Night Patrol Robot

Mrs.Likitha S[1], Ajay Gowda M[2], Deekshitha S[3], Kushala P[4], Kushi Patil[5]

Abstract- *The AI-Driven Night Patrol Robot integrates autonomous navigation and AI-powered surveillance for enhanced night-time security. It utilizes an Arduino Uno, motor drivers, and ultra sonic sensors for path finding, while an ESP32 CAM and YOLOv11 model detect threats like theft and weapon possession. Alerts are sent via Telegram, ensuring prompt responses.*

Keywords- AI-powered surveillance, autonomous navigation, night patrol robot, real-time threat detection, YOLOv11.

I. INTRODUCTION

Ensuring effective night-time security is a growing concern, as traditional systems often rely on human intervention and static monitoring, leading to inefficiencies and delays. The AI-Driven Night Patrol Robot offers an innovative solution to these challenges by integrating autonomous navigation and real-time AI-powered surveillance. This robot aims to revolutionize security practices by providing consistent, proactive monitoring without the need for human oversight.

The system consists of two core modules: autonomous navigation and AI surveillance. The autonomous navigation module is powered by an Arduino Uno, motor drivers, and an ultrasonic sensor, enabling the robot to detect obstacles and dynamically adjust its path, making it suitable for diverse environments such as residential complexes and industrial areas. The AI surveillance module utilizes an ESP32 CAM and a YOLOv11 model to detect and identify suspicious activities, such as theft, fighting, and weapon possession, providing real-time monitoring.

When suspicious activities are detected, the system instantly sends alerts, complete with images and activity labels, to a designated Telegram group. This immediate response capability empowers security personnel to take quick action, potentially preventing further escalation of threats. By automating both navigation and monitoring, the Night Patrol Robot provides a scalable, efficient, and reliable solution for modern security needs. Its modular design ensures adaptability for future upgrades, addressing evolving challenges in surveillance technology.

II. LITERATURESURVEY

1. "Autonomous Mobile Robot for Surveillance"

Authors: John A. Smith, Emily Roberts
*PublishedYear:*2018

This research introduces an autonomous mobile robot designed for indoor surveillance. The robot integrates sensors for navigation and cameras for real-time video recording, using basic image processing techniques to detect motion and identify potential threats. It follows a predefined patrol path to ensure thorough coverage of the target area. The system offers a low-cost solution with easy implementation for small-scale indoor environments. However, its limitations include being restricted to indoor use due to the lack of ruggedness for outdoor conditions and the basic image processing, which hinders its ability to detect complex activities or threats.

2. "AI-Powered Surveillance Systems for Real-Time Threat Detection"

Authors: Michael T. Johnson, Sarah Lee
Published Year: 2020

This study focuses on the application of AI for real-time surveillance, particularly in urban security. The system uses neural networks to analyze video feeds and detect specific activities such as unauthorized entry and violence. Its high accuracy in detecting predefined activities makes it adaptable to various environments and threat types. Despite its advantages, the system requires significant computational resources and depends on extensive training datasets for effectiveness. This reliance on large datasets can make it less efficient in environments with limited data availability, posing challenges for broader deployment in resource- constrained settings.

3. "Obstacle Detection and Navigation in Autonomous Robots"

Authors: Alice Brown, Richard Kumar
Published Year: 2017

This paper explores the integration of ultrasonic sensors with microcontrollers for obstacle detection and avoidance in mobile robots. The proposed system ensures dynamic navigation in cluttered environments, with successful real-world and simulated tests. The system is reliable in detecting obstacles at close range and effective in collision avoidance during navigation. However, it has limitations, such as its ineffectiveness in detecting transparent or very small obstacles and its limited range, which reduces its performance

in larger or more complex environments. These limitations need to be addressed for broader applicability in diverse real-world scenarios.

4. "Real-Time Threat Detection Using YOLO Algorithm"
Authors: David Green, Laura Adams
Published Year: 2021

This work demonstrates the application of the YOLO (You Only Look Once) algorithm in real-time object detection for surveillance systems. YOLO achieves fast and accurate detection of multiple objects, including threats such as weapons and unauthorized personnel. It is suitable for live video feeds, providing high-speed processing. Despite its efficiency, the system requires powerful GPUs for optimal performance, which can be a limiting factor in resource-constrained environments. Additionally, it struggles with detecting objects under low-light conditions or at extreme angles, presenting challenges in certain surveillance environments where these factors are prevalent.

