturers have started to work on self-driving vehicles (Fagnant and Kockelman, 2015). In addition, in recent years, the use of AVs has started to become widespread in many different fields such as defense, mining and agriculture.

Since AVs have become more prominent especially in recent years, the rate of research has increased rapidly after 2013. Studies on this subject have generally focused on security, cost, and feasibility for AVs. In a proposed comprehensive study on AVs, (Fagnant and Kockelman, 2015) suggested the use of these tools, opportunities, potential problems, and policies needed (Harper et al., 2016) estimated the increase in travel with AVs for people with disabilities who are unable to travel due to medical conditions (Haboucha et al., 2017) on the other hand, investigated the factors that determine the decisions of users in choosing AVs in their studies. In a study conducted on the cost of AVs (Wadud, 2017), the cost analysis of owning these fully automatic vehicles was evaluated. In another study, (Van Brummelen et al., 2018) they examined current technologies used for AVs. Apart from these, there are also many studies proposed for the AV prototype in the literature (Hossai et al., 2017) proposed an IoT-based AV prototype using computer vision. In another study, (Deac et al., 2018) developed a Raspberry Pi powered miniature AV. For the use of deep learning methods in AVs, (Bechtel et al., 2018) developed a low-cost, deep neural network-based autonomous car (Rossi et al., 2020), on the other hand, developed an AV prototype for real-time detection of road lines. In another study, (El-Tawab et al., 2020) developed an IoT-supported AV prototype, allowing it to move in a certain route.

In this study, a computer vision based AV prototype is developed that can detect lane tracking and traffic sign control, and detect objects around it with a distance sensor, using images obtained from the camera. In this prototype structure, Raspberry Pi 3B+ hardware module is used to process images and control the motors. In the study, cascade classifier is used to detect traffic signs. In the second section of the study, information about the used materials and the proposed methods are presented. In the third section, the experimental results and application practices of the designed AV are explained. In the last section, the findings and the obtained results are evaluated.

Material and Methods

In this study, a computer vision based AV prototype shown in Figure 1 was developed. With the developed AV prototype, traffic signs can be recognized and objects around the vehicle can be detected. There are two infrastructures in the system, namely hardware and software. Images and videos obtained from the camera hardware in AV are processed on the Raspberry Pi module with the developed software infrastructure, enabling the vehicle to tracking the lane accurately on the road and to recognize the traffic signs.

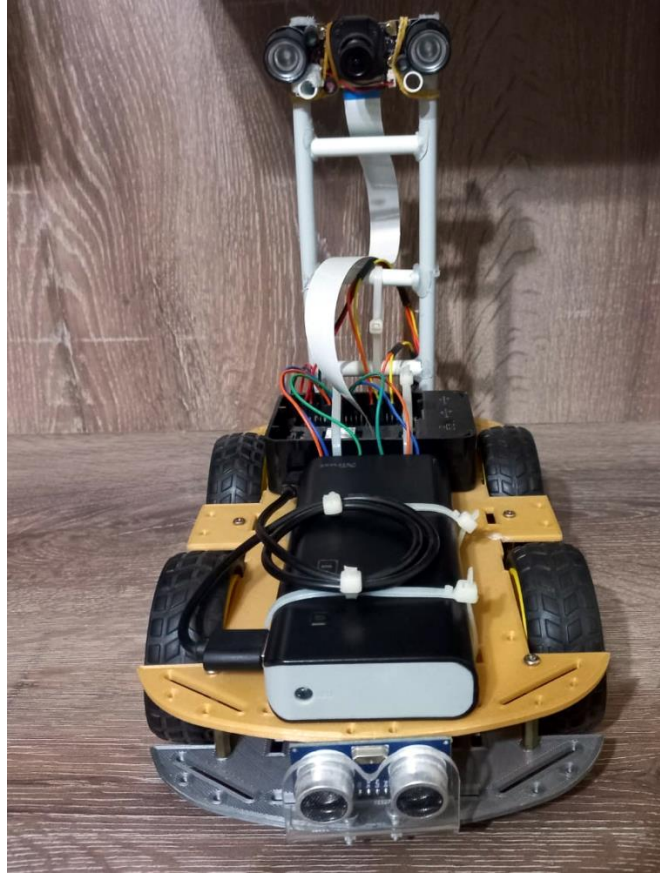


Figure 1. The AV prototype designed in this study

Hardware infrastructure

The representative circuit diagram showing the hardware infrastructure of the AV prototype developed in this study is presented in Figure 2. In the design of the developed prototype, the Raspberry Pi 3B+ model was used as the decision unit to process the images and control the vehicle. LN298N module is used as motor driver circuit. Two HC-SR04 ultrasonic distance sensors were used to detect objects, one to the front and the back of the developed AV prototype. A night vision supported camera module developed for the Raspberry Pi card was used to capture images from the vehicle. After connecting the Raspberry Pi 3B+ on the car kit, DC motors were connected with the LN298N motor driver circuit. 12V DC adapter was used for testing the motors and driver circuit. After the directions of the motors were tested, the power supply connections were made. A 10000 mAh portable charger was used to power the Raspberry Pi board and was connected to the 2.1 A output of the charger. Two 18650 mAh Li-ion batteries were used for the motor driver. For ease of use, two charging modules and a power switch were added to the battery spacer.

