Developing A Real Time Autonomous Car with Raspberry pi

Dr. B. Prasada Rao
Department of Electronics and Communication Engineering
Vignan's Institute of information Technology, Visakhapatnam, India
bvvprasadarao@gmail.com

K. Jahnavi
Department of Electronics and Communication Engineering
Vignan's Institute of Information Technology, Visakhapatnam, India
jahnavikavala@gmail.com

K. Naveen
Department of Electronics and Communication Engineering
Vignan's Institute of Information Technology, Visakhapatnam, India
eswarnani1427@gmail.com

K. Dhanunjay
Department of Electronics and Communication Engineering
Vignan's Institute of Information Technology, Visakhapatnam, India
dhanunjaynaidu5@gmail.com

K. Uday Kiran
Department of Electronics and Communication Engineering
Vignan's Institute of Information Technology, Visakhapatnam, India
urstrulyuday29@gmail.com

***Abstract*— The advent of vehicles capable of operating without human intervention has revolutionized the automotive sector, offering the prospect of safer and more efficient transportation systems. Now a days, accidents are increasing and there is no specific process for the implementation of road safety. In this project, we propose an autonomous driving system implemented on a Raspberry Pi platform, autonomous car, lane detection, functionalities. The system utilizes a Raspberry Pi, paired with a Pi the camera module has the ability to capture images videos with ease for real-time image capture and processing. Lane detection is performed using image processing techniques to detect lane markings on the road, enabling the vehicle to stay within its lane and navigate safely. The proposed system aims to provide a comprehensive autonomous driving solution that can be implemented on low-cost hardware and lightweight deep learning models, making it accessible for research, education, and prototyping purposes. The system demonstrates the potential for autonomous vehicles are designed to navigate safely and intelligently on their own real-world environments.**

Keywords— ***Autonomous driving, Raspberry Pi, Lane detection, Real-time system, Open CV.***

I. INTRODUCTION

Autonomous vehicles, commonly referred to as self-driving or driverless cars, epitomize a revolutionary leap in transportation technology. Possessing the ability to navigate roads, interpret their environment, and autonomously make driving decisions, reducing the reliance on human intervention. Throughout the last decade, autonomous cars have witnessed significant development and become a focal point of innovation and investment across various industries, from automotive giants to tech startups. Develop reliable cars, safer, most responsive for customers of the future generation. The concept of autonomous cars has captured public imagination and generated significant excitement due to their potential for a revolution in commuting, travel, and interactions with transportation systems. With the progress in artificial intelligence and computing capabilities, autonomous vehicles are swiftly transitioning from experimental models to practical solutions for addressing real-world transportation issues.

The rise of autonomous cars is propelled by a combination of factors, including concerns regarding road safety and environmental sustainability. Through the elimination because they eliminate human error, a significant factor in accidents, autonomous vehicles hold the promise of significantly reducing road fatalities and injuries.

Additionally, self-driving cars can enhance traffic flow, minimize commute durations, and improve overall transportation efficiency. In recent times, significant investments in research and development by major automotive manufacturers, technology firms, and startups have been directed towards bringing autonomous cars to market. These endeavours have resulted in the creation of advanced sensors like LiDAR, radar, and cameras, enabling autonomous vehicles to accurately perceive and interpret their surroundings with a high degree of accuracy.

Additionally, advances in machine learning and artificial intelligence algorithms empower autonomous cars to make complex decisions in real-time, adapting to changing road condition and unforeseen obstacles.

Self-driving cars offer enhanced mobility to those who cannot drive due to reasons such as age, disability, or other limitations. By offering a reliable and accessible transportation option, autonomous vehicles improve access to essential services, employment opportunities, and social activities for a broader range of people. Autonomous driving technology allows passengers to utilize the time previously spent driving for other activities such as work, relaxation, or entertainment. This newfound productivity improves individuals' quality of life and supports economic growth. The development of autonomous cars drives innovation in technology, manufacturing, and urban planning. By pushing the boundaries of AI, robotics, and sensor technology, self-driving vehicles pave the way for new advancements that benefit society.

