IOT PROJECT

AUTONOMOUS WILDLIFE MONITORING & POACHING PREVENTION

Team no: 56
Santhosh
Nikhil
Tilak

OBJECTIVES

- Detect Unauthorized Human Movement: Use infrared and motion sensors to detect human presence in restricted wildlife areas, signaling possible intrusion.
- Track and Locate Intruders: Real-time tracking helps locate the exact position of intruders for swift intervention by forest authorities.
- Listen for Poaching Sounds: Integrate sound sensors to pick up unusual noises like gunshots or chainsaws, which could signal poaching activity.
- Capture Visual Evidence: On detecting movement or sound, the system automatically captures photos or short videos to document the incident.
- Send Instant Alerts: Alert messages, along with evidence, are sent instantly to forest officers via wireless communication channels like GSM or LoRa.
- Enable Timely Action: The goal is to empower forest teams to respond quickly and efficiently, increasing chances of saving endangered wildlife.

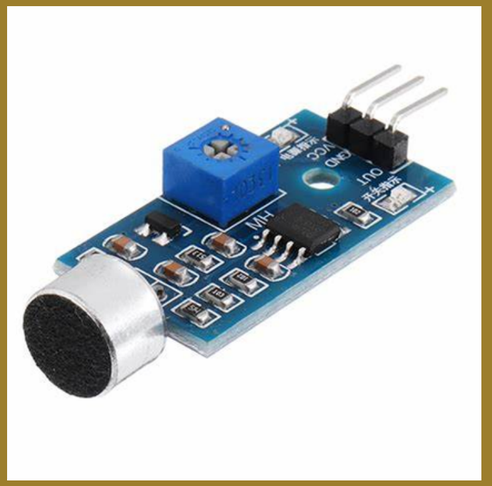
SENSORS USED

PASSIVE INFRARED

Detects infrared radiation from humans and animals, It helps in identifying movement in restricted areas.
It helps track poachers and prevent unauthorized entry.
The system enhances surveillance during nighttime or low-visibility conditions, making it ideal for real-time wildlife protection.

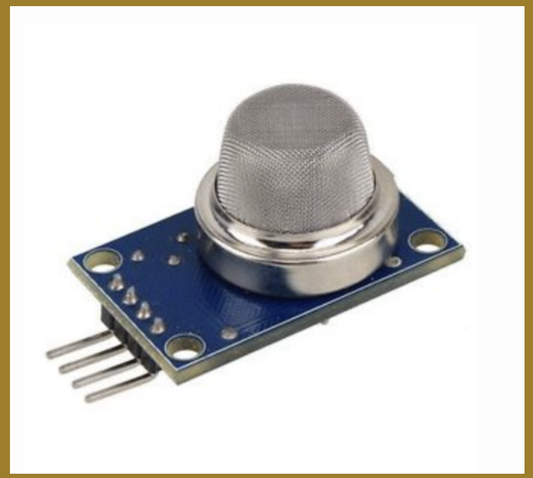
MICROPHONE SENSOR

The microphone sensor detects sudden loud noises in the area.
If the sound crosses a set threshold, it signals the ESP32 to activate the ESP32-CAM.
An image is captured, analyzed by YOLO, and alerts are sent if needed.



AIR QUALITY

The air quality sensor detects harmful gases in the environment.
When gas levels cross a set threshold, it sends a signal to the ESP32.
The ESP32 then activates a buzzer to alert about poor air quality.



PASSIVE INFRARED

The ESP32-CAM captures images when motion or sound is detected in the area.
It processes or sends the image data to a server or cloud for further analysis.
If a poacher or animal is detected, alerts are triggered through other modules.

PROJECT OVERVEIW

- Our project focuses on wildlife monitoring and poaching detection using ESP32-CAM and various sensors.
- A PIR sensor detects movement, triggering the camera to capture images in restricted zones.
- A microphone sensor monitors sound levels to detect suspicious noises like gunshots.
- Captured images are sent to a Flask server and stored in the uploads folder as latest.jpg.
- The Flask backend runs h3.py, where YOLOv8s analyzes the image to detect humans or animals.
- If a human is detected, the image is saved to the humans folder and a Twilio voice call is triggered to alert forest officers.
- The detection status is logged in a CSV file and can be accessed via an API endpoint (/status).
- A separate server (photo.py) serves a gallery webpage (gallery.html) displaying all human detection images.
- We faced challenges integrating ESP32-CAM with ESP32 and used an FTDI module to resolve communication issues.
- While effective, the system has limitations like low-light performance and basic sound differentiation

LESSONS LEARNED

• Motion Detection with PIR Sensors

Understood how PIR (Passive Infrared) sensors detect motion based on infrared radiation changes. Gained knowledge on adjusting sensitivity and delay time for better motion detection accuracy through calibration.

