International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 13, Issue 4, 2025

Self-Driving Car

Rahmath Unnisa[1], Bhashyakarla Sindoori[2], Velagapudi Yoga Nandini[3], Gundrathi Vaishnavi Goud[4]

[1,2,3,4]B. Tech Students, Department Of Ece, Bhoj Reddy Engineering College For Women, India.

r4262702@gmail.com, sindooribhashyakarla@gmail.com, vyoganandini@gmail.com,

vaishnavigoud844@gmail.com

ABSTRACT

One of the major global challenges today is the huge number of traffic fatalities and the growing issue of traffic congestion. Self-driving cars are an advanced application of artificial intelligence, computer vision, image processing and embedded systems to enable autonomous navigation.

Self-driving cars are a smart combination of computer vision, programming, and robotics. In this project, we built a simple and low-cost self-driving car that can follow road lanes on its own. The system uses a Raspberry Pi and a USB webcam to capture live images of the road. The Raspberry Pi runs a Python program that uses the Open CV library to find lane lines and control the car's movement.

The image from the camera is processed using basic image processing steps. These include converting the image to grayscale, removing noise, detecting edges, and focusing only on the region where lane lines are likely to appear. Once the lane lines are found, the Raspberry Pi sends signals to the motor using GPIO pins to keep the car inside the lane.

This project works only on lane tracking and does not include obstacle detection or traffic signal recognition. It is designed to be simple, affordable. The main goal is to show that even without expensive sensors or hardware a working self-driving system can be built using image processing and basic electronics.

This prototype is a good starting point for learning and future improvements. New features like obstacle detection or GPS navigation can be added later. The project proves that with basic tools and coding, it is possible to create a real-time, lane-following self-driving car.

1. INTRODUCTION

Autonomous vehicle systems have become a major focus of technological innovation, driven by advances in embedded systems, computer vision, and real-time control. Self-driving cars aim to replace or assist human drivers by interpreting sensor and camera data to make navigation decisions. A key challenge in this domain is reliable lane detection and tracking, which ensures the car maintains its position within the driving lane. Lane tracking forms the foundation for higher-level functions such as overtaking, turning, and adaptive cruise control.

The software for this project is developed using a user-friendly programming language that works well with image-based applications. The camera captures images of the road in real time, and the program checks each image to find the road lines. It follows a few basic steps to clean the image and focus only on the important area where the lane lines appear. This helps the system clearly identify the lanes so that the car can stay within them while moving. Based on the position of the detected lanes, the Raspberry Pi calculates the appropriate steering direction. Control signals are then sent through the GPIO pins to a motor driver module, which adjusts the car's motors to steer left, right, or continue straight. The system reacts dynamically to changes in the lane, allowing for smooth movement and lane-following behaviour.

This project demonstrates an effective approach to utilizing image data for autonomous vehicle navigation. By focusing specifically on lane tracking, it provides a clear path for understanding key concepts in computer vision and embedded control systems. The use of cost-effective hardware and open-source software ensures that the system remains accessible for practical development and experimentation. Furthermore, this work establishes a solid foundation for future advancements, including the integration of features such as obstacle detection, traffic sign recognition, and GPS-based navigation, thereby contributing to ongoing progress in the field of autonomous vehicle technologies.

2. LITERATURE SURVEY

The Several recent works have contributed to the advancement of autonomous vehicle technology, particularly in the area of lane detection using low-cost hardware and open-source software. Building on these developments, recent implementations have employed advanced image processing techniques such as edge detection, colour thresholding, and region of-interest extraction to enhance lane visibility under varying environmental conditions. Low-cost platforms like Raspberry Pi combined with OpenCV have enabled real-time video analysis and decision-making for autonomous navigation. These solutions demonstrate practical applicability in constrained environments where affordability and scalability are critical.

[1] *Rahul P. Kharapkar et al. (2020) - IoT based Self Driving Car*
This paper proposes a semi-autonomous car model driven by IoT integration. The system is built using a Raspberry Pi as the central processing unit, with ultrasonic and infrared sensors handling obstacle detection. Motor drivers respond to sensor input to facilitate basic navigation, while IoT capabilities allow for remote monitoring and control. The primary advantage of this approach is its cost-effectiveness and simplicity, making it accessible for prototyping and educational use.
However, the system lacks advanced processing techniques. It does not incorporate machine learning, image processing, or real-time object classification, which limits its functionality to basic movement and static obstacle avoidance. The vehicle is not capable of handling dynamic environments, road sign recognition, or complex decision- making required for real-world autonomous driving. This restricts its potential to indoor or highly controlled outdoor settings.

[2] *D. A. Sutrisno et al. (2020) - Real-time Lane Detection and Motion Planning in Raspberry Pi and Arduino for an autonomous vehicle prototype*
This paper introduces a more refined approach that combines real-time lane detection with motion planning. The system tracks lane boundaries using image processing and estimates the optimal steering path accordingly. This makes it more responsive and accurate in guiding a vehicle through two-lane structured roads, providing a realistic step toward semi-autonomous driving.
Nevertheless, the solution is tailored for environments where road conditions are clearly defined. The absence of dynamic object recognition, such as pedestrians or other vehicles, limits its effectiveness in unstructured or unpredictable scenarios. Without integration with sensor data or AI-based recognition, the system is vulnerable to errors when faced with unexpected road features or obstructions.

[3] *Muhammad Zaim bin Mazlan et al. (2022) - A Lane Detection Using Image Processing Technique for Two-Lane Road*
The study focuses on detecting road lanes in real time using image processing techniques. Implemented with OpenCV, the system uses Canny edge detection to identify sharp changes in intensity and the Hough Transform to trace straight lines, effectively outlining lane boundaries. This approach is particularly suitable for structured and well-marked roads and is a good example of how classical computer vision techniques can be applied to vehicle guidance.
Despite its effectiveness under optimal conditions, the system faces challenges in poor lighting or adverse weather conditions. It also cannot detect faded or inconsistent lane markings, limiting its applicability in real-world driving scenarios. Moreover, the system does not detect dynamic obstacles or traffic signs, reducing its capability to operate independently. While technically sound, it remains a partial solution unless combined with other subsystems like object recognition or sensor fusion.

[4] *Ujwal L, and Dr Nandini K S (2020) - Raspberry Pi and Arduino Based Autonomous Car*
This research outlines a hybrid architecture where Raspberry Pi handles image processing tasks and Arduino controls sensor input and motor movement. This division of labour enables efficient performance, as the load is distributed between two microcontrollers. The system uses ultrasonic sensors and camera data to perform basic navigation and obstacle detection, showcasing sensor fusion in a cost-effective model.
However, synchronizing data between the Raspberry Pi and Arduino can introduce latency, especially when managing real-time image data. The setup may struggle in high-speed scenarios or complex decision-making due to communication bottlenecks. Additionally, it lacks learning-based algorithms, reducing adaptability.

[5] *Husam A Almusawi et al. (2024) - Self-Driving Robotic Car Utilizing Image Processing and Machine Learning*
This study leverages machine learning for lane detection and steering angle prediction. The system trains models to interpret road features from camera input and apply predictive algorithms for vehicle control. The project effectively combines image processing with artificial intelligence, representing a more modern approach to autonomous vehicle design.
Yet, the practical implementation is hindered by the need for large datasets and high computational power. Embedded platforms like Raspberry Pi may not be able to handle the computational demand in real-time. Moreover, training and tuning machine learning models requires time, resources, and expertise, which limits its feasibility in smaller projects or low-power systems.

[6] *Zach Isherwood, and Emanuele Lindo Secco (2022) - A Raspberry Pi Computer Vision System for Self-Driving Cars*
This paper presents a lightweight and power-efficient vision system specifically optimized for Raspberry Pi. Using tools like TensorFlow Lite, it performs basic object detection, image preprocessing, and motion estimation. The system strikes a balance between performance and efficiency, making it well-suited for embedded use cases such as small-scale autonomous robots.
Despite these advantages, the system's capabilities

are limited by hardware constraints. It cannot perform complex navigation or respond to a wide range of dynamic driving situations. While suitable for demonstrating computer vision applications in embedded platforms, it falls short of delivering full autonomous behaviour, especially in unpredictable or real-world traffic conditions.

[7] **P. Golabhavi, and B. Harish (2020) - Self-Driving Car Model using Raspberry Pi** This variant follows a similar low-cost implementation strategy using Raspberry Pi, IR sensors, and a basic camera module. It emphasizes affordability and simplicity in construction, suitable for DIY enthusiasts or academic demonstrations. The car is capable of following a predefined path and can avoid static obstacles based on sensor readings.

However, this approach lacks the sophistication required for real-time autonomous navigation. It cannot manage varying road conditions, traffic interactions, or decision- making based on environmental cues. While functional in lab settings or simple track environments, the model cannot scale to real-world applications. Traffic logic, object detection, and environmental awareness are entirely absent.

3.HARDWARE AND SOFTWARE REQUIREMENTS

This chapter details the essential hardware and software components used in the Self-Driving Car project. In this work, we have successfully developed a low-cost, vision-based autonomous car prototype using Raspberry Pi 3 B+, a webcam, and fundamental electronic components. The system is designed to follow a predefined path by capturing real-time road visuals, detecting lane markings using OpenCV, and making steering decisions such as turning left, right, or moving straight based on the lane position.

Unlike high-end autonomous systems that rely on AI, GPS, LiDAR, or machine learning models, this project emphasizes simplicity, affordability, and accessibility, leveraging basic computer vision techniques and Python-based decision logic. This makes the project highly suitable for educational environments, early-stage research, and low-budget experimentation.

The integration of Raspberry Pi for image processing and control, along with a motor driver, DC motors, and a stepper motor for movement and steering, demonstrates a practical and hands-on approach to implementing core concepts of autonomous vehicle navigation. All decisions are processed locally by the Raspberry Pi without the need for external servers or complex infrastructure.

This project not only helps in understanding the basic mechanisms behind self-driving technologies but also serves as a flexible and scalable platform for future enhancements, such as adding traffic sign recognition, obstacle detection, or wireless communication feature- building a strong foundation in the fields of embedded systems, robotics, and computer vision.

Hardware Requirements

1. **Raspberry Pi 3 B+:**

The Raspberry Pi 3 B+ is a powerful single-board computer used as the main processing unit in this project. It runs the OpenCV library and Python scripts to process the video feed from the webcam. In this self-driving car, the Raspberry Pi captures and analyzes road images in real time. It detects lane markings and then makes directional decisions like turning left, right, or moving straight. These decisions are then sent to the motor driver through GPIO pins for motor control, making the Raspberry Pi the central "brain" of the vehicle.

2.Motor Driver:

The motor driver (typically an L298N module) acts as an interface between the Raspberry Pi and the motors. It receives low-power control signals from the Raspberry Pi's GPIO pins and uses them to drive the high-power DC motors and stepper motor. The motor driver controls both the direction and speed of the motors based on the output of the lane detection algorithm. It ensures that the car moves and steers smoothly according to the Raspberry Pi’s real-time decisions, enabling accurate lane following and autonomous navigation.

Software Requirements

The software aspect of the self-driving car system plays a critical role in capturing and processing live video data, detecting lane lines, and issuing navigation commands. All development and execution of the system’s logic are performed directly on the Raspberry Pi 3 B+ using Python and the OpenCV library. The Python code is responsible for processing frames from the webcam in real time, applying computer vision techniques to detect lanes, and making steering decisions. These decisions are then used to control the motor driver through the Raspberry Pi’s GPIO pins. This section highlights the software tools, libraries, and programming environment used to implement the car’s autonomous navigation effectively and efficiently and also about the programming languages we use.

1.IDLE:

The Integrated Development and Learning Environment (IDLE) is used to write, test, and execute Python code on the Raspberry Pi 3 B+ in this project. It offers a user- friendly interface to manage real-time image processing tasks using the OpenCV library. Through IDLE, the Python scripts are developed to capture video frames from the webcam, detect lane markings, and make directional decisions such as turning left or right. The platform supports necessary libraries for computer vision and serial communication, enabling smooth interaction

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 13, Issue 4, 2025

between the Raspberry Pi and motor control systems. Additionally, IDLE's interactive shell and built-in debugger facilitate efficient code testing and troubleshooting during development, ensuring accurate camera input handling and responsive navigation logic.

2.Programming Languages:

The self-driving car project is programmed primarily using the Python programming language, implemented directly on the Raspberry Pi. Python was used to write and execute the complete control logic, including real-time image processing, lane detection, and motor control. The code captures frames from the webcam, analyses them using the OpenCV library, and makes navigation decisions such as turning left, right, or moving forward based on the detected lane position.

4.SELF-DRIVING CAR

Self-driving cars, also known as autonomous vehicles, are transforming the future of transportation by using computer vision, embedded systems, and real-time decision-making to navigate without human control. This project aims to develop a basic prototype of a vision- based self-driving car that can detect lane markings and make steering decisions accordingly. The system is built using a Raspberry Pi 3 B+ as the central controller, along with a webcam for visual input, a motor driver module, and DC and servo motors for movement and directional control. This project offers a hands-on experience with embedded vision systems, GPIO control, and autonomous navigation using only affordable hardware and open-source platforms.

With the rise in demand for intelligent transportation solutions, self-driving technology is becoming increasingly relevant in sectors like smart mobility, logistics, and traffic management. Autonomous vehicles promise to improve road safety, reduce human error, and enhance transport efficiency. This mini project simulates a lane-following self-driving system, utilizing low-cost components and a modular design approach. It focuses on essential functions such as lane detection, real-time image processing, and directional movement, all executed through Python scripts running on the Raspberry Pi. The goal is to demonstrate the core functionality of autonomous navigation using computer vision, without relying on complex or expensive hardware.

At the hardware level, the webcam captures continuous video feed of the road, which is processed by the Raspberry Pi using OpenCV to identify lane markings. Based on the position of these lanes, the Raspberry Pi makes directional decisions and directly controls the motor driver via its GPIO pins. This single-board system architecture ensures compact design, efficient processing, and real-time responsiveness. The use of Python, OpenCV, and GPIO programming provides an excellent learning platform for exploring the fundamentals of embedded vision, automation, and robotics.

Block Diagram & its Explanation

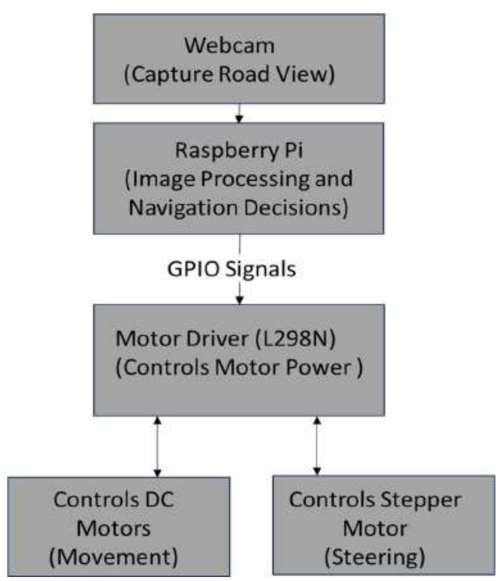

Figure 1 Block Diagram of Self-Driving Car

5.RESULTS

The designed and implemented self-driving car

International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 13, Issue 4, 2025

system was thoroughly tested to validate its functionality, lane detection, and obstacle avoidance capabilities. The system successfully integrates key hardware components—such as a camera module, ultrasonic sensors, an L298N motor driver, DC motors, a servo motor, and a Raspberry Pi 3 B+ with control logic written entirely in Python.

Upon powering up, the Raspberry Pi acts as the central controller, initiating the system and managing all peripherals directly through its GPIO pins. The camera module continuously captures real-time video of the road, which is processed by the Pi using OpenCV for lane detection. Simultaneously, the ultrasonic sensor, connected to the Pi, monitors for obstacles ahead. The sensor data is processed in real-time to ensure collision avoidance.

Based on the detected lanes and obstacle distance, the Raspberry Pi makes dynamic navigation decisions—such as whether to move forward, stop, or adjust steering direction using the servo motor. It then generates appropriate GPIO signals to the L298N motor driver, which drives the DC motors for movement and controls the servo for steering.

Figure 2 Car Navigating a Curved Black Track

This integrated setup—without the need for a master-slave architecture—ensures a compact, efficient, and responsive design where the Raspberry Pi alone performs image processing, sensor data handling, and motor control in real-time.

Key Results:

- Successful Integration of all hardware components (Raspberry Pi, camera, ultrasonic sensor, L298N motor driver, DC motors, and servo motor) into a functional self-driving prototype.
- Accurate Lane Detection using image processing in real-time with Python and OpenCV. And car was able to move forward, turn left/right, and stop based on the detected lanes and obstacle proximity.
- All tasks including image processing, lane detection, decision-making, and motor control were handled efficiently by the Raspberry Pi alone, eliminating the need for master-slave architecture.
- Real-Time GPIO Communication between Raspberry Pi and L298N motor driver ensured smooth and responsive motor and servo control.
- The system responded correctly to real-time environmental changes with logged outputs

International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 13, Issue 4, 2025

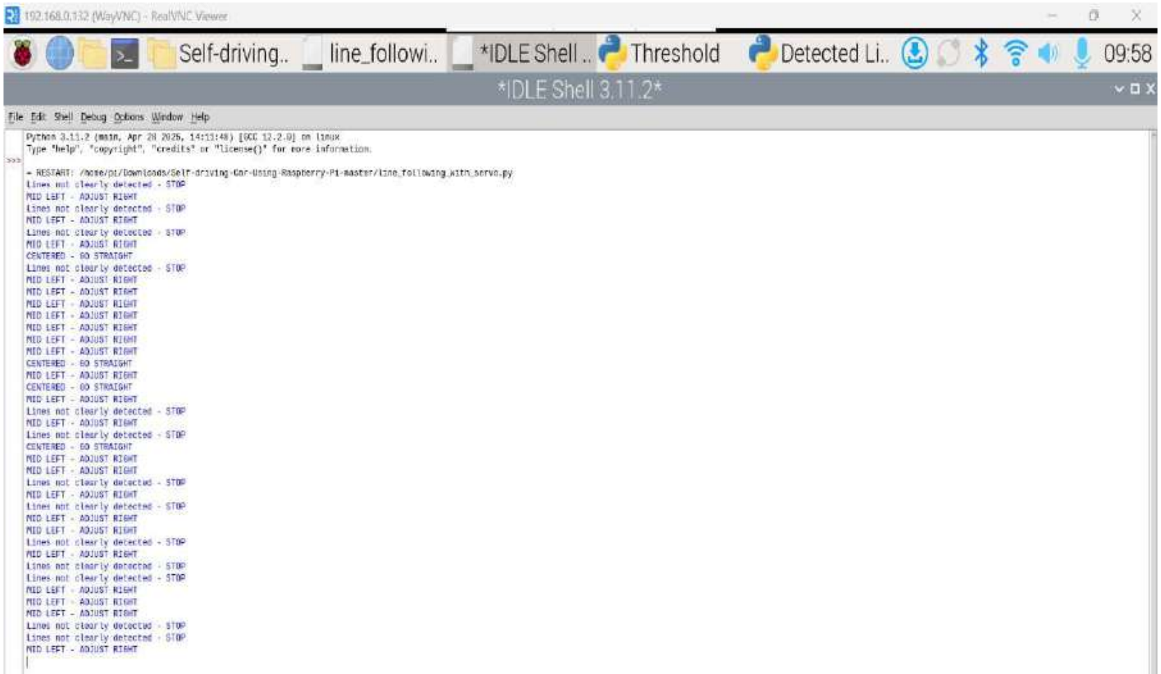

Figure 3 Console Output showing Directional Commands based on Detected Lanes

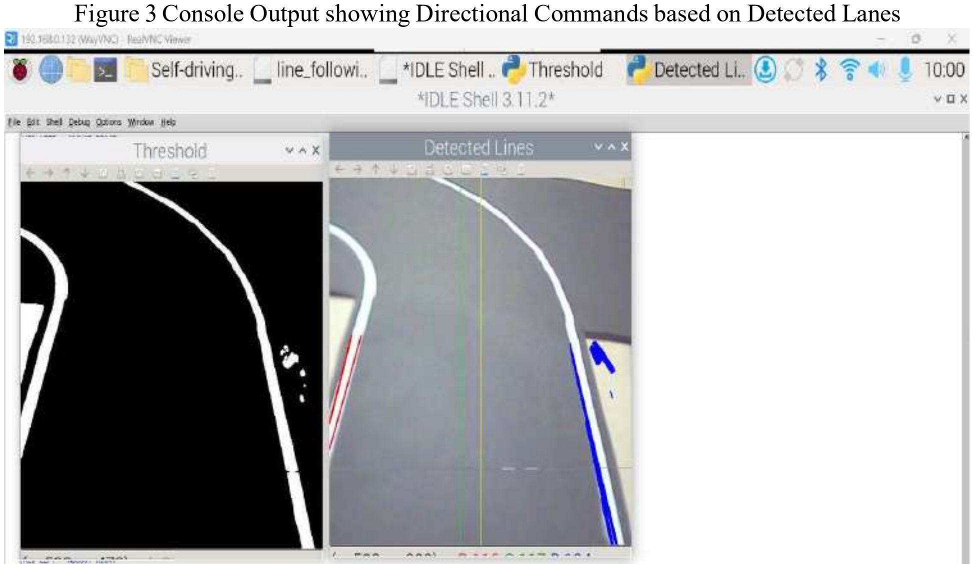

Figure 4 Visual Feedback of Lane Detection

6.CONCLUSION

In this project, we successfully designed and implemented a low-cost, vision-based self-driving car prototype using Raspberry Pi 3 B+, a mounted webcam, and the OpenCV library. The system is centered around essential aspects of autonomous navigation-particularly real-time lane detection and directional decision-making achieved through fundamental computer vision techniques. Instead of relying on high-end tools like GPS, AI models, or sensor arrays, this project focuses on simplicity, affordability, and core functionality.

The webcam captures live road visuals, which are processed by Python scripts running on the Raspberry Pi. Techniques such as Canny Edge Detection and Hough Line Transform allow the system to identify lane markings and determine the appropriate movement direction left, right, or straight. These decisions are translated into motor control signals that drive the DC motors and stepper motor through a motor driver module, enabling responsive and autonomous steering within a controlled track environment.

The use of modular, budget-friendly components ensures that the system is both accessible and easy to replicate, making it an ideal learning tool for students, hobbyists, and developers exploring embedded systems and robotics. Its straightforward design encourages a deeper understanding of perception and control—the foundational pillars of autonomous vehicles.

Thus, this project lays a strong technical foundation for future advancements in self-driving technology. With future upgrades such as adaptive speed control, traffic sign recognition, and enhanced lane tracking algorithms, this prototype can evolve into a more advanced autonomous platform, opening pathways for research, experimentation, and real-world applications in intelligent transportation systems.

REFERENCE

[1] Rahul P. Kharapkar et al., "IoT based Self Driving Car", *IRJET J.*, vol. 7, no. 3, pp. 5177–5179, Mar. 2020.

[2] D. A. Sutrisno et al., "Real-time Lane Detection and Motion Planning in Raspberry Pi and Arduino for an autonomous vehicle prototype", *IUT C.*, Sep. 2020.

[3] Muhammad Zaim bin Mazlan et al., "A Lane Detection Using Image Processing Technique for Two-Lane Road", *IEEE Systems, Process & Control (ICSPC) C.*, pp. 214–215, Dec. 2022.

[4] Ujwal L, and Dr Nandini K S, "Raspberry Pi and Arduino Based Autonomous Car", *iJRDO J.*, vol. 06, no. 4, ISSN: 2456-6055, pp. 1–6, Apr. 2020.

[5] Husam A Almusawi et al., "Self-Driving robotic car utilizing image processing and machine learning", *IOP C.*, vol. 1256, pp. 012024, Jan. 2024.

[6] Zach Isherwood, and Emanuele Lindo Secco, "A Raspberry Pi Computer Vision System for Self-Driving Cars", vol. 13313, pp. 834–846, Oct. 2022.

[7] P. I. Golabhavi, and B. P. Harish, "Self-Driving Car Model using Raspberry Pi", *IJERT J.*, vol. 9, no. 2, pp. 176–178, Feb. 2020.

[8] P. Lohakare, M. Kumar and G. B. Sambhaji, "Self-Driving Car Using Raspberry Pi", *IRJMETS J.*, vol. 6, no. 1, pp. 3183–3184, Jan. 2024.

[9] V. Gaikwad et al., "Smart Car Using IoT and Image Processing", *ICCUBEA J.*, pp. 1–4, Aug. 2023.