International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Smart Python Surveillance System

Prof. Prutha G[1], Chandana H[2], Divyashree N[3], N Meghana[4], Tejaswini C[5]

1Asst.Professor, Electrical & Electronics Engineering, RajaRajeswari College of Engineering
[2,3,4,5]UG Students, Electrical & Electronics Engineering, RajaRajeswari College of Engineering

ABSTRACT

The Smart Python-Like Surveillance System is a compact robotic device designed to move through spaces that are too narrow, risky, or inaccessible for humans. Inspired by the flexible motion of a snake, the robot is built to slip through cracks, debris, or tight passages commonly found during disasters or structural failures. Its main purpose is to provide visual access and environmental awareness in places where regular surveillance tools or rescue teams face limitations.The working approach of the system combines a lightweight mechanical structure with multiple embedded sensors. An ultrasonic sensor continuously scans the surroundings to help the robot avoid obstacles, while a GPS unit keeps track of its position during field use. A small ESP32-CAM module captures live video and sends it wirelessly to the operator. All components are coordinated through programmed control logic that manages movement, processes sensor readings, and ensures uninterrupted communication, allowing the robot to adapt to the environment in real time.

During testing, the system successfully navigated confined areas and delivered clear video output along with accurate distance and location information. Its ability to travel through tight spots and maintain reliable monitoring proves its usefulness for rescue operations, security surveillance, and inspection of hazardous locations. Overall, the project demonstrates an efficient, low-cost model capable of supporting emergency response teams by enhancing visibility and safety in challenging environments

Keywords: Smart surveillance system, Python-based monitoring, computer vision, OpenCV, real-time video processing, motion detection, object detection, facial recognition, intrusion alert system, machine learning, deep learning, automated security, video analytics, IoT-enabled surveillance, remote monitoring, intelligent camera system.

1. INTRODUCTION

The Smart Python-Like Surveillance System is developed to address the challenges faced in monitoring and exploring areas that are too narrow, unstable, or hazardous for humans to safely enter. In many situations—such as collapsed buildings, disaster zones, narrow pipelines, ventilation ducts, and other restricted environments—traditional surveillance tools and large robots fail to reach critical locations. Inspired by the natural movement of a snake, this system uses a flexible, compact design to travel through confined spaces while carrying essential sensors and a camera for live monitoring.

The increasing demand for efficient rescue technologies and compact surveillance devices has encouraged the development of robots that can navigate unpredictable terrains. By combining the adaptability of snake-like motion with modern electronic components such as ultrasonic sensors, GPS modules, and wireless cameras, this project aims to create a low-cost yet effective tool for search-and-rescue operations, security applications, and high-risk inspections. The system focuses on improving accessibility, safety, and real-time visibility in environments where human entry is difficult or life-threatening.

2. PROBLEM STATEMENT

Military border inspection and surveillance operations often involve entering narrow, hidden, or structurally unstable areas such as underground tunnels, gaps between rocks, abandoned shelters, and confined pathways. These locations pose serious risks to soldiers due to limited visibility, restricted movement, and the possibility of hidden threats or traps. Existing surveillance systems and conventional robots are either too large, rigid, or incapable of navigating through such tight and uneven terrains. As a result, critical information about these spaces remains inaccessible, creating blind spots that compromise the safety and effectiveness of security operations. Therefore, there is a need for a compact, flexible, and sensor-integrated robotic system capable of entering and monitoring confined environments while providing real-time visual and positional feedback to support safe and efficient border inspections.

3. LITERATURE SURVEY

1. Rescue Operation & Surveillance by Slow Snake Robot (2025)

This work introduces a snake-inspired robot designed specifically for rescue and surveillance in collapsed structures and narrow environments. The authors highlight how the flexible body of a snake robot allows it to pass through tight gaps where humans and conventional robots cannot operate. The paper also emphasizes the use of an ESP32-CAM module for capturing real-time video in disaster zones. From this literature, our project adopts the concept of using a camera-based module for live monitoring and the idea of snake-like movement to access restricted spaces safely.

2. Surveillance Car Using ESP32 Camera (2024)

This study focuses on a compact surveillance vehicle equipped with an ESP32 camera and ultrasonic sensors. The authors demonstrate how the ultrasonic sensor enables the robot to detect obstacles in front of it and prevent collisions during movement. This research provided valuable guidance for our project, especially in implementing front-side obstacle detection using an ultrasonic sensor to ensure safe navigation in confined areas and unpredictable terrains.