• Sound Detection using Microphone Sensor

Learned how the microphone sensor works by detecting changes in sound levels (measured in decibels). Understood how to set a proper sound threshold to trigger events only when noise exceeds a specific limit. Calibrated the microphone sensitivity to reduce false positives due to background noise or minor disturbances.

• Backend Development with Python & Flask

Learned the basics of Python and used the Flask framework to build a lightweight backend for processing images. Implemented and tested basic object detection models capable of distinguishing between humans and animals using the images captured by the ESP32-CAM.

• ESP32-CAM Configuration & Image Upload

Learned how to access the ESP32-CAM's IP address on the local network. Successfully configured the module to capture and upload images directly to a web server, enabling real-time remote monitoring.

• Hardware Integration & Wiring

Learned the correct wiring of the ESP32-CAM, including connections with the PIR and microphone sensors. Understood the required voltage levels: 5V for the PIR sensor and typically 3.3V–5V for the microphone module, ensuring compatibility and avoiding damage.

• Programming ESP32-CAM using FTDI

Gained hands-on experience in using the FTDI converter to program and communicate with the ESP32-CAM, overcoming direct USB connectivity issues. Successfully operated the ESP32-CAM using the Arduino IDE, including board selection, boot mode, and code uploading.

CHALLENGES FACED

- Limited Sound Detection Accuracy: The microphone sensor used in the project can only measure sound levels in decibels. It cannot accurately distinguish between different sound types like gunshots, animal cries, or human voices, limiting precise sound-based event detection.
- Low-Light Camera Limitations: The ESP32-CAM module features a low-resolution camera that struggles in low-light environments. Without night vision or infrared support, it fails to capture clear, usable images during nighttime, reducing its effectiveness for 24/7 monitoring.
- Basic PIR Detection Capability: The PIR sensor detects motion but does not provide the exact location or direction of the movement. This lack of spatial awareness makes it difficult to pinpoint the specific area of activity.
- ESP32-CAM Interfacing Issues: During implementation, the ESP32-CAM did not function correctly when connected directly to the ESP32 module. It failed to operate as intended.
- Use of FTDI Converter for Stability: To solve the interfacing problem, an FTDI (USB-to-Serial) converter was used. This allowed successful communication and proper functioning of the ESP32-CAM module

Human Detection

Air Pollution

Sound Detection

MONITOR WILDLIFE, PREVENT POACHING, AND PROTECT NATURAL HABITATS

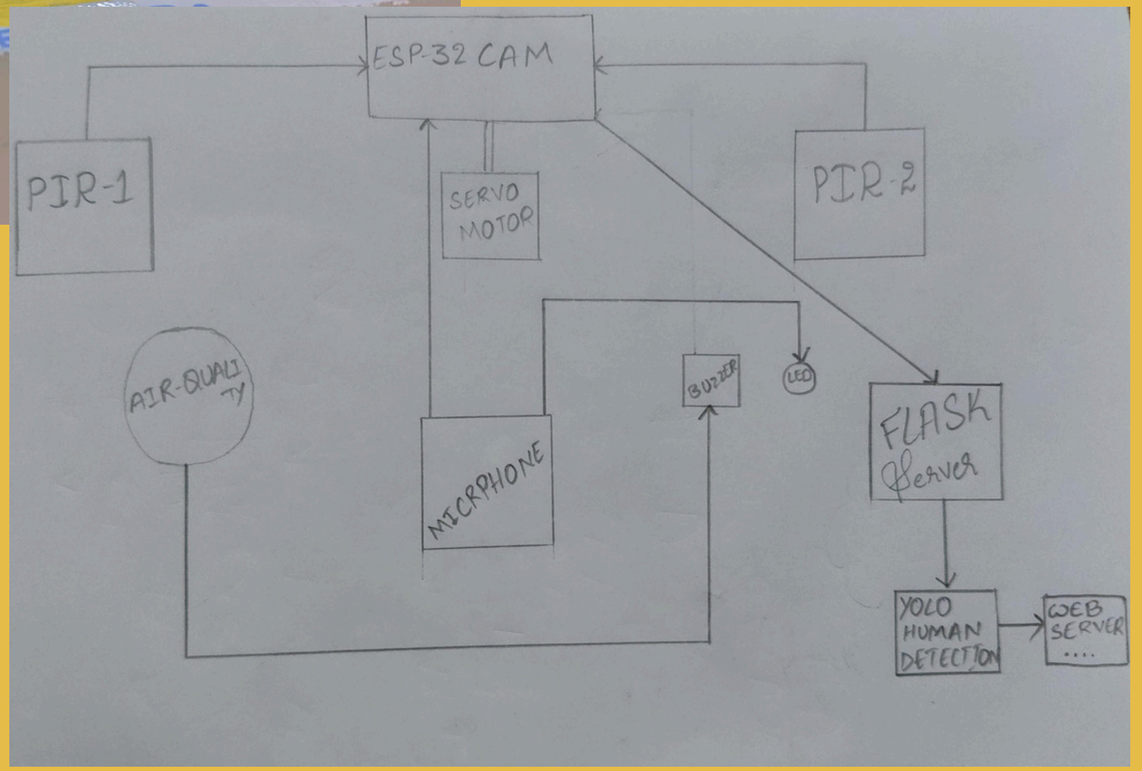
- **SMART SENSOR DEPLOYMENT**

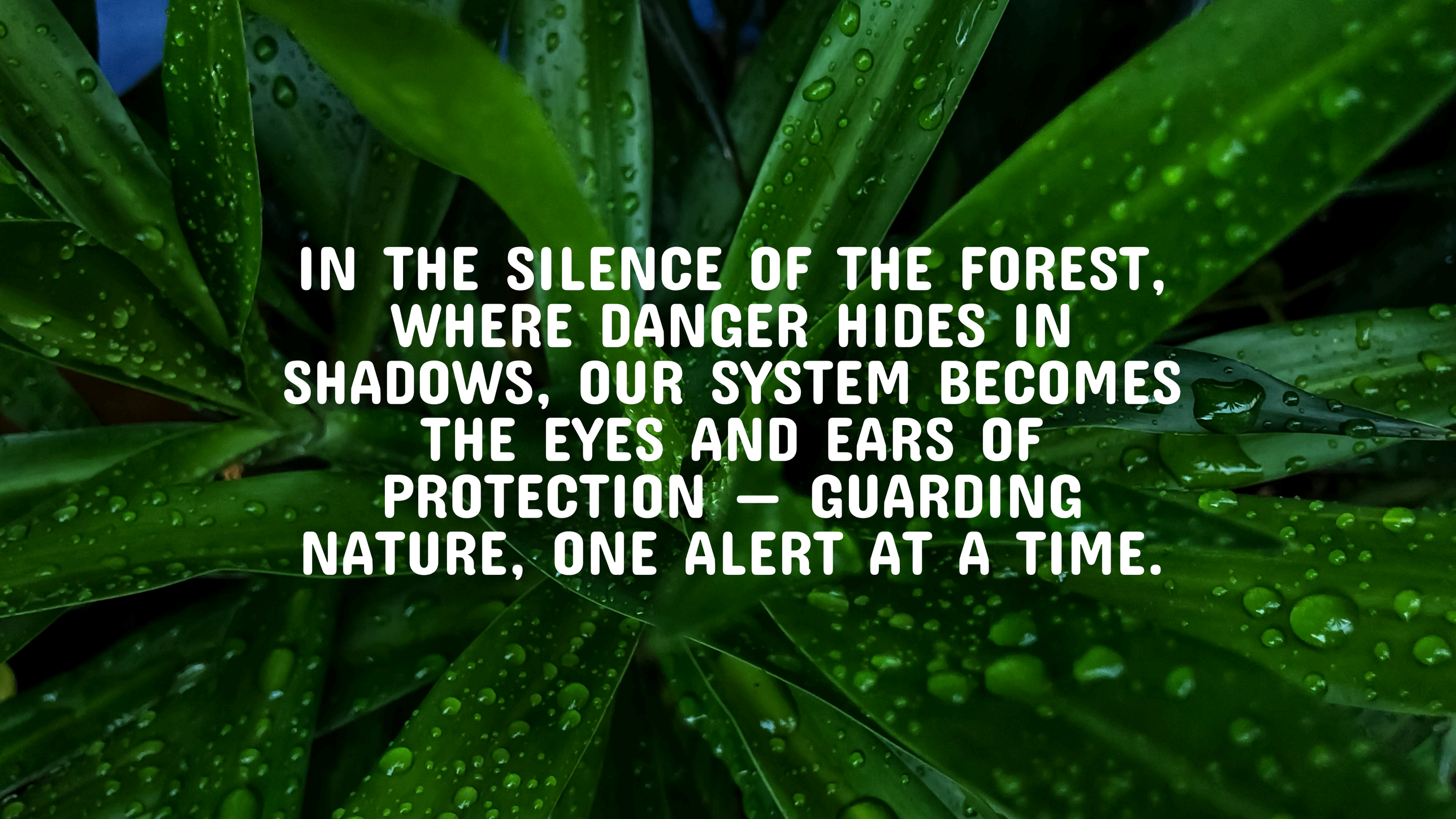
We used PIR and MIC sensors on ESP32 to detect motion or sound in wildlife areas. The setup captures real-time events and triggers image capture when activity is detected.

- **AI-BASED DETECTION SYSTEM**

Captured images are sent to a Flask server, where YOLO detects humans or animals. Based on this, alerts are generated to prevent poaching and ensure wildlife safety

Mini Project




IN THE SILENCE OF THE FOREST,
WHERE DANGER HIDES IN
SHADOWS, OUR SYSTEM BECOMES
THE EYES AND EARS OF
PROTECTION — GUARDING
NATURE, ONE ALERT AT A TIME.

THANK YOU

+91 9236167118

ANIMALSAFETY@GMAIL.COM

FOREST