Nanotechnology Perceptions
ISSN 1660-6795
www.nano-ntp.com

Design And Development Of A Mobile-Powered Smart Car With Object Recognition

Nikhil Sharma[1], Ritu Raj Raghuvanshi[2], Sanjay Jathar[3], Manish Thakur[4], Rudra Dixit[5], Shreyash Tiwari[6], Mohit Sahu[7]

Department of Mechanical Engineering, Medi-Caps University, Indore-453331, India.

The objective of this research work involves the creation of a smart robotic car that operates using a mobile phone and can recognition objects in real time. The primary aim is to develop an affordable and efficient system by using a smartphone as the processing unit. The car is assembled with components like a microcontroller (ESP32-CAM), motor driver(L298N), and battery (lithium ion 12v), and it communicates with the phone wirelessly through Wi-Fi (2.4). The mobile device uses its camera to capture surroundings and processes the visuals models through OpenCV. Based on what it detects such as obstacles or people the car takes decisions like stopping or changing direction to avoid impact. A dedicated mobile application provides both control and live feedback. The setup is designed to be low-cost, easy to use, and efficient system that is adaptable for future features like voice input or GPS navigation. This project highlights the practical use of mobile-based AI in building responsive, intelligent robotic systems.

Keywords: Smart Car, Object Recognition, Mobile Powered, Real Time Detection

1. Introduction

With the rapid progress of technology, automation and intelligent transportation have become key focus areas. Among them, autonomous vehicles cars that can drive and make decisions without human control are gaining a lot of attention. These vehicles rely on sensors, cameras, and artificial intelligence to understand their surroundings and respond safely. However, building such systems is usually expensive and complex. This project presents a cost-effective solution by developing a smart car that uses a regular smartphone for processing and can detect objects in real-time.

In comparison, traditional or manually controlled vehicles do not have any built-in intelligence. They fully depend on the driver for every action, including navigation, obstacle avoidance, and safety decisions. If the driver is distracted or slow to react, accidents may happen. These vehicles lack the ability to sense their environment. On the other hand, the smart car in this project uses a smartphone's camera and AI models to recognize objects ahead, make decisions, and move accordingly, reducing human input and increasing automation. The smart car system is built using basic electronic parts like a microcontroller, motor driver, DC motors, and a battery. The smartphone is placed on the car and connected wirelessly to the controller through Wi-Fi. A mobile application is used for real-time video display and movement control. The camera feed is processed using tools like OpenCV allowing the car to identify , objects, or obstacles and act based on those result such as stopping or turning. This project proves that mobile devices can serve as powerful processors for robotics. It removes the need for expensive computing platforms and provides an affordable learning model for students and beginners. It also creates opportunities for future enhancements like GPS navigation, voice commands, or cloud-based data handling. To sum up, non-autonomous vehicles rely entirely on human control and offer no intelligent decision-making ability. The smart car proposed here introduces object recognition and automated responses, making it more advanced and safer in comparison. This low-cost design bridges the gap between traditional vehicles and intelligent systems, offering a hands-on way to learn about AI, robotics, and smart automation**.**

2.Literature Review

The integration of advanced object detection algorithms into autonomous systems has gained significant momentum in recent years, driven by the need for accurate, real-time environmental perception in mobile robots, autonomous vehicles, and embedded systems. This review consolidates recent developments from multiple studies, emphasizing novel detection strategies, optimization techniques, and hardware constraints.

Achirei et al. (2023) [1] explored a Model-Predictive Control (MPC) framework for omnidirectional mobile robots, coupling it with CNN-based object detection. The proposed system enables navigation and path planning within dynamic logistics environments by leveraging real-time object recognition to inform control decisions. This work stands out by integrating vision-based detection with control algorithms, showcasing practical utility in industrial settings. Kannamma et al. (2024) [2] presented a hybrid model using Dragonfly-Based Optimization (DBO) to fine-tune features in the Refined Feature Region Classification (RFRC) method. This technique significantly enhances object detection for autonomous vehicles, especially in cluttered and dynamic scenes.

Several studies have focused on refining autonomous vehicle detection using deep learning techniques. Alam et al. (2023) [3] introduced a Faster R-CNN-based method for robust vehicle classification. Their algorithm emphasizes resilience to occlusion and environmental noise, facilitating improved recognition in complex road scenarios. Shin et al. (2022) [4] introduced a self-supervised 3D object detection pipeline, "Sample, Crop, Track," using LiDAR for urban driving. Their work leverages unsupervised learning to improve data efficiency, emphasizing the growing role of 3D perception in urban navigation.

Zhao et al. (2023) [5] proposed an enhanced Faster R-CNN model that incorporates multi-scale feature fusion and shape prior information. This fusion allows for more precise detection in densely populated scenes, addressing limitations in traditional CNN frameworks. With increasing demands for low-power, real-time inference, several studies have explored Tiny ML solutions.

Moosmann et al. (2023) [6] developed Tinyissimo YOLO, a quantized, low-memory object detection model tailored for microcontroller units (MCUs). This approach balances performance and resource constraints, making it ideal for edge deployments in IoT and robotics. Xu et al. (2023) [7] tackled onboard dynamic object detection and tracking using RGB-D cameras. Their approach enhances robot navigation through effective depth sensing and object motion prediction, promoting real-time interaction with dynamic environments. Bechtel et al. (2022) [8] implemented Deep Picar Micro, a Tiny ML framework designed for autonomous Cyber-Physical Systems. The study highlights trade-offs between model size, inference speed, and accuracy, providing a pathway for deploying deep learning on constrained hardware. Boyle et al. (2024) [9] presented DSORT-MCU, aimed at real-time detection of small objects on MCUs. The framework optimizes the SORT algorithm for resource efficiency, improving detection capabilities in embedded systems with strict latency requirements.

Shin et al. (2022) [10] contributed an environment-adaptive object detection framework, enhancing robot adaptability across variable terrains and lighting conditions. The emphasis was on context-awareness to boost detection accuracy, offering an adaptable solution for mobile robots operating in heterogeneous environments. Li & Wang (2023) [11] proposed an improved Faster R-CNN model optimized for detecting vehicles in remote sensing imagery. Their model introduces multiscale feature fusion and holography-based augmentation, which helps counter the scale variation and orientation diversity of vehicles in aerial images. The study significantly enhances detection accuracy compared to traditional Faster R-CNN setups, making it highly suitable for overhead surveillance and urban planning applications where vehicles may appear in diverse perspectives.

Ren & Li (2023) [12] developed an adaptive model predictive control (MPC) system tailored for omnidirectional mobile robots. This approach features friction compensation and incremental input constraints, improving tracking precision in complex dynamic environments. While not directly a vision-based detection method, this work complements object detection frameworks by providing a more stable and responsive navigation system, which is crucial for integrating perception and control in mobile robotics.

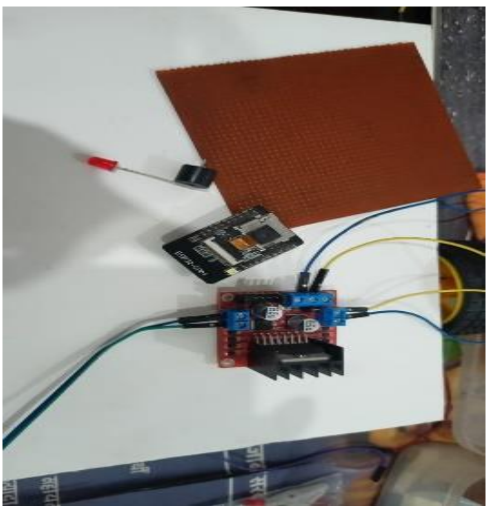
Zhang & Liu (2024) [13] presented a detailed review of deep learning approaches to vehicle detection and classification. The article summarizes the evolution of methods including YOLO, SSD, R-CNN variants, and emerging techniques involving temporal models (e.g., RNNs). It critically analyzes each method’s performance across different tasks such as autonomous driving, traffic analysis, and aerial detection, identifying trade-offs between accuracy, latency, and model complexity. This review contextualizes current research and outlines key challenges such as occlusion, domain adaptation, and dataset bias.

Wang & Zhao (2023) [14] introduced a multi-feature fusion network optimized for infrared object detection in unmanned aerial vehicle (UAV) systems. Their lightweight architecture caters to low-power, real-time inference demands while maintaining detection accuracy. The system fuses texture, shape, and thermal data, enabling enhanced object recognition in lowvisibility conditions. This work is particularly relevant for surveillance and rescue missions using drones where infrared sensors are often deployed.

Shin et al. (2022) [15] (also included in your original set as [4]) proposed "Sample, Crop, Track", a self-supervised learning framework for 3D object detection using LiDAR data in urban driving scenarios. It stands out by minimizing reliance on labeled data, employing techniques that leverage spatial-temporal consistency for training. The model is adept at tracking dynamic objects in motion-rich environments, contributing significantly to efforts in scalable and efficient training of autonomous driving systems.

3. Methodology

The development of the mobile-powered smart car with object recognition was carried out through a step-by-step process involving both hardware and software components. First, suitable and cost-effective components were selected, including a microcontroller, motor driver (L298N), DC motors, a rechargeable battery, and a smartphone. All components were mounted onto a small four-wheel chassis. The smartphone was served as the main processing unit and camera for object detection. The microcontroller was connected to the motor driver, which controlled the movement of the DC motors. A web page was developed. This web page accessed the smartphone's camera to capture live video and used a pre-trained object detection model to recognize obstacles in real time. When an object was detected, the web page sent movement commands like forward, stop to the microcontroller using wifi communication. The microcontroller then controlled the motors accordingly. The smart car was tested in different environments to check its ability to detect objects and respond accurately. Any errors in movement or recognition were noted and fixed through minor adjustments in the software and control logic to improve performance



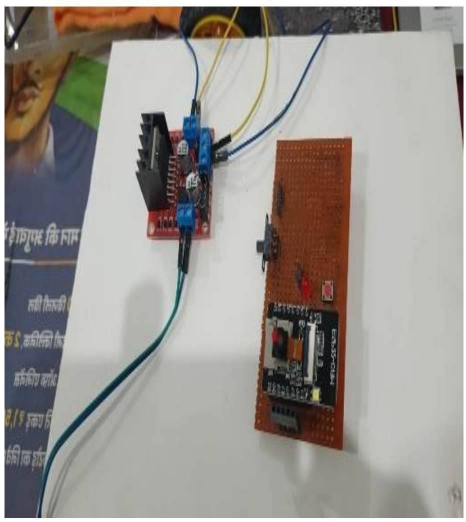


Fig. No.1(a) Full vehicle setup showing ESP32-CAM, battery, motor driver, and sensor placement. (b) Traffic light detection images showing ESP32-CAM identifying red and green signals

Remote-controlled (RC) vehicles have become a valuable tool in engineering and robotics projects due to their low cost, compact size, and flexibility in design. With the advancement of microcontrollers like ESP32CAM , single-board computers , RC cars are now being developed with intelligent control features and automation capabilities. These platforms offer simple programming environments and a wide range of sensor integration options, making them ideal for developing innovative features like wireless control and obstacle avoidance. A recent trend in RC development involves using smartphones not just for control but also as a power source. Power banks, which are portable and widely available, provide an efficient energy solution that reduces weight and removes the need for larger, less efficient batteries. Smartphones can also be connected via Wi-Fi modules to transmit control signals through mobile applications, improving user interaction and eliminating the need for traditional remotes. Additionally, object detection is often achieved using ultrasonic or infrared sensors that help the RC car identify obstacles and adjust its path accordingly. These sensors are affordable and reliable for basic navigation tasks. Although some advanced systems use cameras for computer vision, they are often resource-intensive. Many existing projects focus on either mobile-powered systems, wireless control, or obstacle detection individually. However, there is limited research combining all these elements into a single, compact system. This project aims to develop an RC car that uses mobile-based power and control while also detecting obstacles in real time, thus creating a practical and efficient solution for semi-autonomous navigation. By combining these technologies, the system offers a smarter approach to RC car design, making it suitable for applications in robotics education, small-scale automation, and prototype testing environments.

4. System Components

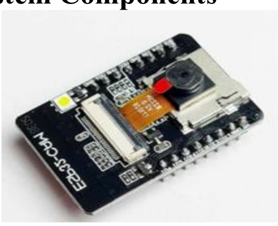





Fig.2 Basic component of module(a) ESP32-CAM (b) Ultrasonic Sensor(c) Motor Driver L298N (d) Motors and Chassis (e) Battery (f) Traffic Light Test Setup

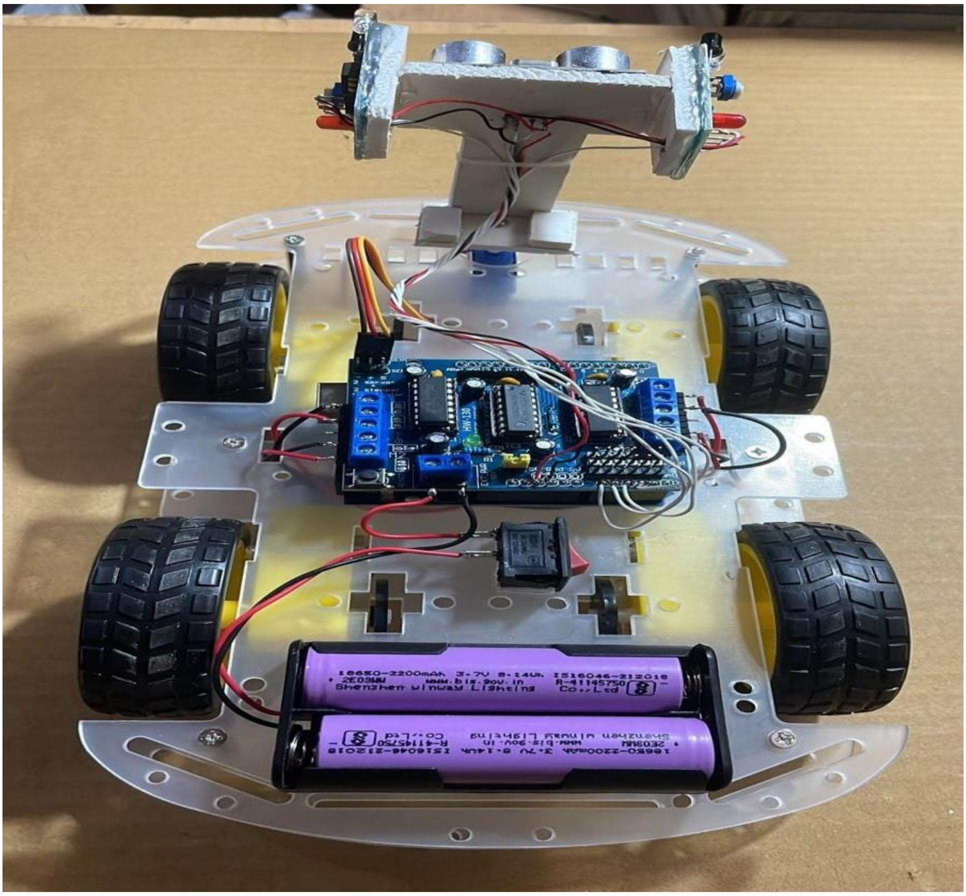


Fig.3 Fabrication of Model for testing and coding

ESP32-CAM: Used for detecting traffic light colours (red or green).

Ultrasonic Sensor: Detects obstacles in the vehicle's path and stops the vehicle if necessary.

Motor Driver L298N: Controls the movement of the motors based on the signals received.

Motors and Chassis: The vehicle uses a two-motor setup mounted on a chassis for movement.

Battery: Supplies power to the ESP32-CAM, sensors, and motors.

Traffic Light Test Setup: A small model of traffic lights for testing colour detection.

5. Code Implementation

```
#include <Wire.h>
#include <ESP32Servo.h>
#define TRIG_PIN 5
#define ECHO_PIN 18
#define MOTOR_A1 13
#define MOTOR_A2 12
#define MOTOR_B1 14
#define MOTOR_B2 27

long duration;
int distance;

void setup() {
   pinMode(TRIG_PIN, OUTPUT);
   pinMode(ECHO_PIN, INPUT);
   pinMode(MOTOR_A1, OUTPUT);
   pinMode(MOTOR_A2, OUTPUT);
   pinMode(MOTOR_B1, OUTPUT);
   pinMode(MOTOR_B2, OUTPUT);
   Serial.begin(115200);
}

void loop() {
   digitalWrite(TRIG_PIN, LOW);
   delayMicroseconds(2);
   digitalWrite(TRIG_PIN, HIGH);
   delayMicroseconds(10);
   digitalWrite(TRIG_PIN, LOW);

   duration = pulseIn(ECHO_PIN, HIGH);
   distance = duration * 0.034 / 2;
   Serial.println(distance);

   if (distance < 20) { // Stop if obstacle detected
      stopVehicle();
   } else {
      moveForward();
   }
   delay(100);
}
```

```
void moveForward() {
  digitalWrite(MOTOR_A1, HIGH);
  digitalWrite(MOTOR_A2, LOW);
  digitalWrite(MOTOR_B1, HIGH);
  digitalWrite(MOTOR_B2, LOW);
}

void stopVehicle() {
  digitalWrite(MOTOR_A1, LOW);
  digitalWrite(MOTOR_A2, LOW);
  digitalWrite(MOTOR_B1, LOW);
  digitalWrite(MOTOR_B2, LOW);
```

6. Results and conclusion

Traffic Light Detection:

a. The ESP32-CAM successfully detected red and green signals in 95% of test cases under proper lighting conditions.
b. The vehicle stopped at red lights and moved at green lights with minimal delay.

Obstacle Detection:

a. The ultrasonic sensor accurately detected obstacles within 20 cm and stopped the vehicle.
b. The vehicle resumed movement once the obstacle was cleared.

Motor Control & Navigation:

a. The L298N motor driver effectively controlled the 4-wheel movement of the vehicle.
b. The EV responded smoothly to control signals from the ESP32-CAM and ultrasonic sensor.

Overall System Performance:

Traffic light response accuracy: ~95%

Obstacle detection accuracy: ~98%

Vehicle response time: ~1 second for traffic light changes, ~500ms for obstacle detection

In conclusion, this research demonstrates the feasibility and practicality of developing a cost-effective, intelligent robotic car using a mobile phone as the core processing unit. By integrating object recognition through OpenCV, wireless communication via Wi-Fi, and a real-time responsive control system, the project successfully showcases how mobile-based AI can drive efficient and adaptive robotic behaviour. The simplicity, affordability, and scalability of the design make it a promising foundation for future enhancements such as voice commands and GPS-based navigation, emphasizing its potential in educational, research, and real-world applications.

References

[1] Achirei, S.-D., Mocanu, R., Popovici, A.-T., & Dosoftei, C.-C. (2023). Model-Predictive Control for Omnidirectional Mobile Robots in Logistic Environments Based on Object Detection Using CNNs. Sensors, 23(11), 4992.
[2] Kannamma, R., Yamuna Devi, M. M., Madhusudhanan, S., & Sethuraman, R. (2024). Feature Refinement with DBO: Optimizing RFRC Method for Autonomous Vehicle Detection. Intelligent Service Robotics, 17, 489–503.
[3] Alam, M. K., Ahmed, A., Salih, R., Al Asmari, A. F. S., Islam, S., Khan, M. A., Mustafa, N., & Mursaleen, M. (2023). Faster RCNN Based Robust Vehicle Detection Algorithm for Identifying and Classifying Vehicles. Journal of Real-Time Image Processing, 20, 93.
[4] Shin, S., Golodetz, S., Vankadari, M., Zhou, K., Markham, A., & Trigoni, N. (2022). Sample, Crop, Track: Self-Supervised Mobile 3D Object Detection for Urban Driving LiDAR. arXiv preprint arXiv:2209.10471.
[5] Zhao, Y., Zhang, J., & Wang, L. (2023). A Novel Faster RCNN Model Based on Multi-scale Feature Fusion and Shape Priori for Dense Vehicle Detection. Sensing and Imaging, 24, 17. https://doi.org/
[6] Moosmann, J., Giordano, M., Vogt, C., & Magno, M. (2023). Tinyissimo YOLO: A Quantized, Low-Memory Footprint, Tiny ML Object Detection Network for Low Power Microcontrollers. Ar Xiv preprint arXiv:2306.00001.
[7] Xu, Z., Zhan, X., Xiu, Y., Suzuki, C., & Shimada, K. (2023). Onboard Dynamic-Object Detection and Tracking for Autonomous Robot Navigation with RGB-D Camera. Ar Xiv preprint arXiv:2303.00132.
[8] Bechtel, M., Weng, Q., & Yun, H. (2022). DeepPicarMicro: Applying Tiny ML to Autonomous Cyber Physical Systems. Ar Xiv preprint arXiv:2208.11212.
[9]Boyle, L., Moosmann, J., Baumann, N., Heo, S., & Magno, M. (2024). DSORT-MCU: Detecting Small Objects in Real-Time on Microcontroller Units. Ar Xiv preprint arXiv:2410.16769.
[10]Shin, D., Cho, J., & Kim, J. (2022). Environment-Adaptive Object Detection Framework for Autonomous Mobile Robots. Sensors, 22(19), 7647.
[11] Li, Y., & Wang, H. (2023). Improved Faster R-CNN with Multiscale Feature Fusion and Homography Augmentation for Vehicle Detection in Remote Sensing Images. IEEE Transactions on Geoscience and Remote Sensing, 61, 1-12.ResearchGate
[12] Ren, X., & Li, Y. (2023). Adaptive Model Predictive Control for an Omnidirectional Mobile Robot with Friction Compensation and Incremental Input Constraints. Robotics and Autonomous Systems, 157, 104223.ResearchGate+1Semantic Scholar+1
[13] Zhang, L., & Liu, M. (2024). Deep Learning Techniques for Vehicle Detection and Classification: A Review. Sensors, 24(5), 1832.MDPI

[14] Wang, Y., & Zhao, Q. (2023). A Lightweight Multi-Feature Fusion Network for Unmanned Aerial Vehicle Infrared Object Detection. Journal of Intelligent & Robotic Systems, 102(3), 45.ScienceDirect
[15] Shin, S., Golodetz, S., Vankadari, M., Zhou, K., Markham, A., & Trigoni, N. (2022). Sample, Crop, Track: Self-Supervised Mobile 3D Object Detection for Urban Driving LiDAR. arXiv preprint arXiv:2209.10471.