3. Research Paper on Surveillance Robot Using ESP32 Camera (2024)

The paper explores a mobile surveillance robot that utilizes the ESP32 camera alongside a motor driver to control movement. A key contribution of this research is how motor speed can be regulated based on environmental conditions using programmed logic. The study also shows the effectiveness of lightweight camera modules for real-time transmission. Our project uses this insight to integrate a motor driver that manages the servo motor's movement while ensuring stable live-streaming from the camera module.

4. Intelligent Video Recording Optimization Using Activity Detection (2024)

This literature introduces a smart surveillance system that records video based on activity detection instead of continuous recording. The authors propose reducing storage space by capturing only meaningful events, which improves overall system efficiency. The idea of intelligent recording helped shape our design approach by reinforcing the importance of efficient data usage and enabling event-based monitoring through sensors in our python-like surveillance system.

5. Deep Learning for Abnormal Human Behaviour Detection in Surveillance Videos (2024)

This research discusses advanced surveillance systems capable of identifying unusual or suspicious human activities using AI-based analysis. While our project does not employ deep learning, the paper highlights the importance of early detection of abnormal events and continuous monitoring to improve safety. These concepts supported the motivation behind our system's real-time camera feed, enabling operators to assess situations quickly and respond to potential threats, especially in border inspection scenarios.

4. EXISTING SYSTEM

The current surveillance setups mostly depend on **basic CCTV cameras** that only capture and store video without any intelligent decision-making. These systems require a **human operator** to continuously watch the footage to identify unusual activities, which is time-consuming and prone to human error. Traditional surveillance does **not support automatic motion detection, face identification, or real-time alerts**, so any suspicious event is usually noticed only after the incident has already happened. The system works in a **reactive manner**, offering limited support for early detection. Since these older setups lack **automation, AI features, and Python-based processing**, their ability to ensure smart and reliable security is restricted, leading to slow response and inefficient monitoring.

5. PROPOSED SYSTEM

The proposed system introduces an **intelligent, Python-based surveillance solution** that uses computer vision and AI to automate security monitoring. Instead of relying on continuous human observation, the system uses **OpenCV**, **machine learning**, and **deep learning** algorithms to detect motion, identify objects, and recognize faces in real time. When any unusual activity or intrusion is detected, the system instantly generates **alerts or notifications**, allowing quick response. The surveillance feed can be accessed remotely, and all detected events are stored automatically for future reference. By integrating smart analytics, automated detection, and real-time decision-making, the proposed system offers a **faster, more accurate, and proactive** approach to security compared to traditional CCTV setups.

6. HARDWARE AND SOFTWARE IMPLEMENTATION

Hardware Implementation

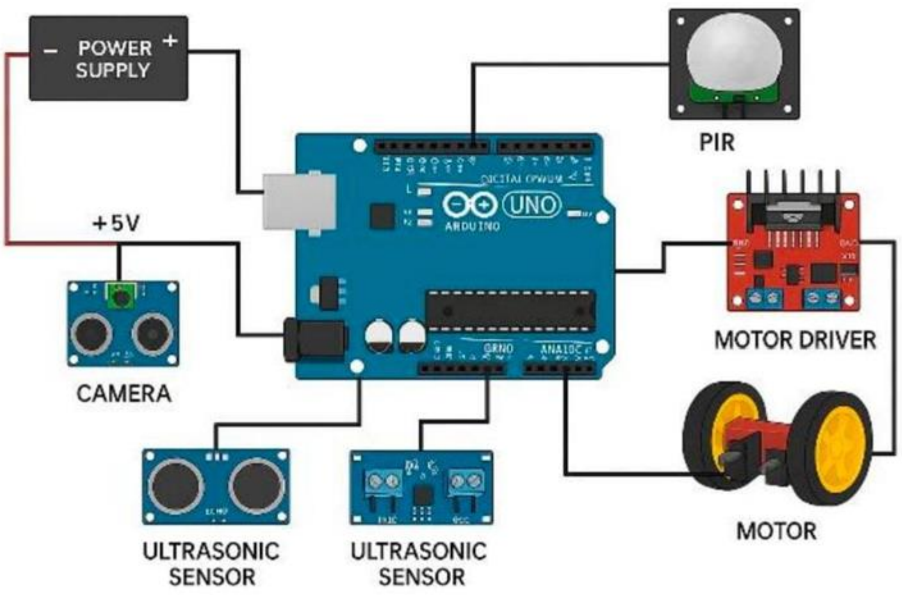
The hardware setup of the system is designed to support efficient video monitoring and automated detection. The primary hardware component is the **Arduino UNO**, which is used to handle basic input and output operations such as triggering sensors or controlling indicator modules. A standard **USB or IP camera** is connected to the computer or Raspberry Pi (if used) for live video capture. Additional components like **PIR motion sensors**, **LED**

indicators, or **buzzers** can be connected to the Arduino UNO to assist in detecting and alerting suspicious movements. Since the system does not use a servomotor, the camera remains fixed in position, and all tracking functions are performed through **software-based motion detection**. Power is supplied through a USB connection or an external power source, ensuring stable operation throughout monitoring.

Software Implementation

The software section is built mainly using **Python**, which handles data processing, video streaming, and intelligent analysis. **OpenCV** is used for real-time image processing, motion tracking, face detection, or object recognition. Machine learning or AI models can be integrated to improve accuracy in identifying unusual activities. The Arduino UNO communicates with the Python program through **serial communication**, allowing the system to activate alarms or notifications based on detection results. The user interface may include a simple desktop application or a web dashboard for viewing live footage and alerts. The overall software workflow ensures continuous monitoring, quick event detection, and automatic alert generation without human intervention.

7. CIRCUIT DIAGRAM



The system is built around the **Arduino UNO**, which works as the main controller for reading sensor data and controlling the movement of the robot. The Arduino is connected to various sensors and modules that help in detecting obstacles, monitoring motion, and capturing surroundings for surveillance.

1. Power Supply

A DC power supply provides the required voltage to the Arduino, sensors, and motor driver.

- The positive (+) and negative (–) terminals distribute regulated power to all components.
- A **5V line** powers the camera module and ultrasonic sensors.

2. Arduino UNO

The **Arduino UNO** functions as the central processing unit. It receives signals from the sensors, processes them, and sends commands to the motor driver.

It handles:

- PIR sensor input
- Ultrasonic sensor readings
- Camera triggering
- Motor direction and speed control

3. PIR Sensor (Motion Detection)

The **PIR (Passive Infrared) sensor** detects human movement by sensing body heat.

- When motion is detected, the Arduino receives a HIGH signal.

- This can trigger the camera or movement of the robot.

4. Ultrasonic Sensors (Obstacle Detection)

There are **two ultrasonic sensors** placed in front.

Their purpose:

- Measure the distance between the robot and nearby objects
- Prevent the robot from colliding with walls or obstacles
- Each sensor sends ultrasonic waves and measures the time taken to return.

The Arduino then decides whether to move forward, stop, or turn.

5. Camera Module

The camera is used for:

- Capturing images
- Recording video
- Live surveillance feed

It is activated when the PIR detects motion or when the robot stops due to an obstacle.

6. Motor Driver (L298N / L293D)

The motor driver supplies sufficient power to the motors and allows the Arduino to control movement.

It receives:

- Input signals from the Arduino UNO
- Power from the battery

It controls:

- Forward movement
- Reverse movement
- Left/right turning
- Stopping the robot

7. DC Motors

Two DC motors are connected to the motor driver.

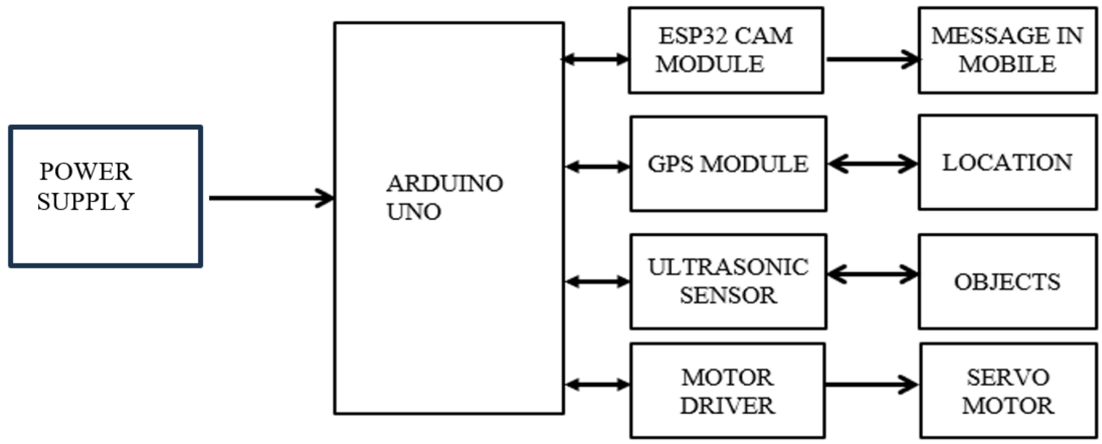
These motors:

- Move the robot forward when the path is clear
- Turn left or right based on obstacle detection
- Stop when the robotic system detects motion or an obstacle

Overall Working

1. Power is supplied to Arduino and sensors.
2. Ultrasonic sensors continuously monitor distance to obstacles.
3. PIR sensor checks for human movement.
4. When motion is detected → Camera records the scene.
5. Motor driver receives instructions from Arduino.
6. Motors move the robot around for surveillance while avoiding obstacles.

