Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 06, ACSCON - 2026

ML based Self Driving Car

Prathush H.
Dept. of ECE,
CMR Institute of Technology
Bengaluru, India
prah22ece@cmrit.ac.in

Likith S.
Dept. of ECE,
CMR Institute of Technology
Bengaluru, India
liks22ece@cmrit.ac.in

Manoj S.
Dept. of ECE,
CMR Institute of Technology
Bengaluru, India
mans22ece@cmrit.ac.in

Rakesh K S
Dept. of ECE .,
CMR Institute of Technology
Bengaluru, India
raks22ece@cmrit.ac.in

P. Susheelkumar Sreedharan
Professor, Dept. of ECE,
CMR Institute of Technology
Bengaluru, India
susheelkumar.s@cmrit.ac.in

Abstract*— Here in this research project, we have embarked on a journey towards implementing a low-cost intelligent self-driving vehicle prototype using a Raspberry Pi 3 B+ aided with real-time computer vision and sensor-based control techniques. The primary objective of the work is to demonstrate how embedded systems and image processing can be effectively integrated to achieve basic autonomous navigation and traffic awareness in a cost-efficient and scalable manner. The proposed system is aimed at researching and developing a prototype for academic and educational applications. Hence our main focus is on affordability and simplicity. The self-driving vehicle is equipped with a camera module mounted on the front of the car to capture continuous real-time video of the surrounding environment. The captured video stream is transmitted to a processing server, where OpenCV-based image processing algorithms are employed to assess and analyze the traffic conditions. The vision module is responsible for detecting traffic signal colors—red, yellow, and green—by extracting color features and applying thresholding techniques. In addition to traffic light detection, the system identifies and estimates the number of nearby vehicles present in the field of view, enabling the vehicle to make informed decisions regarding speed and movement. Based on the outcomes of the computer vision analysis, control commands are generated and sent to the Raspberry Pi, which serves as the central processing and control unit of the vehicle. When a green traffic signal is detected, the vehicle is allowed to move at normal speed. Detection of a yellow signal results in a controlled reduction of speed to ensure cautious movement, while a red signal triggers an automatic halt of the vehicle, ensuring compliance with traffic rules. The adaptive speed control mechanism enhances safety and smooth navigation by dynamically responding to changing traffic conditions. To complement the vision-based perception system, an ultrasonic sensor is integrated for real-time obstacle detection.

Keywords—Raspberry Pi, vision, OpenCV, vehicle, obstacle, object detection

I. INTRODUCTION

Autonomous driving systems have gained significant research attention due to their potential to improve road safety, reduce traffic congestion, and minimize human error. Modern self-driving vehicles are designed to perceive their surroundings, analyze traffic conditions, and make real-time control decisions using a combination of embedded computing, computer vision, and sensor-based technologies. While commercial autonomous platforms achieve high accuracy using advanced sensors and high-performance hardware, their high cost and system complexity limit their suitability for academic and educational environments.

To address this limitation, low-cost autonomous vehicle prototypes have emerged as effective tools for research and learning. Embedded platforms such as the Raspberry Pi provide a flexible and affordable solution for implementing intelligent vehicular systems. The Raspberry Pi 3 B+ offers sufficient processing capability, networking support, and peripheral interfaces to connect cameras, sensors, and motor drivers, making it suitable for small-scale autonomous driving applications.[1]

Vision-based perception plays a critical role in autonomous navigation by enabling the detection of traffic signals and surrounding vehicles. Classical traffic signal recognition techniques commonly employ color segmentation in the HSV color space combined with morphological operations, offering computational simplicity suitable for embedded systems [1]. Similarly, vehicle detection and counting techniques range from contour-based approaches to more advanced feature-based and deep learning methods, which are often executed on remote servers to reduce onboard processing load [2]. Offloading computationally intensive image processing tasks to a server enables real-time performance while maintaining a lightweight vehicle-side architecture [2].