Software infrastructure

Using the software infrastructure of the developed AV prototype, the images coming from the camera hardware were processed, so that the recognition of traffic signs and lane tracking of the vehicle were realized successfully. While the Haar Cascade Classifier was used to recognize traffic signs on an

image dataset prepared in the study, the Hough Transform method was used to find the road lines to track the lanes while the vehicle was moving on the road.

Haar Cascade Classifier

Object detection method using Haar feature-based cascade classifiers was proposed by (Viola and Jones, 2001). Viola-Jones algorithm works fast and can reach high accuracy by using rectangular features to identify the object in the image (Dandıl and Özkul, 2019). A dataset is needed to train the Viola-Jones algorithm. The training set has to contain both positive images with the desired object and negative images without it. In order to detect the object in the positive image, as seen in Figure 3, sub-windows of 24*24 pixels are slid across the entire image and an object search is performed. Thus, during the training of the cascade classifier, feature extraction from images is performed using Haar Cascade.

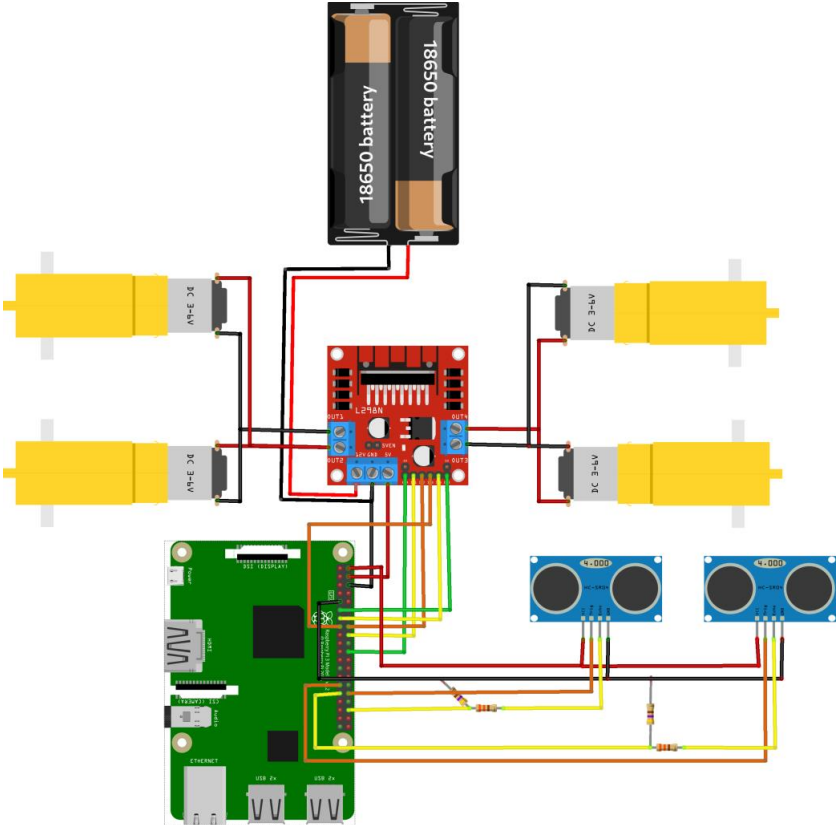


Figure 2. Hardware infrastructure of the developed AV prototype

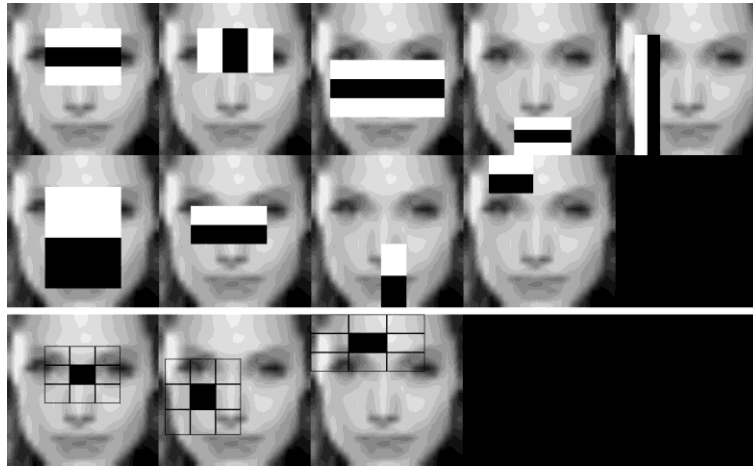


Figure 3. Extracting features from image using Haar Cascade during cascade classifier training (OpenCV, 2021b)