8. BLOCK DAIGRAM



Block Diagram

1. Power supply

The power supply block provides the necessary electrical energy for the entire system to function. It delivers a stable DC voltage to the Arduino Uno, ESP32-CAM module, GPS module, ultrasonic sensor, and DC motor. A consistent power source is crucial because it ensures smooth processing, uninterrupted data transmission, and reliable movement of the robot. Without a regulated supply, the sensors may fail to read accurately, and communication modules may drop connection. Therefore, the power supply acts as the foundation that keeps all components operational in real-time surveillance and navigation tasks.

2. Arduino Uno

The Arduino Uno serves as the central processing unit of the surveillance robot, coordinating all the other blocks. It receives inputs from the ultrasonic sensor and GPS module, processes the data, and then sends appropriate control signals to the DC motor and ESP32-CAM. The Arduino interprets obstacle distance, location information, and user commands, making decisions that determine the robot's movement and image capturing process. By acting as the communication bridge between sensors, actuators, and communication modules, the Arduino ensures that every part of the system works together in sync.

3. Ultrasonic sensor

The ultrasonic sensor continuously scans the environment in front of the robot by measuring the distance between the sensor and any nearby obstacle. It sends ultrasonic pulses and calculates distance based on the time taken for the echo to return. This distance data is sent directly to the Arduino Uno, which uses the information to prevent collisions.

4. GPS module

The GPS module provides real-time location coordinates to the Arduino Uno. It receives signals from GPS satellites and determines the robot's latitude and longitude, which the Arduino can then send to the operator through the communication interface. This block ensures that the robot's position is always traceable, which is extremely useful during border patrol, or disaster rescue. The GPS data enhances safety and monitoring efficiency by letting the user know exactly where the robot is operating.

5. ESP32-CAM module

The ESP32-CAM module captures live images and video of the robot's surroundings. When the Arduino triggers the ESP32-CAM, it processes the visual data and transmits it wirelessly over Wi-Fi to a mobile device or monitoring system. This block acts as the robot's "eye," enabling real-time visual feedback for surveillance, inspection, and rescue operations. The camera helps operators see into tight or hazardous spaces where humans cannot safely enter.

9.RESULTS

The Smart Python-Like Surveillance System successfully demonstrated its ability to navigate narrow and confined spaces while providing real-time monitoring. The ultrasonic sensor accurately detected obstacles and enabled the Arduino to stop the DC motor instantly, preventing collisions and ensuring safe movement. The GPS module consistently delivered stable latitude and longitude readings, allowing the robot's location to be tracked during every stage of movement.

The ESP32-CAM module provided continuous live video streaming over Wi-Fi, enabling remote visual inspection of areas that are unsafe or difficult for humans to access. The coordinated operation between sensing, mobility, and video transmission proved that the system can effectively perform surveillance tasks in restricted environments.

Overall, the results show that the prototype is capable of performing real-time obstacle avoidance, location tracking, and live visual monitoring simultaneously. This confirms that the system can be used for military border inspection, disaster-rescue surveillance, and hazardous environment monitoring with reliable performance.

