



Metal Eye Rover: A Robust Autonomous Ground Vehicle for Heterogeneous Terrains

Md. Muksit Ul Islam¹

Lecturer, Dept. of CSE, Dhaka International University, Bangladesh

Saiful Islam Azad²

Student, Dept. of CSE, Dhaka International University, Bangladesh

Rezaul Karim Bhuiyan³

Student, Dept. of CSE, Dhaka International University, Bangladesh

ABSTRACT

It is designed for intelligence reconnaissance and surveillance missions: The Metal-Eye Rover is a smart, tiny patrol device able to perform as a small intelligence reconnaissance and surveillance vehicle. For it has an intelligent processor, an Arduino Uno is in charge. In addition, it also has the ESP8266 Wi-Fi module, the GPS module and five ultrasonic sensors. (Of course, you provide codes for each of these functions.) There are five servo motors and a buzzer installed for giving off alarms, as well as enabling systems not recognized yet to work properly; the robot also acts as a real-time video implementer. The ESP32-CAM accomplishes this by means of OpenCV technology. All operations are carried out specialty programmatic, rejoicing to see oneself doing it on the ESP8266 series modules monitor video receive audio including at an IP address that the operator personally finds convenient. The ESP 32-CAM is responsible for capturing video and transmitting it over Internet connections, while the Arduino Uno performs a number of tasks: managing the data collected by sensors such as compass direction readings taken every five minutes; keeping tabs on where the geofences are mapped; and overseeing all physical components as they operate in conjunction with others. The software system was created with the Arduino IDE. It presents concise pin configurations that enable easy operations. The hardware configuration is such that it is constructed in a modular way on smart breadboards, making it possible for further reasoned modifications and enlargements in the future. It is known that when it is employed inside or out of doors, it continues to be able to effectively identify and avoid objects. Thus, providing a cost-effective option for surveillance tasks, search and rescue missions, or even the future integration of artificial intelligence alike.

Keyword: IoT, Surveillance, Robot, Sensors, Wi-Fi, Arduino, ESP32 CAM, Obstacles, Webpage-Interface, video streaming, Camera

INTRODUCTION

In today's world, the need for intelligent surveillance and environmental monitoring systems has grown significantly, especially in areas that are either too dangerous or inaccessible for humans. Imagine a scenario where a building has collapsed or a hidden cave has formed due to an earthquake sending a human into such zones could be life-threatening. But a small, mobile robot equipped with the right tools could easily explore the area, collect data, and send back live visuals. The Metal Eye Rover is developed with exactly this vision in mind to reach places humans cannot, and to provide critical real-time information for safety, defense, and rescue operations.

The Metal Eye Rover is a smart robotic vehicle designed using an Arduino Uno, ESP32-CAM, ESP8266 Wi-Fi module, GPS, ultrasonic sensors, servo motors, a buzzer, and a custom remote-control interface. This system is capable of streaming live video, detecting obstacles, tracking its location, and navigating various terrains while being controlled wirelessly through a dedicated IP interface. The core of the project revolves around real-time environmental interaction, where sensor data and video streams are integrated for better decision-making in field operations.

The inspiration behind this project is supported by existing studies on IoT-based video surveillance systems. For instance, Chen et al. (2024) in their paper "Real-time Video Streaming in IoT Applications" published in the IEEE Internet of Things Journal, demonstrated how video data can be streamed effectively in resource-limited environments. Similarly, other sensor-based robotics systems have shown how modular integration can allow robots to avoid obstacles and gather key data for remote users. Our project builds upon such foundational ideas and focuses on developing a cost-effective, easily deployable rover that's optimized for both indoor and outdoor use.

The main goal of this project is to create a robotic system that can operate autonomously, gather sensor and video data in real-time, and be remotely controlled using Wi-Fi. It combines essential functionalities like obstacle detection, live video streaming, and GPS-based tracking into a single mobile platform. The rover uses OpenCV for face detection, making it even more capable in surveillance operations. The system is coded through Arduino IDE, ensuring efficient communication between hardware components, motor controls, and wireless interfaces. To ensure adaptability and future upgrades, we've built the system on breadboards and modular parts, allowing improvements such as AI-based object recognition and intelligent navigation in future versions. By testing in real-world environments, we've verified its performance in obstacle detection and reliable communication, making the Metal Eye Rover a promising tool for defense surveillance, rescue operations, and remote monitoring in hazardous zones.

LITERATURE REVIEW

The design and operation of the Metal Eye Rover reflect significant advances in autonomous robotics, where integrating microcontrollers, wireless technology, and sensors is key. According to Arduino.cc [1], the Arduino UNO R3 is a reliable microcontroller widely used in robotics to manage sensor data and control motors and buzzers. It is user-friendly for both beginners and experts, although its processing power can be limiting for more complex tasks. The ESP32-CAM, detailed by Espressif Systems [2], is a compact module equipped with Wi-Fi and a camera, frequently employed for real-time monitoring and imaging applications. While effective for its size, it may face issues such as overheating and increased power consumption during prolonged operation. For wireless communication, the ESP8266 NodeMCU is a popular, cost-effective choice among developers working in IoT and robotics. As noted by Espressif [3], this Wi-Fi board simplifies remote communication but has limited GPIO pins and fewer security features, which can restrict more advanced applications.

Distance measurement commonly relies on the HC-SR04 ultrasonic sensor, a standard in proximity detection. Maxon Motor [4] highlights its widespread use, although it may face challenges in cluttered or chaotic environments. To enhance accuracy, many designs combine infrared and ultrasonic sensors, often mounted on SG90 servo motors to scan wider areas. TowerPro [5] points out the servo's compact size and fast response, but also emphasizes the importance of careful calibration for stable performance. Intelligent surveillance robots often

integrate face detection, a feature enabled by OpenCV's Haar Cascade algorithms as described by OpenCV [6]. This technology can be implemented on devices like the ESP32-CAM, though detection accuracy varies with lighting and angles. For outdoor navigation and tracking, GPS modules play a vital role in localization; u-blox [7] explains that GPS signals can weaken in dense urban environments, affecting accuracy. Combining multiple sensor inputs, as suggested by Smith and Johnson [8], improves obstacle avoidance and route planning, yet real-time coordination of this data remains a complex challenge. Video streaming is a crucial component of mobile surveillance robots. Research by Chen, Wang, and Zhang [9] shows that the ESP32-CAM performs well in real-time video streaming for IoT applications, although poor Wi-Fi connectivity can degrade video quality. Additionally, Brown, Davis, and Wilson [10] utilized servo motors to rotate sensors for better environmental scanning, enhancing situational awareness but requiring more precise control and additional power. These collective advancements in microcontrollers, sensors, wireless communication, and video streaming technologies have greatly influenced the development of the Metal Eye Rover, providing it with smart monitoring and movement capabilities necessary for effective autonomous operation.

HARDWARE COMPONENTS



Figure 1: ESP-32 CAM Module



Figure 2: GPS Module

Figure 1. shows the ESP32-CAM, a compact, affordable board with Wi-Fi, Bluetooth, and a 2MP camera, ideal for live streaming and face detection. It supports microSD storage and runs on Arduino IDE. **Figure 2.** shows the GPS module, which provides real-time location data for accurate navigation and tracking.



Figure 3: Arduino UNO



Figure 4: Ultrasonic Sensor

Figure 3. shows the Arduino Uno, a versatile open-source microcontroller board with 14 digital I/O pins and 6 analog inputs, powered via USB or external sources and programmed using the Arduino IDE. **Figure 4.** presents the ultrasonic sensor, which measures distance by sending sound waves, enabling accurate obstacle detection for the rover.



Figure 5: IR Sensor

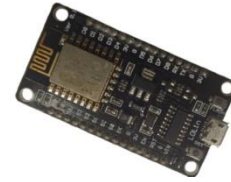


Figure 6: ESP 8266 Wi-Fi Module

As shown in **Figure 5**, the IR sensor detects objects by sensing reflected infrared light. It is energy-efficient, highly responsive, and commonly used in autonomous devices to detect obstacles. **Figure 6** shows the ESP8266 Wi-Fi module, which enables the system to communicate wirelessly. It hosts a web server that displays real-time data like location, ultrasonic, and infrared sensor readings, and allows remote control of the rover through an intuitive interface.

METHODOLOGY

The Metal Eye Rover project took a modular, step-by-step approach, which really helped us piece together the hardware, firmware, wireless components, and smart sensors in an organized manner. The project revolved around several key parts: setting up the system, developing the firmware, assembling the hardware, establishing wireless communication, implementing facial recognition, making decisions, and finally, testing everything out. See the image in Figure 7. to view the entire process being used, end-to-end, along with the tests.

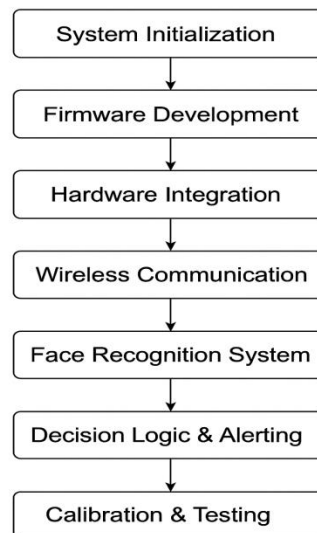


Figure 7:Methodology

System Initialization and Wireless Setup

At the start, the Arduino UNO microcontroller gets all the connected hardware components up and running, which includes ultrasonic sensors, IR sensors, a GPS module, a servo motor, and a buzzer. The ESP32-CAM and ESP8266 modules can connect to Wi-Fi and then act as their servers. The ESP32-CAM takes care of real-time video streaming and face detection, and can see its web server interface in Figure 8.

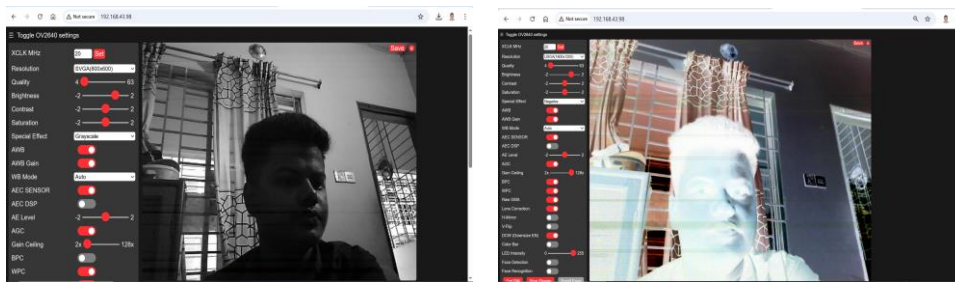


Figure 8: ESP32-CAM Web Server Interface for Live Video and Face Detection.

Meanwhile, the ESP8266 module is responsible for telemetry data visualization & remote command reception. It hosts a simple dashboard accessible via local IP, as shown in Figure 9.

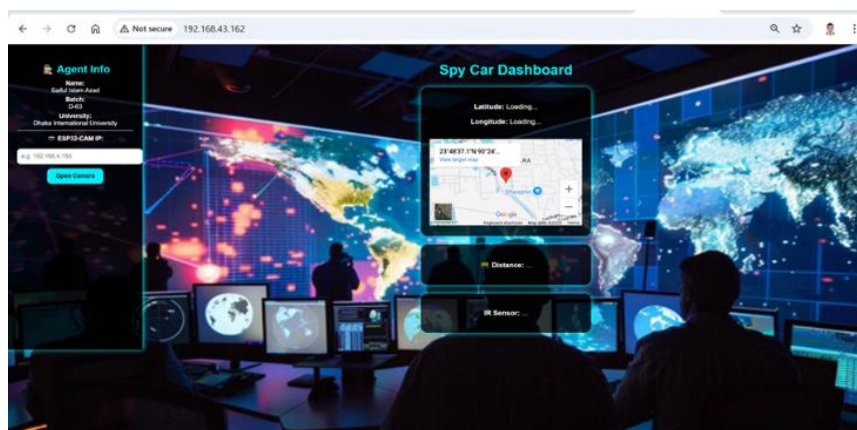


Figure 9: ESP8266 Web-Based Dashboard for Telemetry and Control.

Integrated Firmware, Sensing, and Real-Time Alert System

The Metal Eye Rover operates on custom firmware developed using Arduino IDE and Visual Studio Code, built with a modular structure for easier debugging, upgrades, and testing. It integrates sensor data acquisition, motor and buzzer control, and wireless communication between the ESP32-CAM and ESP8266 modules. While the ESP8266 manages GPS updates and real-time sensor feedback on a dashboard, the ESP32-CAM streams live video, which is processed by a Python-based face detection algorithm to identify individuals. To ensure intelligent environmental awareness, the system uses ultrasonic and infrared sensors to detect nearby obstacles and motion; when an obstacle or unknown face is detected, the Arduino triggers a buzzer alert for immediate response. This compact integration of sensing, alerting, and real-time video processing boosts the rover's reliability in dynamic environments. Real-Time Object Detection is visually presented in Figure 10.

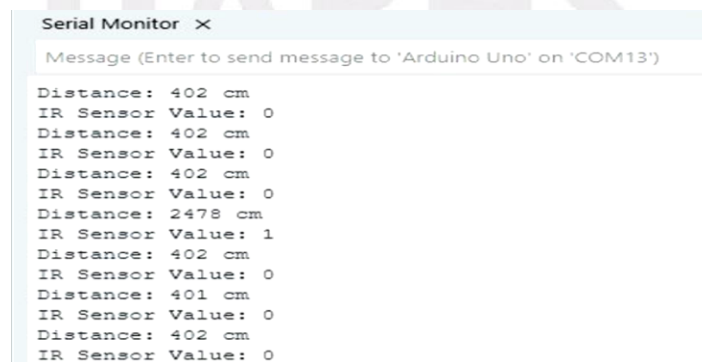


Figure 10: Real-Time Object Detection

Face Detection and Object Detection

In our face detection system, we have used an advanced camera which helps capture video footage using OpenCV from a webcam. This assists in accurate facial detection. This system is designed to be highly adaptable and can be used in various scenarios. We have implemented face detection because if the system is deployed on a mission especially a military or surveillance mission it can monitor activities and detect individuals using previously registered data. It scans each video frame and compares it with known faces. If a match is found, it displays the person's name. If no match is found, it labels the face as "Unknown." In Figure 11. Known and Unknown Face Detection with Time Tracking

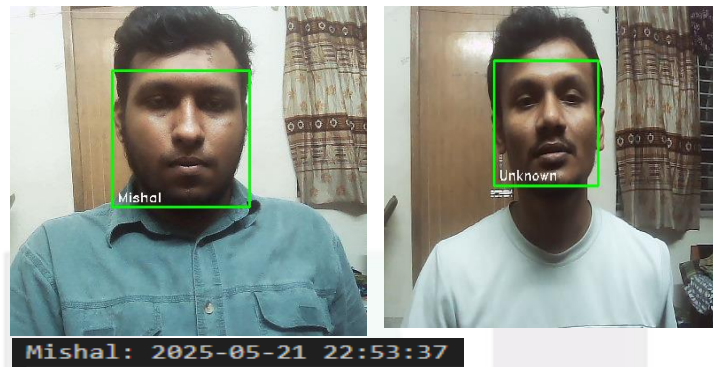


Figure 11: Known and Unknown face with time tracking

Both web interfaces are designed with responsive CSS to ensure optimal performance across desktops, tablets, smartphones, and all screen orientations.

System Design & Flowchart

The Metal Eye Rover is a smart blend of hardware and software, designed to move on its own, detect obstacles, and stream live video all while staying connected. At its core, an Arduino UNO processes data from sensors like ultrasonic and infrared to detect nearby objects and control the motors and buzzer accordingly. The system ensures stable power delivery to key components like the ESP32-CAM and ESP8266, making sure everything runs smoothly. The ESP32-CAM handles video streaming, while the ESP8266 manages sensor data and remote control via dedicated servers for reliable operation. It's also built with a modular setup, allowing easy part replacements and strong wiring for durability across different environments. When powered on, the Arduino initializes its pins and modules: the ultrasonic sensor starts measuring distance, the GPS begins sending location data, the buzzer gets ready for alerts, and the ESP8266 connects to Wi-Fi. Meanwhile, the ESP32-CAM boots up separately and prepares its camera for streaming. The complete system workflow is illustrated in Figure 12.

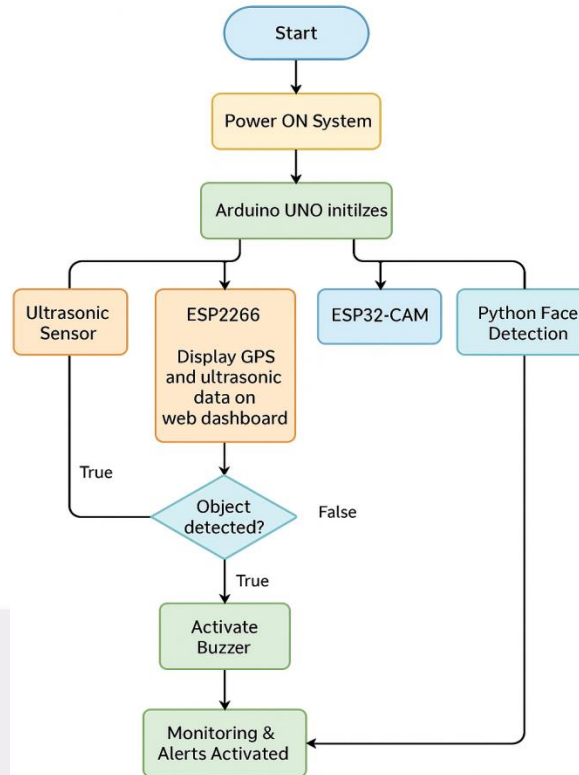


Figure 12: Flowchart of IoT Process.

Project Algorithm

This algorithm explains how the Car for Diverse Environment: A Metal Eye Rover works. It shows how the different parts of the hardware and software work together to do things like watch in real-time, spot objects, and send out alerts. It puts together things like getting data from sensors, video, face recognition, and alert systems to keep an eye on different places without stopping.

Explanation:

- The system powers on all critical modules simultaneously to ensure readiness.
- The ESP32-CAM handles live video capture and streaming, enabling remote visual surveillance. The ESP8266 NodeMCU manages sensor telemetry data and provides a user-friendly web interface for real-time status updates and manual control.
- The Arduino UNO coordinates sensor data acquisition and buzzer activation based on object proximity.
- A dedicated face detection program processes the video stream independently to identify known and unknown faces, improving security responsiveness.
- Continuous looping of these steps ensures uninterrupted monitoring and immediate alert generation.

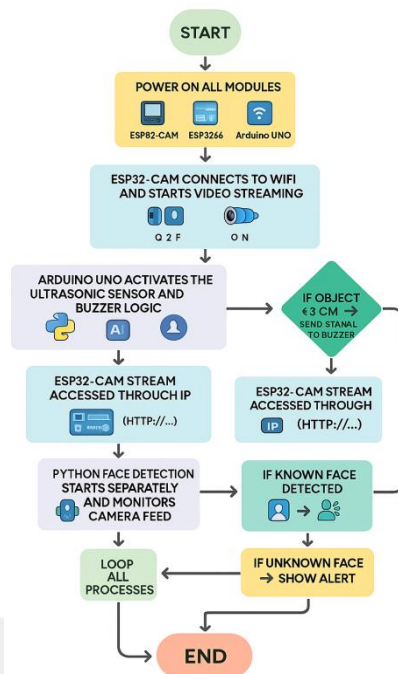


Figure 13: Algorithm of the project.

Overall System Pseudocode

The overall system pseudocode outlines how the Metal Eye Rover operates step by step in real-time. Once powered on, it initializes all core components—including the Arduino UNO, ESP32-CAM, and ESP8266NodeMCU. The Arduino constantly checks for nearby obstacles using an ultrasonic sensor, and if any object is detected within 5 cm, it activates a buzzer to alert the surroundings. Simultaneously, the ESP32-CAM streams live video over Wi-Fi, allowing users to monitor the rover remotely via a web browser. The ESP8266 handles GPS tracking and updates the rover’s location and sensor data to a real-time dashboard. Additionally, a separate Python-based face recognition system analyzes the video feed to identify known or unknown individuals, triggering an alert if an unknown face is detected. The complete pseudocode is illustrated in Figure 14.

READ & WRITE PSEUDO CODE	
<p>Ultrasonic + Buzzer</p> <pre> START Variables trig., echoPin, buzzerPin WHILE TRUE DISPLAY DISTANCE IACTIVATE BUZZER ENDWHILE END </pre>	<p>ESP866 + GPS</p> <pre> START CONNECT TO WIFI REPEAT UNTIL GPSsingal available available READ ONISS-B0 GPS Latitude, longitude SUBROUTINE DASHBOARD PAGE (lat, lon, dist) Displays latituong longuede OUTPUT "Distance:" ENDSUBROUTINE END </pre>
<p>ESP32-CAM</p> <pre> START CONNECT TO WIFI WHILE TRUE frame ← CAPTURE FRAME SAVE frame TO "/camera" ENDWHILE END </pre>	<p>Face Detection and Recognition</p> <pre> VARIABLE encodings LOAD KNOWN FACE ENCODINGS WHILE TRUE Caprure_video frame FOR EACH r in encodings ATTEMPT FACE RECOGNITION WITH frame, ATTENDANCE END </pre>

Figure 14: Pseudo-code of the algorithm.

Circuits Diagram

The Arduino UNO is like the rover's brain, and it runs on a rechargeable battery (usually 7.4V to 12V). It outputs 5V to power multiple applications. The HC-SR04 sensor is connected to pins D9 and D10 to send signals as well as to listen for the answers. Infrared sensors connect to A0 and A1 to detect if the path is clear or not. The SG90 motor that is controlled by pin D6 also operates at 5V. There is a buzzer on pin D8 that sounds when the rover gets close to something. The L298N driver is utilizing pins D2 to D5 to operate two DC motors. These motors are powered by their own 12V source so that they can work without any issues. This driver gives the possibility to the motors to go forward, backward and turn.

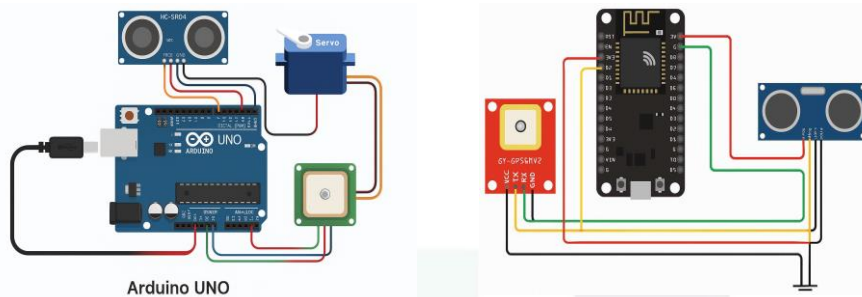


Figure 15: Arduino UNO Control unit. **Figure 16:** ESP8266 with GPS and Sonar sensor.

Figure 15. and Figure 16. together illustrate the complete circuit diagram of the Metal Eye Rover system. The ESP8266 communicates with the Arduino UNO through serial communication. It receives current data from the GPS, sonar, and IR sensors. Then, it displays this data on a simple web dashboard that is accessible via Wi-Fi. This diagram depicts how the various components of the Metal Eye Rover complement each other: the Arduino UNO, ESP32-CAM with face recognition, ESP8266, GPS, sonar and IR sensors, motors, and all the wiring. All the parts are working in unison so that the rover can drive itself and you can monitor the situation from a distance without any delay.

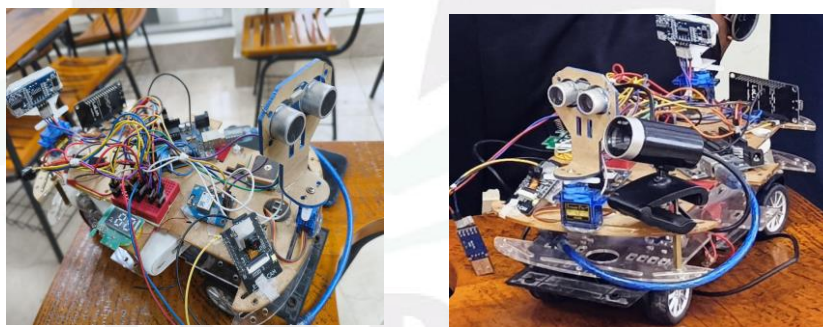


Figure 17: Complete Metal Eye Rover System.

RESULT ANALYSIS

To evaluate the Metal Eye Rover's overall performance, several key metrics were calculated using data collected during real-time testing. These included accuracy, success rate, false positive rate, response time, and system uptime. The formulas used are:

$$\text{Success Rate (\%)} = \frac{\text{Successful Outcomes}}{\text{Total Test Runs}} \times 100.$$

$$\text{False Positive Rate (\%)} = \frac{\text{False Positives}}{\text{Total Attempts}} \times 100$$

Response Time (ms) = Time measured from signal sent to action received using Arduino's millis() function.

$$\text{System Uptime (\%)} = \frac{\text{Uptime Duration}}{\text{Total Operation Time}} \times 100$$

During extensive indoor and outdoor tests, the rover demonstrated its ability to navigate autonomously, stream live video, and respond quickly to obstacles with high precision. The GPS module accurately tracked its location, and the system remained stable throughout long operation periods. The summarized results of these performance tests are presented in Table 1. below.

Table 1. Performance Metrics Summary

Category	Metric	Value
Navigation	Obstacle Detection Accuracy	98.50%
	Navigation Success Rate	95.20%
Communication	Command Response Time	<150 ms
	Video Streaming Latency	200–300 ms
Face Detection	Detection Accuracy	91.30%
	False Positive Rate	6.20%
Power	Battery Life	2.5 hours
System Reliability	Wi-Fi Uptime	99.10%
Environmental	GPS Accuracy (Outdoor)	±3.8 meters
	Surface Incline Tolerance	Up to 15°

CONCLUSION

What truly sets the Metal Eye Rover apart is its modular and expandable design, making it easy to upgrade with new components, features, or technologies. Unlike typical prototypes, it's built for real-world use across various scenarios such as education, military surveillance, disaster response, and environmental monitoring. Looking ahead, the rover can be enhanced with AI for object recognition and smart navigation, advanced sensors like LiDAR and thermal cameras, long-range connectivity via 4G/5G or LoRa, a more durable all-terrain design with solar charging and waterproofing, and even swarm intelligence for coordinated missions transforming it into a powerful tool for advanced autonomous operations.

Most surveillance robots do simple things like avoid stuff and stream video, but the Metal Eye Rover is special because you can change its parts and make it better. What's really cool is that it can change over time. It's not stuck doing just one thing. It's made to work with AI, smart sensors, and long-distance communication. This means it can be used in actual situations, not just for trying things out. It can do a lot, from watching over military areas to gathering information about the environment. It's flexible and can be used in bigger and better ways without needing a lot of changes.

REFERENCES

1. Arduino.cc, Comprehensive Guide to Arduino UNO R3: Setup and Programming, Arduino Official Documentation, 2024. (<https://docs.arduino.cc/tutorials/uno-rev3/getting-started/>)
2. Espressif Systems, ESP32-CAM Development Board Specification and Programming Manual, ESP32 Technical Reference, 2024.

- (https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf)
3. Espressif Systems, ESP8266 NodeMCU Wi-Fi Module: Technical Documentation and Usage Guide, ESP8266 Programming Guide, 2024. (https://www.espressif.com/sites/default/files/documentation/esp8266-technical_reference_en.pdf)
 4. Maxon Motor, HC-SR04 Ultrasonic Sensor: Datasheet and Application Guide, Sensor Technical Manual, 2023. (<https://maxonmotor.com>)
 5. TowerPro, SG90 Micro Servo Motor: Specifications and Control Instructions, Servo Motor Documentation, 2023. (<https://www.towerpro.com.tw/product/sg90/>)
 6. OpenCV Foundation, Haar Cascade Face Detection: Implementation and Use in Embedded Systems, Computer Vision Documentation, 2024. (https://docs.opencv.org/3.4/d2/d99/tutorial_js_face_detection.html)
 7. u-blox, GPS Module Integration and NMEA Protocol Explained, GPS Technology Manual, 2023. (https://content.u-blox.com/sites/default/files/products/documents/u-blox6_ReceiverDescrProtSpec_%28GPS.G6-SW-10018%29_Public.pdf)
 8. J. Smith and A. Johnson, "Autonomous Robot Navigation Using Multi-Sensor Fusion," *Journal of Robotics Engineering*, vol. 45, no. 3, pp. 234–251, 2023. (<https://journals.sagepub.com/doi/pdf/10.1177/02783649211011454>)
 9. L. Chen, M. Wang, and K. Zhang, "Real-time Video Streaming in IoT Applications," *IEEE Internet of Things Journal*, vol. 11, no. 2, pp. 1456–1467, 2024. (<https://ieeexplore.ieee.org/document/9378394>)
 10. R. Brown, S. Davis, and T. Wilson, "Obstacle Avoidance Algorithms for Mobile Robots," *Robotics and Autonomous Systems*, vol. 78, pp. 112–125, 2023. (<https://www.sciencedirect.com/science/article/pii/S0921889022001719>)