An autonomous car employing Raspberry Pi and OpenCV harnesses the processing power of Raspberry Pi's hardware and the computer vision capabilities of OpenCV to navigate its surroundings. The system begins by integrating various sensors, such as cameras, pi Cam to capture environmental data. OpenCV then processes the visual data obtained from these sensors, employing algorithms for task like lane detection. Using OpenCV, the car identifies and interprets objects in its path, including pedestrians, vehicles, and road markings. It utilizes this information to generate a real-time understanding of the environment, creating a virtual map to navigate through. This mapped data informs decision-making algorithms on path planning, route optimization, and collision avoidance.

This paper presents a practical and economical strategy for autonomous driving using a lightweight deep learning model. The model efficiently functions on devices with limited resources, facilitating real-time processing. A small-scale vehicle equipped with scalar and vision sensors and powered by a Raspberry Pi serving as the central processing unit, it navigates autonomously along a predefined route. Following points outline the operational framework of this project

1. A Driver-less model car with raspberry pi is preferred, it has some tasks:

a. The Driver-less car model autonomously navigate along a predetermined track and possesses the ability to move in these directions: forward, left, turn right, backward and stop.

b. The model car employs detection and recognition algorithms to identify different task like lane detection enabling it to respond appropriately based on their meanings and instructions.

c. Lane detection plays a vital role in autonomous cars systems, allowing vehicles to navigate securely within designated lanes. Our approach involves real-time image processing techniques to detect lane markings from a video feed captured by a Raspberry Pi camera module.

2. We utilize the Canny edge detection algorithm to enable the robot to move independently. This method enables the robot to perceive its surroundings and navigate autonomously.

II. LITERATURE SURVEY

Rahul P. Kharapkar in 2020 discussed an IoT-based self-driving car utilizes sensors and connectivity to navigate autonomously. It employs technologies like GPS, cameras, and lidar for environment perception. The car communicates with a central server or cloud platform to access real-time data and updates. Machine learning algorithms analyse sensor data to make decisions on steering, acceleration, and braking. Because they eliminate human error, a significant factor in accidents, autonomous vehicles hold the promise of significantly reducing road fatalities and injuries. Applications range from urban mobility solutions to delivery services and ride-sharing platforms. The car's performance relies on network reliability, sensor accuracy, and algorithm robustness. Overall, it represents a cutting-edge integration of IoT and autonomous driving technologies [1].

Ming-Han Lee Presented the design of an autonomous and manual driving system for a 4WIS4WID vehicle incorporates both human-controlled and automated driving capabilities. The system incorporates various sensors like cameras, lidar, and radar to perceive the environment. Machine learning algorithms analyse the sensor data to facilitate autonomous navigation, taking into account aspects such as obstacle avoidance and route planning. The system includes a user interface for manual driving mode, allowing drivers to take control when desired. It ensures seamless transition between autonomous and manual modes for enhanced flexibility and safety. Applications span from personal vehicles to industrial and military applications. Performance hinges on sensor accuracy, algorithm efficiency, and user interface intuitiveness. Overall, it presents an advanced solution for versatile and efficient transportation [2].

Anindtiya Anggari Nuryono in 2020 discussed the robotic vision is a robot utilizes a camera to track lines utilizing image processing methods, the system identifies the path on the ground and adjusts its movement accordingly. The robot's onboard software analyses the camera feed to stay centered on the line. It employs algorithms like PID control to regulate its speed and direction. This system allows the robot to follow predefined paths accurately. It's commonly used in educational settings for teaching robotics and computer vision concepts. The robot's performance depends on factors like lighting conditions and line contrast. Overall, it's an effective method for implementing autonomous navigation in simulated environments [3].

Sagar Shetty in 2019 discussed a line-following robot using image processing employs a camera to capture the robot's path. It processes the images to detect lines and determines the robot's position relative to the path. Algorithms like PID control are often this data is utilized to manage the robot's motion. By analysing pixel data, the robot adjusts its direction to stay on course. This method is commonly used in robotics competitions and industrial automation for precise navigation tasks. It's an efficient way to implement autonomous guidance systems without relying on physical sensors. The robot's performance depends on factors such as image quality and processing speed. Overall, it offers a versatile solution for various line-following applications [4].

Chun-Fei Hsu in 2018 discussed the control system for a two-wheel self-balancing robot utilizes vision-based line-following, incorporating a camera for real-time path detection. Image processing algorithms analyse the camera feed to identify the line's based on this data, the robot modifies its wheel speeds to maintain its position relative to the path. Techniques like PID control are commonly employed to maintain stability and accuracy in movement. This approach eliminates the need for physical sensors on the ground, enhancing flexibility and adaptability. It finds applications in areas such as warehouse automation and autonomous transportation. Performance is influenced by factors like lighting conditions and camera resolution. Overall, it offers a sophisticated solution for precise navigation tasks in diverse environments [5].

Reza Javanmard Alitappeh presented a line-following autonomous driving robot using deep learning utilizes a neural network to process camera input for line detection. The network learns to identify the path and control the robot's movement based on visual cues. Deep learning algorithms, which include convolutional neural networks, they are frequently utilized for this purpose. The robot's onboard system continuously analyzes the camera feed to maintain alignment with the line. This approach offers flexibility and adaptability to various environments without the need for predefined rules. It finds applications in fields like agriculture, logistics, and automotive industries. Performance depends on factors such as training data quality and neural network architecture. Overall, it presents a sophisticated solution for robust and precise autonomous navigation [6].

Daping Jin presented a reliable autonomous tracking technique for mobile robots navigating dynamic surroundings. employs advanced algorithms to track and follow moving targets effectively. It utilizes sensor data fusion from multiple sources such as cameras, lidar, and GPS to ensure accurate perception of the surroundings. Machine learning techniques enable the robot adjusts to environmental changes and anticipates the movements of the target. The system prioritizes safety by incorporating collision avoidance strategies and real-time path planning. It offers seamless navigation even in complex and unpredictable scenarios, enhancing efficiency and reliability. Applications include surveillance, logistics, and search and rescue missions. Performance depends on the accuracy of sensor data fusion and the robustness of the prediction algorithms. Overall, it provides a sophisticated solution for autonomous following in dynamic environments [7].

III. METHODOLOGY

Designing a methodology for lane detection in autonomous cars using OpenCV and image processing involves several key steps to ensure accurate and reliable performance in real-world environments. This methodology integrates various techniques to detect and track lanes on the roadway, facilitating safe and efficient vehicle navigation.

Firstly, preprocessing of input images is essential to enhance the quality and clarity of the captured frames. Techniques such as colour space transformation, noise reduction, and contrast enhancement can be applied to improve the visibility of the lines and reduce the impact of environmental factors such as shadows and glare.

Once pre-processed, the image is then segmented to isolate the region of interest containing the road markings. This is typically achieved using techniques like thresholding, which converts the grayscale image into a binary representation where pixels are classified as either belonging to the line or background based on their intensity values.

Following segmentation, edge detection techniques such as the Canny edge detector are utilized to detect potential line segments within the image. This detector identifies regions of notable intensity changes, effectively outlining the edges of objects present in the scene, such as road markings.

Subsequently, the detected edges are further processed to extract lines using techniques such as the Hough transform is employed. It converts the Cartesian representation of lines ($y = mx + b$) into a parametric in the realm of space, lines are defined by their slope and intercept. This allows for robust detection of lines even in the presence of noise and gaps in the edge map.

Once lanes are detected, filtering and refinement steps are applied to eliminate false positives and improve the accuracy of the line detection. This may involve criteria such as lane length, orientation, and proximity to other detected lines. Additionally, techniques like line fitting can be used to model the detected line segments as continuous lanes, providing smoother and more consistent representations of the road markings.

Furthermore, to enhance robustness in dynamic environments, the methodology may incorporate adaptive techniques such as Kalman filtering or Bayesian inference to predict the trajectory of the detected lines over time. This enables the autonomous car to anticipate changes in the road layout and adjust its path accordingly.

Finally, the detected lanes are used to generate control commands for steering the vehicle along the desired path. By analysing the position, curvature, and continuity of the detected lines, the autonomous car can determine the appropriate steering angle to follow the road markings accurately while maintaining a safe trajectory.

In summary, the methodology for line detection in autonomous cars using OpenCV and image processing involves preprocessing, segmentation, edge detection, line extraction, filtering, refinement, and predictive modelling to enable robust and reliable autonomous navigation in dynamic environments. Through the integration of these techniques, the autonomous car can effectively detect and track road markings, facilitating safe and efficient driving experiences.
The conclusion highlights the successful integration of hardware and software components, marking a significant step towards autonomous navigation.

IV. PROPOSED MODEL

A. Block diagram of Proposed Model

The block diagram represents a lane detection autonomous car using Raspberry Pi outlines the essential components and their interconnections in the system in Fig 1. The Raspberry Pi board serves as the core processor, orchestrating the autonomous functions of the vehicle. The Raspberry Pi connects with different sensors, such as a mounted camera module on the vehicle, to capture images of the road ahead.

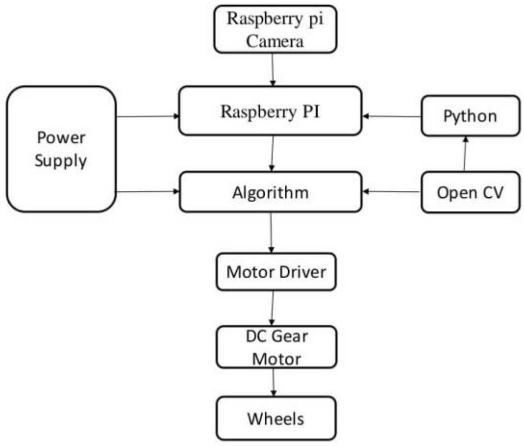

Fig 1: Block Diagram of lane detection autonomous car using raspberry pi

These images are then processed using image processing algorithms implemented in Python, leveraging libraries like OpenCV. The processed data is fed into the lane detection module, which identifies and tracks the lanes on the road. This information is then used to generate control signals for steering the car autonomously. Additionally, the system may incorporate other modules for obstacle detection, navigation, and additionally, it facilitates communication with external devices or networks. In summary, the block diagram demonstrates the amalgamation of hardware and software elements to achieve autonomous lane detection and navigation in the Raspberry Pi-based vehicle.

Raspberry pi

The Raspberry Pi 4 is renowned for its compact design and remarkable capabilities, making it a standout choice among single-board computers for its versatility. Developed by the Raspberry Pi Foundation, it boasts a quad-core ARM Cortex-A72 processor, delivering substantial performance enhancements compared to earlier models. Available with 2GB, 4GB, or 8GB of RAM options, it suits a broad spectrum of uses, ranging from personal projects to industrial automation. Featuring USB 3.0 ports, dual-band Wi-Fi, Bluetooth 5.0, and gigabit Ethernet are all featured. Offers extensive connectivity possibilities. Its HDMI and DisplayPort interfaces enable dual-display configurations, rendering it well-suited for multimedia tasks and desktop computing purposes.

Motor Driver

An electronic device known as a motor driver is tasked with controlling the speed, direction, and torque of an electric motor. It translates signals transmitted from a microcontroller or control unit system and supplies the required power to operate the motor. Motor drivers fulfil a vital function across various fields, including robotics, automation, and the realm of electric vehicles.

DC Gear Motor

An electronic component known as a motor driver component designed to manage the velocity, orientation, and rotational force of an electric motor. It receives and interprets signals from a microcontroller or control system, delivering the required power to operate the motor. Motor drivers are critical elements utilized across a wide array of applications, spanning from robotics and automation to electric vehicles.

PI Camera

A camera is a device that utilizes optics to capture and record images or videos. It includes a lens that focuses light onto a photosensitive surface, such as a digital sensor or film. Cameras are used in various devices, including smartphones, digital cameras, surveillance systems, and scientific instruments, for capturing visual information.

Power Supply

A power supply functions as a tool or apparatus it converts input voltage from a source into a specific output voltage, current, or power level suitable for powering electronic devices. It typically regulates and distributes power to ensure stable and reliable operation of electronic equipment, ranging from small gadgets to industrial machinery a power supply ensures stable and reliable operation for electronic equipment, catering to a wide range of applications from small gadgets to industrial machinery. It converts input voltage to provide specific output levels of voltage, current, or power.

B. Software tools

Algorithm

The Canny edge detector is an image processing algorithm used for edge detection. It involves multiple steps, the process involves smoothing, calculating gradients, implementing non-maximum suppression, and applying hysteresis thresholding. By identifying abrupt changes in pixel intensity, it accurately detects edges in images, making it extensively utilized in tasks related to computer vision and image processing.

Open CV

OpenCV, an open-source Computer Vision Library (CVL), offers a suite of programming functions primarily designed for real-time applications within computer vision. It furnishes a range of tools and algorithms for analysing images and videos, encompassing tasks such as tasks encompass image processing, object detection, machine learning, and deep learning are among its core functionalities. Its versatility and functionality render it extensively utilized across diverse applications and research domains.

C. Software description

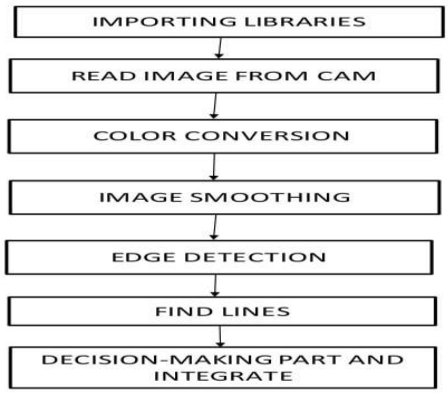

Fig 2: Flow chart of Program Flow in Open CV

The Fig 2 depicts a flowchart outlining the fundamental steps in basic image processing. It commences with importing libraries, which are most likely software tools essential for image processing. These specific libraries would depend on the chosen programming language and the desired image processing tasks.

Following this, the flowchart shows reading the image from the camera (CAM). This step could involve capturing a new image or opening an existing image file.

Next, the image might undergo a colour conversion stage. Images can be stored in various colour formats, and converting the image to a more suitable format for processing can be crucial. For instance, an image captured from a camera might be converted from RGB (red, green, blue) to grayscale for further processing.

Image smoothing is then applied. This step helps reduce noise or blurriness within the image. Noise can be introduced during image acquisition or transmission, and smoothing helps to mitigate this effect.

Following smoothing, the flowchart depicts edge detection. Edges in images represent the boundaries

between areas with different brightness or colour. Detecting these edges can be useful for various image processing applications, such as object recognition.
There is a decision-making part after edge detection. The criteria for the decision are not shown in the image, but this step likely involves analysing the features extracted during the previous steps (e.g., smoothed image, detected edges) to determine how to proceed further. For instance, the decision might be to classify the objects present in the image or to perform further analysis based on the edge characteristics.
Finally, the process concludes with an integration step. The purpose of integration is not depicted in the flowchart, but it likely involves combining the results of the previous steps to achieve the desired outcome. For example, the integration might involve combining the classified objects or edge information into a single data structure or using it to make a final decision about the image content.

IV. RESULTS AND DISCUSSION

The project explores lane detection within an autonomous car utilizing Raspberry Pi, illustrating both the system's efficacy and its inherent limitations. The results typically include metrics such as accuracy, precision, and computational efficiency. These metrics assess the system's ability to detect lane markings accurately under various conditions like different lighting, road surfaces, and weather.

Discussion usually revolves around the performance trade-offs and areas for improvement. For instance, the system may perform well under normal lighting conditions but struggle in low-light environments. The discussion may also cover the impact of computational resources on real-time processing, as Raspberry Pi has limited computing power compared to dedicated hardware.
Step 1: Importing libraries and read image from cam. Raspberry pi camera module gives input to the raspberry pi. The given input will be in the form of video, then video is converted to images are shown in Fig 3.

Fig 3: Raspberry pi camera captured image

Step 2: colour conversion, converting an image from RGB to grayscale is necessary, especially for smaller images, to simplify the data representation. RGB images contain three colour channels (red, green, and blue), resulting in high-dimensional data is shown in Fig 4.

Fig 4: RGB into grayscale converted image

Step 3: Image smoothing and edge detection. Artificial intelligence is employed to recognize these edges automatically. The Canny edge detection algorithm is frequently employed for edge detection tasks. It works by identifying significant changes in intensity within the image, thereby highlighting edges effectively shown in Fig 5.

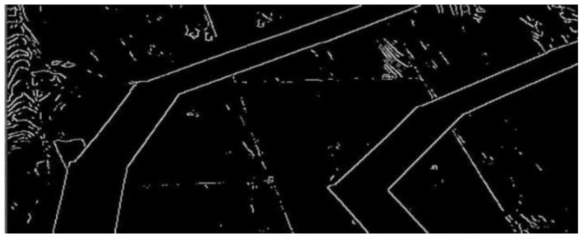

Fig 5: Edge detection

Step 4: Find lines, Following the detection phases, the robot employs machine learning technology to analyse the collected data and decide on the appropriate direction to turn, based on the identified edges and lines. This step involves determining whether a left or right turn is necessary, according to the patterns recognized from the previous detections.

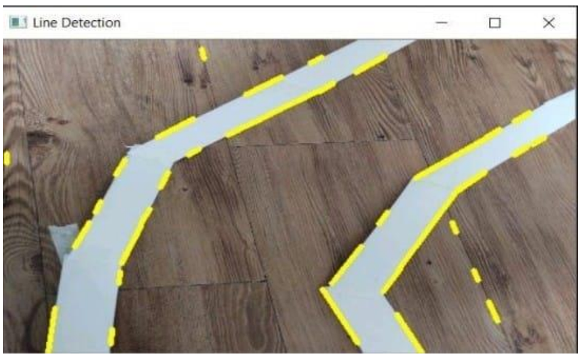

Fig 6: line detection

Step 5: Next, a threshold value is established by comparing the difference between the threshold and theta values. Once this threshold is determined, a command is transmitted to the raspberry pi, instructing it to drive the motor in the specified direction has shown in Fig 7.

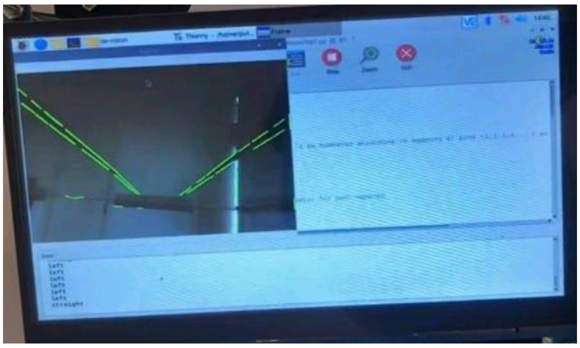

Fig 7: Deciding whether to turn left or straight

Step 6: Integrate, Subsequently, the robot proceeds towards its intended destination using the implemented algorithms. Along the way, it continuously scans for obstacles to ensure a safe path forward has shown in Fig 8.

After executing the algorithms, the robot advances towards its target location. During this movement, it actively scans its surroundings for any obstructions or hazards obstructing its path. If any obstacles are detected, the robot adjusts its trajectory accordingly to avoid collisions. This ongoing detection and adjustment process ensures smooth and safe navigation towards the desired destination. Additionally, the robot remains vigilant, continuously assessing its environment to pre-emptively address any potential obstacles that may arise.

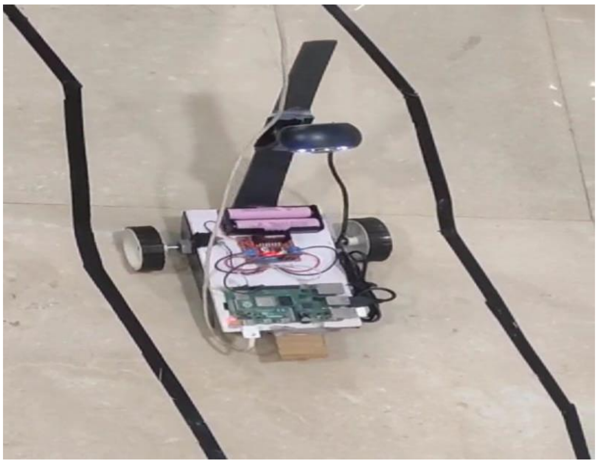

Fig 8: Model moving through the path to destination

V. CONCLUSION

The lane detection system implemented on a Raspberry Pi platform signifies a significant advancement in autonomous vehicle technology. Through the fusion of computer vision and machine learning algorithms, the system accurately identifies lane markings and navigates the vehicle safely. Its affordability and versatility make it accessible for further research and development. Despite challenges like varying road conditions, the system's capability to detect obstacles enhances safety. This innovation marks a crucial step towards the realization of fully autonomous vehicles, promising safer and more efficient transportation systems for the future.

VI. FUTURE SCOPE

Future advancements in lane detection for autonomous cars using Raspberry Pi may include improvements in robustness, adaptability to diverse road conditions, integration of advanced sensors, real-time decision-making capabilities, and collaboration with industry partners for widespread adoption, ensuring safer and more efficient transportation systems.