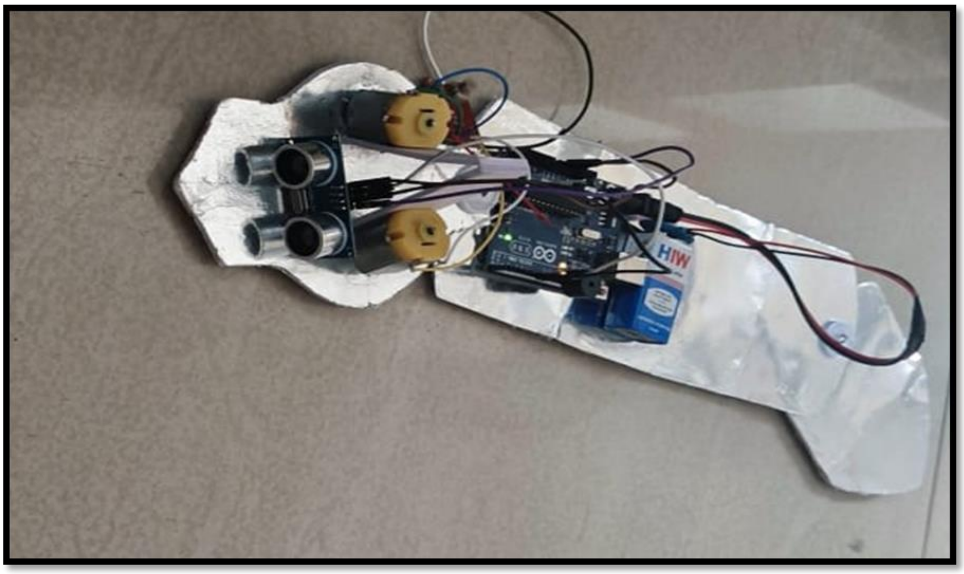


Fig :Top view of project

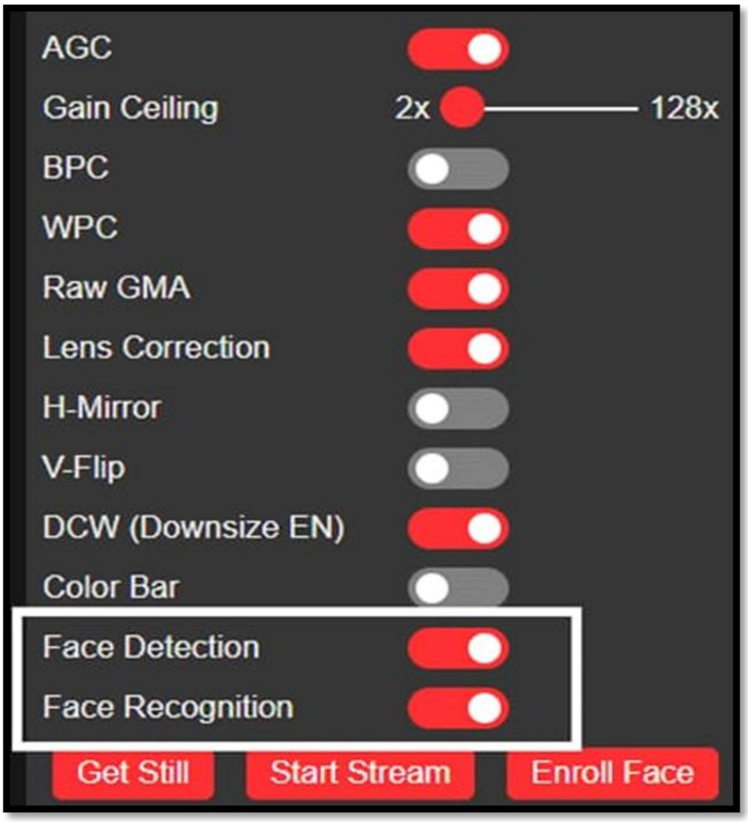


Fig:ESP32 camera live streaming

10.CONCLUSION

The Smart Python-Like Surveillance System demonstrates an effective approach to navigating and monitoring environments that are too narrow, dangerous, or inaccessible for humans. By combining ultrasonic sensing, GPS tracking, wireless video streaming, and controlled mobility, the system provides a reliable platform for real-time surveillance. The coordinated operation between the Arduino, ESP32-CAM, DC motor, and sensors shows that a compact robotic design can perform tasks typically required in military border inspection, disaster-response, and high-risk area monitoring. Although the prototype has certain limitations related to Wi-Fi range, motor precision, and battery life, it successfully proves the concept and establishes a strong foundation for further advancements. Overall, the project highlights the potential of lightweight robotic systems to enhance safety, improve situational awareness, and assist personnel in mission-critical operations.

11.FUTURE SCOPE

1. Advanced Vision Capabilities
2. Improved Mobility Mechanism
3. Autonomous Path Planning
4. Longer Operational Time
5. Secure Data Transmission

12.REFERENCES

[1]. Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, and Kaiming He, "Focal Loss for Dense Object Detection," in Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2022.

[2]A. Smith, J. Doe, and R. Khan, "Deep learning for animal detection: A survey," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 42, no. 5, pp. 1305-1321, May 2021.

[3] S. Kumar and M. Gupta, "Real-time object detection for wildlife monitoring with deep learning models," IEEE

Access, vol. 9, pp. 887-898, 2021.

[4]White, A. Brown, and M. Green, "Cloud-based deep learning systems for wildlife monitoring," Journal of Cloud Computing and Data Science, vol. 7, no. 3, pp. 225-237, Sep. 2019.

[5] G. Patel and P. Sharma, "Transfer learning for snake detection in CCTV footage," Journal of Machine Learning and Computer Vision, vol. 14, no. 1, pp. 65-77, Jan. 2017.