However, vision-based systems alone may suffer from reduced reliability under poor lighting conditions or visual occlusions. To enhance system safety, ultrasonic sensors are widely used for short-range obstacle detection due to their simplicity, low cost, and fast response time. These sensors provide reliable distance measurements and enable immediate emergency stopping when obstacles are detected within a predefined threshold. The integration of vision-

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 06, ACSCON - 2026

based perception with ultrasonic sensing improves overall system robustness through sensor fusion.

II. RELATED WORKS

Existing research demonstrates the extensive use of IoT, embedded systems, and wireless communication technologies for intelligent monitoring, security, and autonomous applications. Several studies focus on sensor-based detection and real-time alerting systems using low-cost microcontrollers and wireless networks to improve situational awareness and response time. Embedded platforms such as Arduino and Raspberry Pi are widely adopted due to their flexibility and affordability, enabling efficient sensor interfacing, data processing, and control operations.

Vision-based and machine-learning-assisted approaches further enhance system intelligence by enabling object detection, traffic analysis, and decision-making, often through server-side processing to overcome embedded hardware limitations . Additionally, GPS, flame, tilt, and ultrasonic sensors are commonly integrated to improve location tracking, safety, and environmental awareness . Recent studies emphasize sensor fusion, energy efficiency, and scalable network architectures to improve reliability and adaptability of intelligent systems. [1] – [23].

Table 1: Comparative analysis of Literature Survey

Ref.	Main Focus	Key Features	Components Used
[1] – [5]	IoT-based monitoring & security	Real-time alerts, remote sensing	Sensors, IoT network
[6], [13]	Embedded platforms	Low-cost control, interfacing	Arduino / Raspberry Pi
[8], [14]	ML & vision systems	Object detection, analysis	Camera, ML models
[7], [20]	GPS tracking	Location monitoring	GPS module
[15], [16], [17]	Safety sensing	Fire & tilt detection	Flame, tilt sensors
[11], [12]	Scalable communication	Long-range & aerial monitoring	LoRa, drones
[22], [23]	Future intelligent systems	Sensor fusion, scalability	IoT + ML

III. METHODOLOGY

The proposed ML-based self-driving car system follows a modular and layered methodology, as illustrated in the system block diagram. The overall operation is divided into sensing, perception, decision-making, and actuation stages to ensure reliable and real-time autonomous navigation.

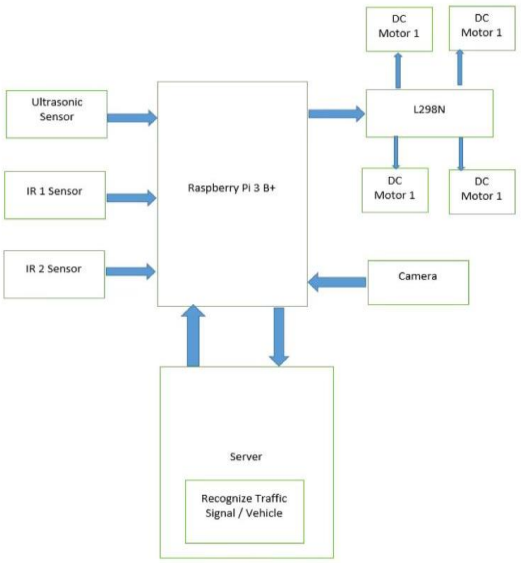

Fig 1: Block diagram of the project

- **System Initialization and Embedded Control Unit**

The Raspberry Pi 3 B+ acts as the central embedded controller of the system. Upon power-up, the Raspberry Pi initializes all connected peripherals, including the camera module, ultrasonic sensor, IR sensors, motor driver, and network interface. It establishes a wireless connection with the remote server to enable continuous video streaming and reception of control commands. The Raspberry Pi also continuously monitors sensor inputs to ensure safe system operation.