5. "Integration of Robotics and AI for Enhanced Surveillance"
Authors: Paul W. Edwards, Megan Young
Published Year: 2019

This paper discusses the integration of robotics and AI to create advanced surveillance systems. The system combines autonomous navigation with AI-driven video analysis to identify threats and enhance security measures. Additionally, the system includes an alert mechanism for real-time communication with authorities, ensuring timely responses to threats. While the system offers a comprehensive solution that combines mobility with AI analysis, it faces challenges in terms of high implementation costs due to the complexity of hardware and software requirements. Moreover, maintenance of such an integrated system can be challenging due to the intricacies involved in merging robotics and AI technologies.

III. METHODOLOGY

1. Hardware and Sensor Setup

- Components Assembly: The Arduino Uno, ultrasonic sensors, motor drivers, and ESP32 CAM are assembled onto the robot chassis.
- Calibration: Ultrasonic sensors are calibrated to detect obstacles within a range of 20 cm.
- Testing: Individual hardware components are tested to ensure proper functionality before integration.

2. Software Development

- Programming Environment: The Arduino IDE is used to program the microcontroller for interpreting sensor data and controlling the motors.
- Obstacle Avoidance Logic:
  - The HC-SR04 ultrasonic sensor continuously measures distances.
  - Detected obstacles trigger calculations for alternate routes, and motor drivers adjust the robot's direction accordingly.

3. AI Model Training

- Dataset Preparation:
  - Annotated datasets of predefined threats (e.g., weapons, theft, fighting) are prepared using Label Studio.
  - Images are labeled and categorized for model training.
- Model Development:
  - The YOLOv11 model is used for object detection and threat identification due to its high accuracy and efficiency.
  - Training is conducted using Google GPUs to ensure robust model performance.
- Optimization:
  - Detection thresholds and bounding box precision are fine-tuned for reliable threat detection.

4. Real-Time Video Analysis

- ESP32 CAM Integration:
  - Configured to capture real-time video frames.
  - Frames are transmitted to the YOLOv11 model for processing.
- Threat Detection:
  - Captured frames are analyzed, and bounding boxes are drawn around detected objects.
  - Labels are assigned to identified threats for interpretation.

5. Alert Mechanism

- **Notification System**:

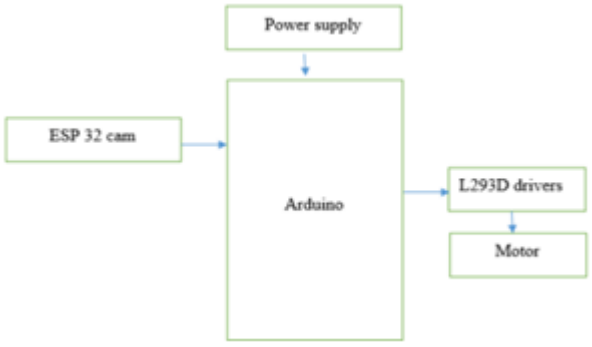
- Detected threats trigger alerts via Telegram.
- Alerts include images of the threat and activity labels for immediate action.

6. Testing and Deployment

- **System Testing**:
  - The navigation system is tested in different environments to ensure adaptability.
  - The surveillance module is tested under various scenarios to verify detection accuracy and responsiveness.
- *Deployment:*
  - The fully integrated system is deployed and monitored to ensure reliability in real- world applications.

IV. SYSTEM ARCHITECTURE

The architecture of the AI-Driven Night Patrol Robot is a hybrid design integrating autonomous navigation with AI-powered surveillance. At its core, the system is divided into two main components: the hardware navigation system and the AI-driven surveillance module. The hardware navigation system, built around an Arduino Uno microcontroller, controls the robot's movements using inputs from ultrasonic sensors and motor drivers. This subsystem is responsible for ensuring smooth and efficient patrolling by dynamically adjusting the robot's path based on real-time obstacle detection. The surveillance module, powered by the ESP32 CAM, captures live video frames which are processed by a YOLOv11 AI model deployed on a computing system. The processed frames help identify suspicious activities, and if any threats are detected, the system triggers an alert mechanism, sending notifications via Telegram. The architecture ensures seamless integration between navigation and surveillance, enabling real-time data processing and response while minimizing latency.



V. HARDWAREANDSOFTWARE REQUIREMENTS

1. Hardware Requirements

The hardware components ensure smooth functionality and performance for the Night Patrol Robot:

1. **Processor(CPU):**
   - **Requirement:** Intel Core i5/i7 (or equivalent) or higher.
   - **Purpose:** Handles image preprocessing, modelinference,andoverallcomputational tasks. