International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 07 | July - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Gesture Controlled Rover with Visual Processing Using ESP32-CAM and Edge AI

Vijay Mahali[1], Sunil Kumar[1], Vishal Kumar[1] , Mrs. Anamika Kumari[2] , Mr.Hare krishna[2]

Department of Electronics and Communication Engineering
[1] B.Tech Scholar , RTCIT Ormanjhi Ranchi, Jharkhand, India
[2] Assistant Professor, RTCIT Ormanjhi Ranchi,Jharkhand India

---

Abstract - *This research introduces a novel, budget-friendly robotic system that integrates gesture-based control and real-time object detection on a mobile rover. The system uses a smartphone's gestures for directional control and the ESP32-CAM module for capturing visual data, processed locally using a compact machine learning model trained via Edge Impulse. This hybrid configuration allows seamless human-machine interaction and intelligent behavior in various environments. The rover is well-suited for surveillance, rescue operations, and intelligent automation, combining affordability with innovation.*

Keywords

Gesture-Controlled Rover, ESP32-CAM, Edge AI, Real-Time Object Detection, Embedded Machine Learning, Edge Impulse, Bluetooth Navigation, Visual Processing Robot, Arduino Robotics, Smart Surveillance, Low-Power AI, DIY Robotics, On-Device Inference.

1.Introduction

With the evolution of embedded systems and artificial intelligence, smart robots are becoming more accessible and applicable to real-world problems. This project presents a multi-functional rover that combines two powerful technologies: gesture-based wireless control and edge-based visual object recognition. The robot is designed to interpret human gestures sent from a smartphone and simultaneously analyze its surroundings using a camera and onboard AI.

By integrating machine learning directly onto the ESP32-CAM, the system avoids dependence on the internet or external servers, enabling it to operate autonomously even in remote or hazardous environments. The objective is to build a responsive and intelligent rover that can be deployed in applications like search-and-rescue, patrolling, or environment monitoring with minimal human interaction and high efficiency

2. Objectives

- Develop a real-time gesture-controlled robotic platform using mobile sensors and Bluetooth communication.

- Enable on-device visual recognition using compact, memory-efficient ML models.

- Achieve autonomous environmental awareness and intelligent responses using low-cost hardware.

- Promote scalable, low-power solutions for field deployment in areas with limited infrastructure.

3. Literature Review

The advancement of robotics and embedded systems has significantly influenced the development of smart autonomous platforms. In the domain of mobile robotics, traditional control mechanisms have primarily relied on manual interfaces such as joysticks, button-based remotes, or pre-programmed movement patterns. While these approaches offer basic functionality, they often lack flexibility, adaptability, and user-friendly operation in dynamic environments.

Recent years have witnessed an increasing interest in gesture-based control systems as a more intuitive human-machine interaction method. Gesture control, particularly through smartphones, leverages the widespread accessibility of mobile sensors to offer a wireless and natural interface for controlling robotic platforms. This method not only reduces hardware complexity but also enhances mobility and accessibility, making it suitable for non-expert users in diverse scenarios.

Parallel to this, the integration of computer vision in mobile robots has opened new avenues for real-time environmental awareness. Conventional image processing

techniques typically depend on high-performance computing resources or cloud servers, posing limitations in areas with unreliable network connectivity. However, the emergence of edge computing platforms, such as the ESP32-CAM, has enabled local visual processing with minimal hardware requirements.

The ESP32-CAM, known for its compact size, low cost, and wireless communication capabilities, has gained popularity for lightweight visual recognition tasks. When coupled with optimized machine learning models, it provides a practical solution for real-time object detection at the edge, eliminating the need for constant cloud dependency. Despite its resource constraints, techniques such as model quantization and the use of platforms like Edge Impulse have made it feasible to deploy trained AI models directly on these microcontrollers.

Although separate research efforts have explored gesture-controlled robots and edge-based visual recognition individually, there remains a noticeable gap in literature that effectively combines both technologies into a single, coherent robotic system. The integration of gesture-based control with onboard AI-driven visual processing offers a hybrid solution that enhances both user interaction and autonomous decision-making.

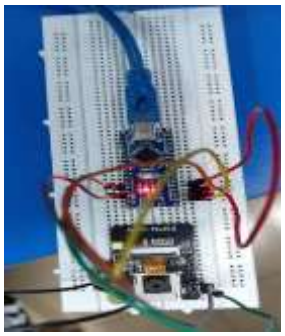
The present work addresses this gap by proposing a gesture-controlled rover that not only responds to user gestures via a smartphone interface but also actively perceives its surroundings using embedded visual processing. Unlike conventional remote-controlled robots, this system leverages the synergy between human inputs and autonomous environmental understanding, thereby improving functionality, user experience, and operational independence.

To the best of the author's knowledge, limited research exists that utilizes low-cost hardware such as ESP32-CAM for real-time visual recognition in tandem with smartphone-based gesture control on a mobile robotic platform. This project thus contributes a novel approach to bridging manual and autonomous robotic operation, targeting real-world applications where affordability, portability, and intelligent behavior are essential.

4. System Architecture

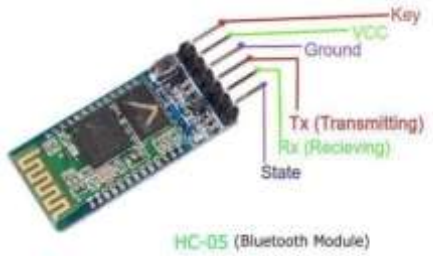
3.1 Key Hardware Components:

- ESP32-CAM: A compact microcontroller with a built-in camera (OV2640), Wi-Fi, and support for embedded. AI inference

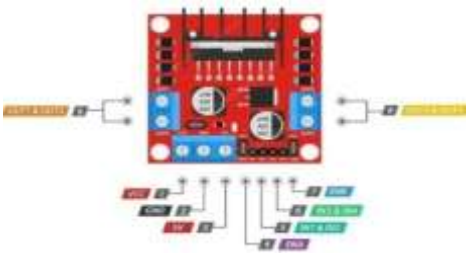


- Arduino Uno (ATmega328P): Serves as the motor control unit, interpreting Bluetooth commands from the.smartphone

- HC-05 Bluetooth Module: Enables gesture signal transmission between the smartphone and Arduino.



- L298N Dual H-Bridge Motor Driver: Drives the DC motors based on Arduino's signals.



- Lithium-ion Battery Pack (3 x 3.7V): Powers the entire system wirelessly.

- Smartphone with Gesture App: Acts as the user input device for controlling the rover.

3.2 Software Tools and Platforms:

- Arduino IDE: Used to program both ESP32-CAM and Arduino Uno.

- Edge Impulse Studio: A no-code/low-code platform for training and deploying ML models.

- TensorFlow Lite: Supports lightweight model deployment for edge inference.

- EloquentESP32CAM Library: Assists in image acquisition and web interface generation.

5. Methodology

5.1 Gesture-Based Rover Control:

The smartphone app interprets user hand movements or touchscreen gestures and transmits control signals (e.g., forward, backward, left, right) via Bluetooth. These commands are processed by the Arduino Uno, which then activates the motor driver to move the rover accordingly. This enables intuitive, wireless navigation using simple gestures.

5.2 Visual Processing with Embedded AI:

The ESP32-CAM captures live images using its built-in camera and processes them locally using a trained neural network.

- Data Collection: Images of specific object categories (e.g., bottle, person, obstacle) are captured using the ESP32-CAM.

- Model Training: Edge Impulse is used to label data, extract features, and train a CNN-based object detection model optimized for memory constraints.

- Model Deployment: The trained model is converted into C++ Arduino-compatible code and flashed into the ESP32-CAM.

- Inference & Action: The ESP32-CAM performs object recognition at the edge and can trigger responses such as alerts or avoidance.

6. Results and Analysis

- Accuracy: The ML model achieved an object detection accuracy between 75–85% in well-lit environments, with correct classification of objects like mobile phones, vegetables, and persons.

- Speed: The average inference speed was 1–2 frames per second (FPS), which is suitable for low-bandwidth robotics applications.

- Response Time: Gesture inputs showed sub-500 ms latency from smartphone to motor actuation.

- Power Efficiency: The system operated continuously for over 2 hours on a standard battery pack, validating its field readiness.

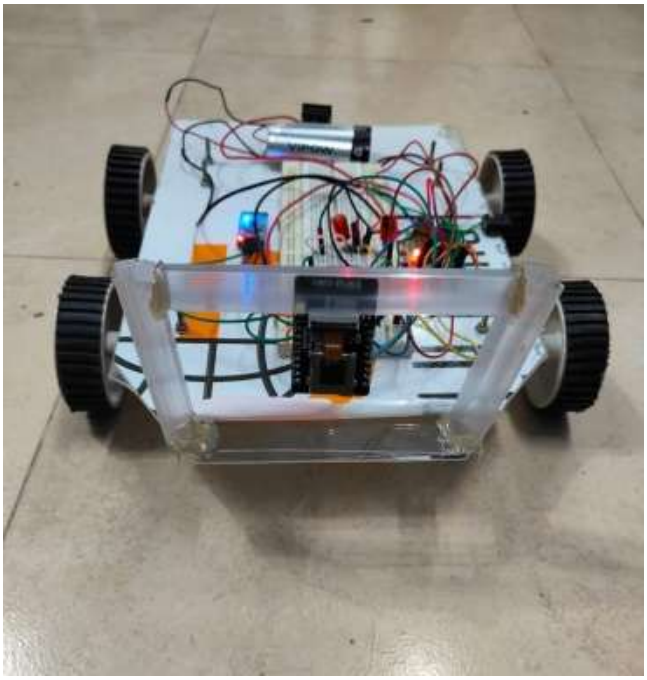


Fig.6.1 Front View

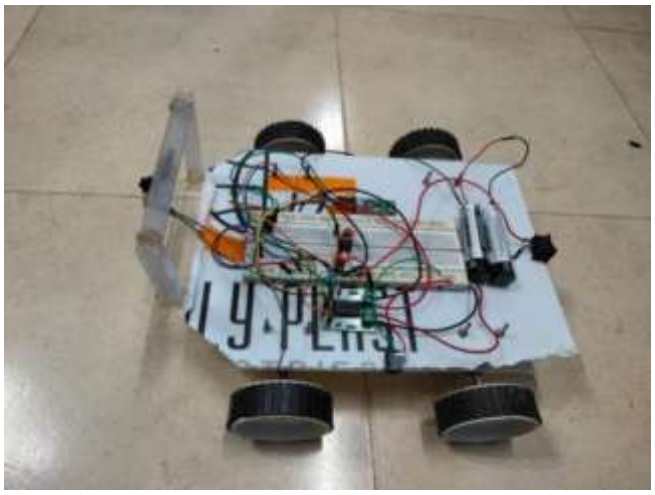


Fig.6.2 Side View

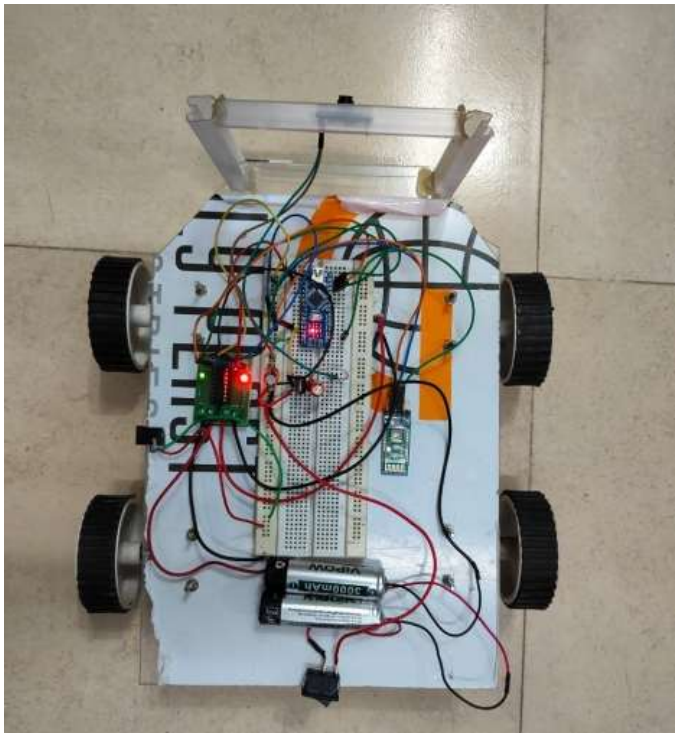


Fig.6.3. Top View

7. Applications

- Remote Surveillance: Acts as a mobile monitoring system for closed or restricted areas.

- Disaster Management: Useful in detecting survivors or hazards in post-disaster zones.

- Environmental Monitoring: Can detect anomalies or changes in natural surroundings (e.g., animal intrusion in farms).

- Military Patrol: Offers remote reconnaissance with reduced risk to personnel.

- Educational Robotics: Serves as a practical and affordable learning tool for robotics and embedded AI.

8. Limitations

- The current camera module supports limited resolution, which affects detection under poor lighting.

- Only a small number of object classes can be loaded due to memory constraints (typically <4MB).

- Frame rate is low, not ideal for fast-moving objects or high-speed detection. –

Gesture recognition is limited to Bluetooth range (approx. 10–15 meters).

9. Future Enhancements

- Hardware Upgrade: Replace ESP32-CAM with ESP32-S3 or Raspberry Pi Zero 2 W for enhanced processing and camera quality.

- Cloud Integration: Add cloud storage and dashboard for data logging and visualization.

- Long-Range Communication: Integrate LoRa or GSM modules for remote deployments.

- Voice/Gesture Fusion: Combine voice recognition with gestures for a multimodal interface.

- Advanced Navigation: Implement SLAM (Simultaneous Localization and Mapping) for autonomous path planning.

10. Conclusion

The project successfully demonstrates a unique and efficient system that bridges manual control with autonomous intelligence. By combining smartphone-based gesture navigation with real-time image recognition using edge computing, the rover performs robustly in various operational scenarios. The solution emphasizes accessibility, affordability, and modularity—making it a strong candidate for scalable deployment in education, security, and remote sensing.

The simplicity of the design, paired with its smart capabilities, opens up new avenues in DIY robotics and embedded machine learning, showing that impactful innovation doesn't always require expensive infrastructure.

11. References

I. S. Rajesh and P. Kumar, Introduction to Embedded Systems, McGraw Hill Education, 2020.

II. K. A. Krishnamurthy, Wireless Communication Principles and Applications, Tata McGraw Hill, 2019.

III. M. A. Mazidi, The 8051 and Embedded Systems: Using Assembly and C, Pearson Education, 2018.

IV. A. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," arXiv preprint arXiv:1704.04861, 2017.

V. J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2009.

VI. J. L. Raheja, R. Shyam, U. Kumar, and P. B. Prasad, "Real-Time Robotic Hand Control using Hand Gestures," in Proc. 2nd Int. Conf. on Machine Learning and Computing, 2010.

VII. P. B. Nayana and S. Kubakaddi, "Implementation of Hand Gesture Recognition Technique for HCI Using OpenCV," International Journal of Recent Developments in Engineering and Technology, vol. 2, no. 5, pp. 17–21, 2014.

VIII. M. K. Ahuja and A. Singh, "Static Vision-Based Hand Gesture Recognition Using Principal Component Analysis," in Proc. 2015 IEEE Int. Conf. on MOOCs, Innovation and Technology in Education (MITE).

IX. L. Bretzner, I. Laptev, and T. Lindeberg, "Hand Gesture Recognition Using Multi-Scale Colour Features, Hierarchical Models, and Particle Filtering," in Proc. 5th IEEE Int. Conf. on Automatic Face and Gesture Recognition, 2002.

X. F.-S. Chen, C.-M. Fu, and C.-L. Huang, "Hand Gesture Recognition Using a Real-Time Tracking Method and Hidden Markov Models," Image and Vision Computing, vol. 21, no. 8, pp. 745–758, 2003.

XI. L. Dipietro, A. M. Sabatini, and P. Dario, "A Survey of Glove-Based Systems and Their Applications," IEEE Transactions on Systems, Man, and Cybernetics – Part C: Applications and Reviews, vol. 38, no. 4, pp. 461–482, 2008.

XII. G. Dong, Y. Yan, and M. Xie, "Vision-Based Hand Gesture Recognition for Human-Vehicle Interaction," in Proc. Int. Conf. on Control, Automation and Computer Vision, 1998.

XIII. P. Garg, N. Aggarwal, and S. Sofat, "Vision-Based Hand Gesture Recognition," World Academy of Science, Engineering and Technology, vol. 49, pp. 972–977, 2009.