- **Camera-Based Data Acquisition and Video Streaming**

A camera module mounted on the front of the vehicle captures real-time video of the surrounding environment. The captured frames are streamed from the Raspberry Pi to a remote processing server using a Wi-Fi network. This streaming-based approach reduces the computational burden on the embedded platform while ensuring uninterrupted visual data transmission for real-time analysis.

- **Server-Side Image Processing and Traffic Analysis**

The remote server receives the live video stream and performs image processing using OpenCV-based algorithms. Traffic signal detection is carried out using color-based segmentation techniques to identify red, yellow, and green

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 06, ACSCON - 2026

signals. In parallel, vehicle detection and counting algorithms are applied to estimate traffic density within the field of view. The processed results are converted into high-level control decisions such as stop, slow down, or move forward.

- **Traffic Signal Interpretation and Speed Decision Logic**

Based on the detected traffic signal state, predefined rules are applied to determine vehicle behavior:

- Red signal: Vehicle is commanded to stop
- Yellow signal: Vehicle speed is reduced
- Green signal: Vehicle moves at normal speed

These commands are transmitted from the server back to the Raspberry Pi for execution.

- **Ultrasonic-Based Obstacle Detection**

An ultrasonic sensor mounted at the front of the vehicle continuously measures the distance to obstacles. The Raspberry Pi processes the echo time to calculate object distance. If an obstacle is detected within a predefined safety threshold, the system immediately triggers an emergency stop. This safety mechanism has the highest priority and overrides all server-generated commands.

- **IR Sensor-Based Proximity and Edge Detection**

IR sensors integrated into the system provide short-range detection capabilities. These sensors assist in detecting edges, close obstacles, or line-based navigation scenarios, particularly in indoor or controlled environments. The sensor inputs supplement ultrasonic and vision data to improve navigation reliability.

- **Decision-Making and Sensor Fusion**

The Raspberry Pi performs sensor fusion by combining inputs from:

- Server-side vision processing
- Ultrasonic distance measurements
- IR sensor feedback

A rule-based control algorithm prioritizes safety-related inputs and determines the final motion command. This approach ensures stable operation even under communication delays or vision failures.

IV. Implementation Detail

Hardware Components

- **Raspberry Pi 3 B+:**

The Raspberry Pi 3 B+ serves as the main embedded controller of the system. It interfaces with sensors, manages communication with the remote server, executes control logic, and sends actuation commands to the motor driver for vehicle movement.

- **Camera Module**

A camera module is mounted on the vehicle to capture real-time video of the surrounding environment. The video stream is transmitted to a remote server, where image processing and traffic analysis are performed.

- **Ultrasonic Sensor (HC-SR04)**

The ultrasonic sensor is used for real-time obstacle detection by measuring the distance between the vehicle and nearby objects. It enables emergency braking by triggering an immediate stop when an obstacle is detected within a predefined safety range.

- **IR Sensors**

IR sensors provide short-range proximity detection and assist in identifying nearby obstacles or edges. These sensors enhance navigation reliability, particularly in indoor or controlled environments.

- **L298N Motor Driver**

The L298N motor driver acts as an interface between the Raspberry Pi and the DC motors. It amplifies low-power control signals and enables bidirectional control of motor speed and direction.

- **DC Motors**

DC motors are responsible for the physical movement of the vehicle. Their speed and direction are controlled through the motor driver based on commands generated by the control algorithm.

- **Chassis**

The chassis provides mechanical support and structural stability to the vehicle. It ensures proper alignment and mounting of all electronic and mechanical components.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 06, ACSCON - 2026

- **Power Supply/Battery**

A rechargeable battery is used to supply power to the Raspberry Pi, sensors, and motors. Voltage regulation ensures stable and uninterrupted system operation.

Software Components

- **Raspberry Pi OS**

Raspberry Pi OS is used as the operating system to manage hardware resources, device drivers, and network connectivity. It provides a stable environment for executing control and vision-processing programs.