REFERENCES

[1] Rahul P. Kharapkar, Abhishek S. Khandare, Zeeshaan W. Siddiqui, “IoT based Self Driving Car”, International Research Journal of Engineering and Technology (IRJET), Vol. 07, No. 3, pp: 5177-5181, March 2020.

[2] Tzuu-Hseng S. LI, Ming-Han Lee, Chia-Wei Lin, Guan-Hong Liou, Wei-Chung Chen, “Design of Autonomous and Manual Driving System for 4WIS4WID Vehicle”, IEEE Access, Vol. 4, pp: 2256 – 2271, March 2016.

[3] Alfian Ma’arif, Aninditya Anggari Nuryono, Iswanto, “Vision-Based Line Following Robot in Webots”, International Conference on Electrical Engineering (FORTEI- ICEE), September 2020.

[4] J. Sarwade, S. Shetty, A. Bhavsar, M. Mergu, and A. Talekar, “Line following robot using image processing,” in 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC), 2019.

[5] Chun-fei Hsu, Chien-ting su, Wei-fu Kao, Bore-Kuen lee, “Vision-based line- following control of a two-wheel self-balancing robot”, International Conference on Machine learning and Cybernetics, China, July 2018, doi:10.1109/icmlc.2018.8526952.

[6] Reza Javanmard Alitappeh, Amir Hossein Zabbah, Mohammad Karimi, Abdolmaleki, Kassar Jeddisaravi, “Line Following Autonomous Driving Robot using Deep Learning”, 6th Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS), December 2020. doi:10.1109/icspis51611.2020.9349547.

[7] Daping Jin, Zheng Fang, Jiexin Zeng, “A Robust Autonomous Following Method for Mobile Robots in Dynamic Environments”, IEEE Access, Vol. 8, pp: 150311 - 150325, August 2020.

[8] Yasir A, Aiman Salim, Arya Dileep, Anjana S, “Autonomous Car using Raspberry PI and ML”, International Journal of Recent Technology and Engineering (IJRTE), Vol. 9 No. 2, July 2020.

[9] FENICHE, M., & MAZRI, T. (2019). Lane Detection and Tracking for Intelligent Vehicles: A Survey. 2019 International Conference of Computer Science and Renewable Energies (ICCSRE). doi:10.1109/iccsre.2019.8807727.

[10] H Tanveer and A.Sgorbissa, “An Inverse Perspective Mapping Approach usingMonocular Camera of Pepper Humanoid Robot to Determine the Position of Other Moving Robot in Plane” in 15th International Conference on Informatics in Control, Automation and Robotics, IEEE International Conference on, (2018).

[11] J. E. Stellet, M. R. Zofka, J. Schumacher, T. Schamm, F. Niewels, and J. M. Zöllner, “Testing of advanced driver assistance towards automated driving: A survey and taxonomy on existing approaches and open questions,” in Intelligent Transportation Systems (ITSC), 2015 IEEE 18th International Conference on, pp. 1455–1462, IEEE, 2015.

[12] Y.Yenİaydin, K.W.Schmidt, “A lane detection algorithm based on reliable lane markings” in Signal Processing and Communications Applications Conference (SIU), IEEE International Conference on , (2018).

[13] M. Sajjad et al., “An efficient and scalable simulation model for autonomous vehicles with economical hardware,” IEEE Trans. Intell. Transp. Syst., vol. 22, no. 3, pp. 1718–1732, 2021.

[14] K. Gao, B. Wang, L. Xiao, and G. Mei, “Incomplete Road information imputation using parallel interpolation to enhance the safety of autonomous driving,” IEEE Access, vol. 8, pp. 25420–25430, 2020.

[15] M. Sajjad et al., “An efficient and scalable simulation model for autonomous vehicles with economical hardware,” IEEE Trans. Intell. Transp. Syst., vol. 22, no. 3, pp. 1718–1732, 2021.

[16] S. Sujitha, S. Pyari, W. Y. Jhansipriya, Y. Roopeswar Reddy, R. Vinod Kumar, and P. Ravi Nandan, “Artificial Intelligence based Self-Driving Car using Robotic Model,” in 2023 Third International Conference on Artificial Intelligence and Smart Energy (ICAIS), 2023.