Image Dataset of Traffic Signs

An image dataset was created for the recognition of traffic signs by using the Haar Cascade Classifier on the AV prototype developed in our study. In order for the prototype designed in the study to be recognized correctly, seven different traffic sign models such as right turn, left turn, stop, road work, 30 km/h maximum speed limitation, 50 km/h maximum speed limitation and 80 km/h maximum speed limitation were created for the recognition of traffic signs. The models created for the traffic sign were prepared by sticking the printouts of the traffic signs taken from the Traffic Signs Handbook (KGM, 2015) of General Directorate of Highways (KGM) onto a cardboard material. In addition, the models are mounted on 12 cm long plastic pipes with a plastic cap at the bottom. Figure 4 shows some traffic sign models prepared within the scope of the study.



Figure 4. Some traffic sign models prepared within the scope of the study

The traffic sign dataset was prepared using the images obtained from the above models. For each traffic sign, a dataset consisting of 7000 images in total, consisting of 500 positive images with traffic

signs from different angles and 500 negative images without traffic signs was created. On the obtained images, the parts containing traffic signs were cropped and then sized as 64*64 pixels. Some of the positive images prepared for the stop sign are given in Figure 5.



Figure 5. Stop sign images captured at different angles from the image dataset

Hough Transform

Hough transform is an algorithm developed by (Hough, 1962) and proposed to recognize complex lines in images. Over time, many improvements and new methods for Hough transform have been developed (Mukhopadhyay and Chaudhuri, 2015). In this study, Hough transform was applied to determine the lines of interest in the path route of the AV.

Design and Implementation of AV Prototype

In this study, Haar Cascade Classifier was used to recognize traffic signs. All image processing applications applied to perform lane tracking operations were carried out with programming language (Python, 2021) using (OpenCV, 2021a) and (NumPy, 2021) libraries. The OpenCV library was used for the recognition of traffic signs and lane tracking application used in the AV prototype designed in this study. In the study, the training of the Haar classifier was carried out using Cascade Trainer GUI (Ahmadi, 2021) software on the prepared dataset. In addition, Gentle AdaBoost (GAB) was used to reduce the time loss of scanning in the image.

A path was prepared for the AV prototype developed in the study, as seen in Figure 6. Images of the dataset used for the recognition of traffic signs with the AV prototype were collected on this path. Images were transferred to the Raspberry Pi based module with the camera on the vehicle. As a result of the experimental studies, it was determined that the height of the camera from the ground has to be 24 cm and the angle of the camera with respect to the X axis has to be 18 degrees in order to obtain the images from the camera in the most appropriate way.

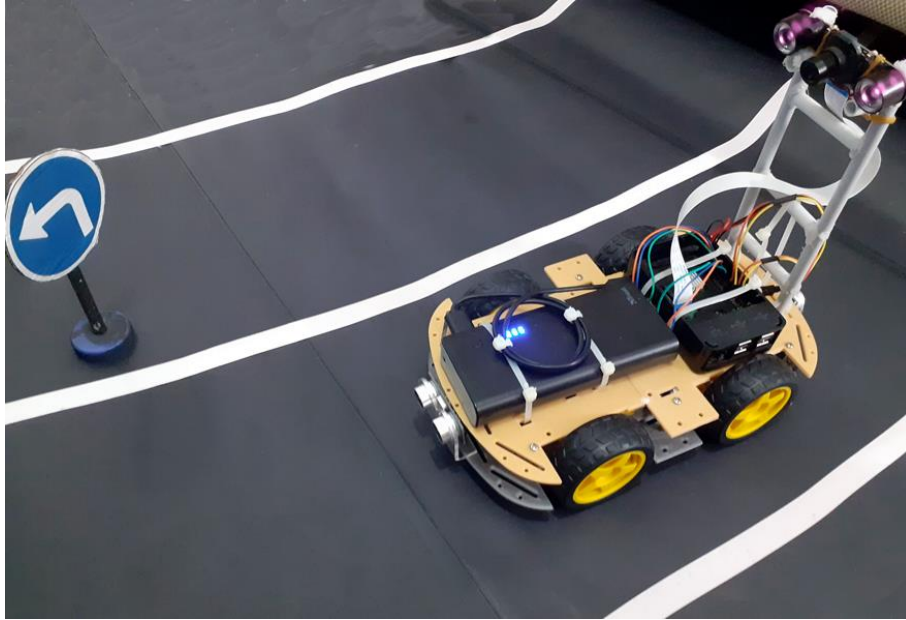


Figure 6. The path designed for the AV prototype developed in this study

In Figure 7, there is a flowchart showing the operation of the designed and implemented AV prototype. As can be seen from this flowchart, motor and sensor connections are adjusted when the system is started. After the system starts to work, a query is made whether there is a traffic sign in the image obtained from the camera. If the traffic sign is not detected on the image and the data coming from the sensors is not less than the determined distance, the following angle is calculated by applying lane tracking operations on the image. The movement of the vehicle is provided according to the scenarios determined by the angle information obtained from here.

Two different computer vision processes are operated in the developed AV prototype. The first of these is to track the lane on the AV route and the other is to recognize the traffic signs with the Haar Cascade Classifier. For lane tracking, finding the edge, extracting the region of interest, determining the straight lines, determining the lane vectors and creating the following vector were applied sequentially. At this stage, firstly, the image obtained from the camera was converted to a gray scale image. Then, according to the light of the environment in which the system was operated, a certain value thresholding process was applied to the related image. The edge detection process was completed with the Canny edge detection algorithm on the black and white image resulting from the thresholding. The edges detected by Canny in some road images obtained from the AV camera are shown in Figure 8.

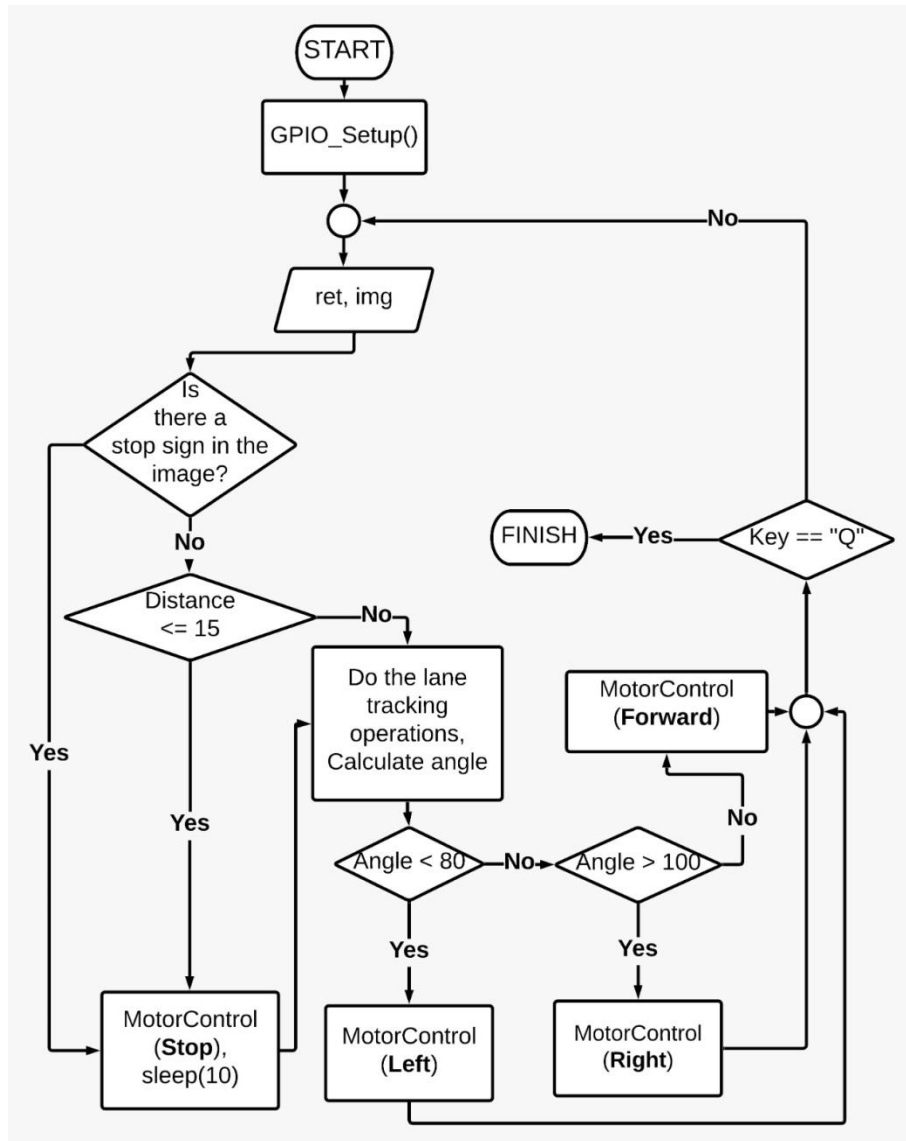


Figure 7. Flowchart diagram of the designed AV prototype



Figure 8. Detecting edges with Canny in some road images obtained from AV camera. (a) RGB image, (b) gray-level image, (c) thresholding and (d) edge detection

It is sufficient to track the lane in a limited area (region of interest), since using the entire image obtained after edge detection operations in lane tracking can tire the system and the lane lines can appear very close to each other due to perspective after a certain distance. In order to determine the region of interest on the image, the appropriate region was extracted by performing a bitwise-and operation with a mask. Removing the region of interest on the in-lane tracking image is shown in Figure 9. After the region of interest was determined, the Hough transform was used to determine the

lines in the region of interest. The determination of the lines in the region of interest with the Hough transform is shown in Figure 10.



Figure 9. Determination of the region of interest on the image for lane tracking

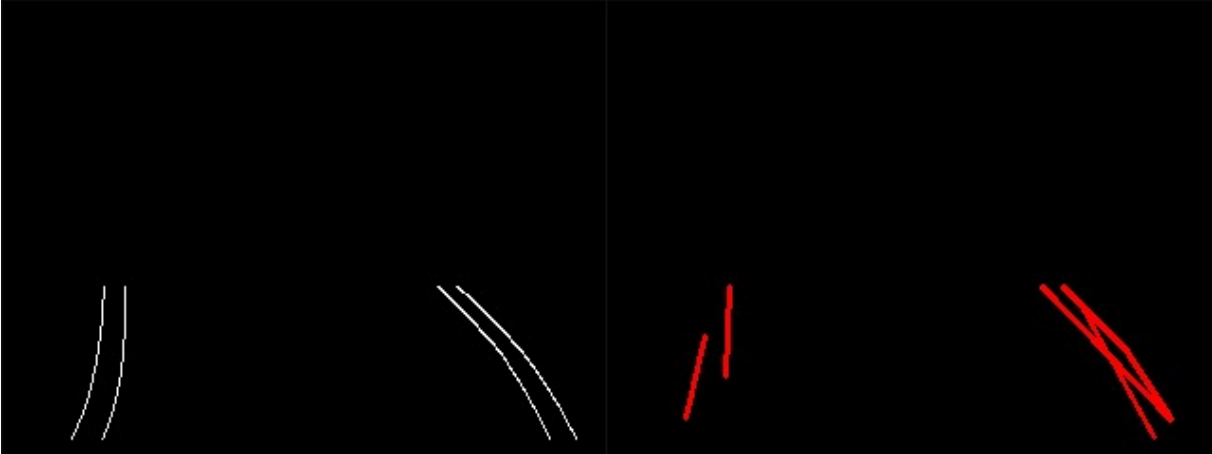


Figure 10. Determination of lines in the region of interest with Hough transform

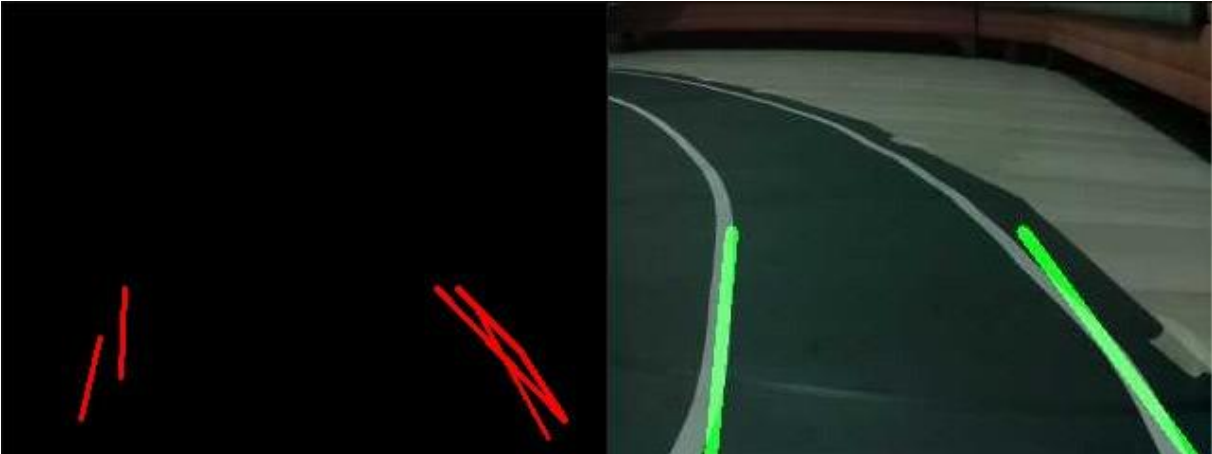


Figure 11. Determination of lane vectors

The slopes of the lines obtained as a result of the Hough Transform were calculated and divided into two separate parts as right and left. By calculating the mean vectors of the two separated line segments, the vectors of the right and left lanes were calculated. Calculated lane vectors are indicated as seen in image Figure 11. By using the obtained lane vectors, the average of the distances of the two vectors to each other was obtained, as in Figure 12, to obtain the tracking vector. After calculating the

slope of the obtained tracking vector, the angle it made with the X axis was calculated. The resulting angle value was used to control the direction of the vehicle. Thus, it was ensured that the AV moves to the right in the range of 0-80 degrees, to the left in the range of 100-180 degrees, and straight in the range of 80-100 degrees.



Figure 12. Determination of the tracking vector

For real-time lane tracking in the developed AV prototype, the steps applied for lane tracking on Raspberry Pi from the images obtained from the camera are shown in Figure 13. In the Raspberry Pi environment shown in the figure, scripts developed in Python programming language for lane tracking were applied for finding the edge, extracting the region of interest, determining the straight lines, determining the lane vectors and creating the following vector operations, respectively.

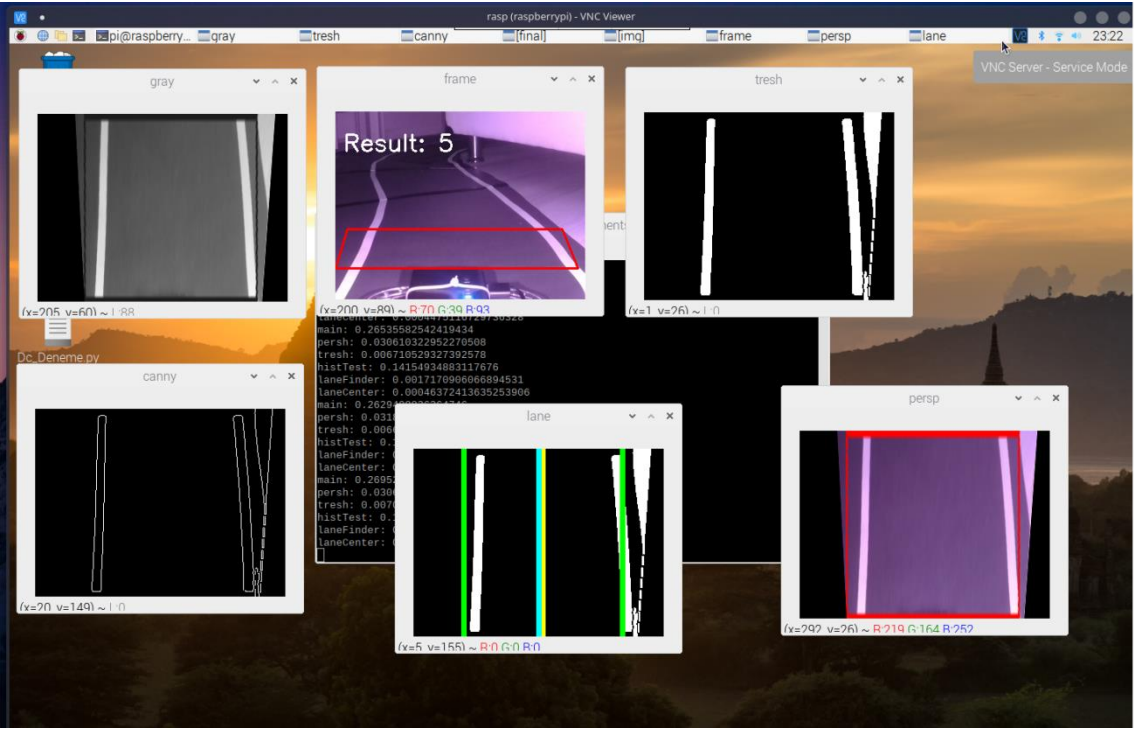


Figure 13. Lane tracking on Raspberry Pi using the developed AV prototype

With the AV prototype designed and implemented in this study, another process performed after lane tracking on the path is the recognition of traffic signs. By using Haar Classifier on the prepared dataset, the Stop traffic sign was successfully recognized as in Figure 14 and the lanes on the path were marked and tracked. With the developed AV prototype, right turn, left turn, stop, road work, 30 km/h maximum speed limitation, 50 km/h maximum speed limitation and 80 km/h maximum speed limitation traffic signs were recognized successfully.



Figure 14. Recognition of 'Stop' traffic sign by using Haar Classifier with AV on the prepared dataset

In the developed AV prototype model, performance tests were carried out for FPS (frame per second) in seven different scenarios (S1, S2, S3, S4, S5, S6 and S7) to recognize seven different traffic signs on the path. In this experimental study, the FPS values obtained from the AV prototype were compared, depending on the number of recognized traffic signs, while the vehicle's lane tracking operations and distance sensors were constantly on and off. Performance test results for recognized traffic signs are shown in Table 1 based on average FPS. According to these findings obtained after the test, it is seen that the vehicle has average FPS values of 6.11 in the best scenario and 1.54 in the worst scenario. Since some of the experiments reached according to the results in this table did not give the expected results, the S7 scenario in which only one traffic sign was detected in other experimental studies was chosen within the scope of our study.

Table 1. Performance test results based on average FPS for traffic signs

Traffic signs	Scenario						
	S1	S2	S3	S4	S5	S6	S7
Road work	on	off	off	off	off	off	off
Max 30 km/h speed limitation	on	on	off	off	off	off	off
Max 50 km/h speed limitation	on	on	on	off	off	off	off
Max 80 km/h speed limitation	on	on	on	on	off	off	off
Right turn	on	on	on	on	on	off	off
Left turn	on	on	on	on	on	on	off
Stop	on	on	on	on	on	on	on
FPS (average)	1.54	1.72	1.93	2.27	2.94	3.83	6.11

In another experimental study carried out for the AV designed in this study, experimental studies were performed on a test path with a length of approximately 3.5 m and a slope of 30 degrees in the horizontal plane, in a scenario consisting of two stop signs and an obstacle, by including 10 seconds of waiting time at each obstacle and stop sign. In this test, it was observed that the vehicle completed the path in approximately 86 seconds. In order to measure the performance of the developed AV to recognize traffic signs, 1000 video frames of traffic signs from different angles were obtained from the camera and tested in real time. Different traffic signs in 948 images out of 1000 images tested were successfully recognized by the developed AV. Accordingly, the success of the developed AV in recognizing traffic signs were calculated with an accuracy of 94.8%.

Conclusion

In this study, a computer vision based AV prototype was designed and implemented. Using the highly successful Haar Cascade Classifier, the traffic signs were recognized, and the lane tracking of the vehicle was successfully achieved by using image processing techniques on the images obtained from the camera. In addition, it was achieved that the vehicle can cruise safely with the distance sensors.

In the experimental studies, traffic signs were recognized in seven different scenarios and the performance of the AV was compared in terms of FPS. According to the findings, it has been seen that the Raspberry Pi board alone cannot meet the desired needs in real-time image processing theologies. Experimental studies with different up-to-date equipment are expected to be more successful. In this carried out study, in experimental studies with only one traffic sign, recognition of traffic signs was achieved with an accuracy of 94,8%. In addition, it has been observed that the light level of the environment where the images are obtained is a major factor in the detection of traffic signs and the success of lane tracking operations. In the study, it is predicted that if a more powerful hardware is used instead of the used Raspberry Pi card, the accuracy performance and real-time operating speed can be higher. In future studies, it is planned to recognize more traffic signs on a more advanced AV prototype in terms of hardware and software.

Acknowledgment

A part of this study was presented as a Project Study at Department of Computer Engineering, Faculty of Engineering, Bilecik Seyh Edebali University.

Statement of Conflict of Interest

Authors have declared no conflict of interest.

Author's Contributions

The contribution of the authors is equal.

*This study was presented as a summary paper at the International Conference on Engineering, Natural and Applied Sciences (ICENAS'21) held online on 24-26 November 2021.

References

- Ahmadi A. Cascade classifier GUI. Available online: <https://amin-ahmadi.com/cascade-trainer-gui/>. (Accessed on: 29.01.2021)
- Anderson JM., Nidhi K., Stanley KD., Sorensen P., Samaras C., Oluwatola OA. Autonomous vehicle technology: A guide for policymakers. 2014; Rand Corporation.
- Anthony S. Google's self-driving car passes 700,000 accident-free miles, can now avoid cyclists, stop at railroad crossings. 2014; Available online: <https://www.extremetech.com/extreme/181508-google-self-driving-car-passes-700000-accident-free-miles-can-now-avoid-cyclists-stop-for-trains>. (Accessed on: 20.10.2021)
- Bechtel MG., McEllhiney E., Kim M., Yun H. Deeppicar: A low-cost deep neural network-based autonomous car. 2018 IEEE 24th international conference on embedded and real-time computing systems and applications (RTCSA), 28-31 August 2018, pp:11-21, Hakodate, Japan.
- Burns LD. A vision of our transport future. Nature 2013; 497(7448): 181-182.
- Dandıl E., Özkul İ. Futbol maçları için bilgisayarlı görü destekli gol karar sistemi (golkasis): Bir prototip çalışma. Gazi Üniversitesi Fen Bilimleri Dergisi Part C: Tasarım ve Teknoloji 2019; 7(1): 213-224.
- Deac MA., Al-doori RWY., Negru M., Dănescu R. Miniature autonomous vehicle development on raspberry pi. 2018 IEEE 14th International Conference on Intelligent Computer Communication and Processing (ICCP), 6-8 September 2018, pp: 229-236, Cluj-Napoca, Romania.
- El-Tawab S., Sprague N., Mufti A. Autonomous vehicles: Building a test-bed prototype at a controlled environment. 2020 IEEE 6th World Forum on Internet of Things (WF-IoT), 2-16 June 2020, pp: 1-6, New Orleans, LA, USA.

- Fagnant DJ., Kockelman K. Preparing a nation for autonomous vehicles: opportunities, barriers and policy recommendations. *Transportation Research Part A: Policy and Practice* 2015; 77: 167-181.
- Gouda M., Chowdhury I., Weiß J., Epp A., El-Basyouny K. Automated assessment of infrastructure preparedness for autonomous vehicles. *Automation in construction* 2021; 129: 103820.
- Haboucha CJ., Ishaq R., Shiftan Y. User preferences regarding autonomous vehicles. *Transportation Research Part C: Emerging Technologies* 2017; 78: 37-49.
- Harper CD., Hendrickson CT., Mangones S., Samaras C. Estimating potential increases in travel with autonomous vehicles for the non-driving, elderly and people with travel-restrictive medical conditions. *Transportation Research Part C: Emerging Technologies* 2016; 72: 1-9.
- Hossain MRT., Shahjalal MA., Nuri NF. Design of an IoT based autonomous vehicle with the aid of computer vision. 2017 International Conference on Electrical, Computer and Communication Engineering (ECCE), 16-18 February 2017, pp: 752-756, Cox's Bazar, Bangladesh.
- Hough PV. Method and means for recognizing complex patterns. 1962; Google Patents.
- KGM. Karayolları Genel Müdürlüğü (KGM) Trafik İşaretleri El Kitabı. 2015; Available online: <https://www.kgm.gov.tr/SiteCollectionDocuments/KGMdocuments/Trafik/IsaretlerElKitabi/TrafikIsaretleriElKitabi2015.pdf>. (Accessed on: 29.10.2020)
- Meyer J., Becker H., Bösch PM., Axhausen KW. Autonomous vehicles: The next jump in accessibilities?. *Research in transportation economics* 2017; 62: 80-91.
- Mukhopadhyay P., Chaudhuri BB. A survey of hough transform. *Pattern recognition* 2015; 48(3): 993-1010.
- NHTSA. National Highway Traffic Safety Administration-Automated Vehicles for Safety. Available online: <https://www.nhtsa.gov/technology-innovation/automated-vehicles-safety>. (Accessed on: 20.10.2021)
- NumPy. Available online: <https://numpy.org/doc/stable/>. (Accessed on: 04.04.2021)
- OpenCV. Available online: <https://opencv.org/>. (Accessed on: 29.01.2021)
- OpenCV. Cascade classifier training. Available online: https://docs.opencv.org/4.5.4/dc/d88/tutorial_traincascade.html. (Accessed on: 29.01.2020)
- Python. Available online: <https://www.python.org>. (Accessed on: 29.01.2021)
- Rossi A., Ahmed N., Salehin S., Choudhury TH., Sarwar G. Real-time lane detection and motion planning in raspberry pi and arduino for an autonomous vehicle prototype. arXiv preprint arXiv:2009.09391 2020.
- Van Brummelen J., O'Brien M., Gruyer D., Najjaran H. Autonomous vehicle perception: The technology of today and tomorrow. *Transportation Research Part C: Emerging Technologies* 2018; 89: 384-406.

- Viola P., Jones M. Rapid object detection using a boosted cascade of simple features. 2001 IEEE computer society conference on computer vision and pattern recognition (CVPR 2001), 8-14 December 2001, pp: I-511-I-518, Kauai, HI, USA.
- Wadud Z. Fully automated vehicles: A cost of ownership analysis to inform early adoption. *Transportation Research Part A: Policy and Practice* 2017; 101: 163-176.
- Wiseman Y. Autonomous vehicles, *Encyclopedia of Information Science and Technology*, Fifth Edition 2021; IGI Global, pp:1-11.