- **Python Programming Language**

Python is used for implementing control logic, sensor data processing, and communication between the Raspberry Pi and the server. Its simplicity and extensive library support make it suitable for rapid development.

- **OpenCV Library**

OpenCV is employed for image processing tasks such as traffic signal detection and vehicle analysis. It enables efficient real-time processing of video frames using computer vision algorithms.

- **Machine Learning Techniques**

Machine learning models are utilized to enhance object detection and classification accuracy. These techniques assist in interpreting visual data and generating intelligent control decisions.

- **Wi-Fi Communication**

Wi-Fi enables wireless transmission of live video streams and control commands between the Raspberry Pi and the server. This communication framework supports real-time system operation.

- **Server-Side Processing**

Computationally intensive image processing and analysis are performed on a remote server. This approach reduces the processing load on the Raspberry Pi while maintaining real-time performance.

V. Results and Discussion

- **System Functionality Validation**

The complete system was successfully implemented and tested to validate the integration of all hardware and software modules. The Raspberry Pi 3 B+ effectively controlled the camera, ultrasonic sensor, IR sensors, and motor driver, while maintaining stable communication with the remote server. All system components operated reliably throughout the testing phase.

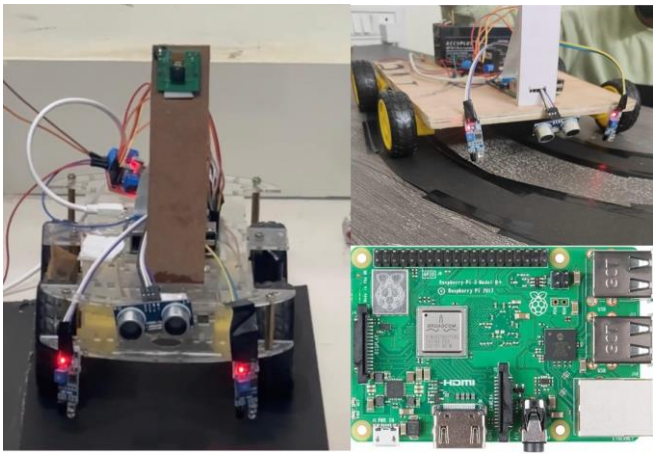

Fig 2: Project Implementation

- **Traffic Signal Detection Performance**

The vision-based traffic signal detection module accurately identified red, yellow, and green signals from real-time video streams. The detected signal states were correctly interpreted, and corresponding control commands were generated. The vehicle consistently stopped at red signals, reduced speed at yellow signals, and moved forward at green signals, demonstrating effective perception and response.

- **Vehicle Detection and Traffic Analysis**

The vehicle detection module successfully identified surrounding vehicles within the camera's field of view. Traffic density information was used to influence speed control decisions, enabling adaptive vehicle movement. Server-side processing ensured smooth and continuous analysis without overloading the embedded system.

- **Obstacle Detection and Collision Avoidance**

The ultrasonic sensor provided accurate distance measurements and reliably detected obstacles within the predefined safety range. Upon obstacle detection, the system immediately initiated an emergency stop, overriding all other control commands. This ensured effective collision avoidance and enhanced overall system safety.

- **IR Sensor-Based Proximity Support**

IR sensors contributed to short-range proximity detection and improved navigation reliability, particularly in confined or indoor environments. The integration of IR sensing complemented ultrasonic and vision-based inputs, resulting in more stable system behavior.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 06, ACSCON - 2026

- **Real-Time Performance and System Response**

Offloading computationally intensive image processing tasks to a remote server significantly reduced the processing load on the Raspberry Pi. The system achieved real-time responsiveness with minimal latency in video transmission, decision-making, and motor actuation.

VI. Conclusion

The results confirm that the proposed ML-based self-driving car system effectively integrates computer vision, sensor fusion, and embedded control. The system demonstrates reliable autonomous navigation using low-cost hardware and open-source software, making it suitable for educational and research-oriented applications.

References

[1] R. K. Verma and S. Kumar, "IoT-based anti-poaching system for wildlife protection," *Journal of Wildlife Conservation*, vol. 8, no. 4, pp. 213–221, 2019.
[2] A. N. Williams, "IoT-based security systems: A case study on anti-poaching," International Journal of IoT, vol. 5, pp. 98–104, 2020.
[3] H. Z. Lee, "A comprehensive approach to anti-poaching systems using IoT," IEEE Transactions on Sensor Networks, vol. 21, pp. 657–663, 2022.
[4] A. K. Singh and S. Pandey, "Wireless communication systems for remote monitoring," Journal of Communication Engineering, vol. 15, no. 1, pp. 122–130, 2020.
[5] P. J. Brown, "Arduino Uno: A low-cost solution for embedded systems," Electronics for Beginners, vol. 12, pp. 45–48, 2018.
[6] J. P. Martinez and R. D. Rodriguez, "The role of GPS technology in environmental monitoring," Journal of Geospatial Information, vol. 14, no. 3, pp. 99–107, 2021.
[7] K. S. Patil, "Integrating machine learning with IoT for wildlife protection," Journal of Artificial Intelligence, vol. 9, pp. 80–88, 2022.
[8] T. H. Wang and L. R. Yang, "Applications of IoT in conservation and anti-poaching," Nature and Technology, vol. 13, pp. 50–56, 2020.
[9] M. Sharma and V. Mehra, "GSM-based monitoring systems in remote areas," International Journal of Communication Systems, vol. 11, pp. 65–72, 2019.
[10] S. R. Gupta, "LoRaWAN in IoT applications: Enhancing long-range communication," Journal of Wireless Networks, vol. 16, pp. 88–94, 2021.
[11] R. L. Singh, "Use of drones for wildlife surveillance," Journal of Remote Sensing, vol. 19, pp. 120–127, 2020.
[12] B. D. Taylor and A. M. Patel, "Designing cost-effective anti-poaching systems using IoT," International Journal of Embedded Systems, vol. 10, pp. 34–41, 2019.
[13] P. K. Yadav, "Flame detection sensors: Use in fire safety and anti-poaching," Journal of Sensor Technology, vol. 12, pp. 90–96, 2021.
[14] J. A. Singh, "Tilt sensors for wildlife monitoring," International Journal of Embedded Applications, vol. 8, pp. 55–61, 2018.
[15] L. M. Roberts, "The impact of IoT on conservation and poaching prevention," Journal of Environmental Monitoring, vol. 17, pp. 44–51, 2020.
[16] H. S. Kumar and A. R. Patel, "Energy-efficient IoT devices for environmental monitoring," Journal of Power Systems, vol. 14, pp. 121–130, 2019.
[17] P. R. Sharma, "Security solutions using IoT in remote areas," Journal of Cyber Security, vol. 10, pp. 99–105, 2020.
[18] M. A. Davis, "Bluetooth technology in IoT-based anti-poaching systems," International Journal of Communication Networks, vol. 18, pp. 10–17, 2021.
[19] V. C. Johnson, "Wireless sensor networks for poaching detection," Journal of Environmental Protection, vol. 14, no. 2, pp. 85–92, 2020.
[20] M. V. Gupta, "Real-time wildlife tracking and monitoring using GPS," Journal of Environmental Systems, vol. 13, pp. 50–55, 2022.
[21] R. M. Patel, "Bluetooth communication in embedded systems," International Journal of Embedded Computing, vol. 9, pp. 23–30, 2018.
[22] N. A. Jones, "The future of IoT in anti-poaching systems," Journal of Technology in Conservation, vol. 20, pp. 75–82, 2022.
[23] L. F. Choi, "Advances in IoT-based wildlife protection," Environmental Technology Journal, vol. 15, pp. 122–128, 2021.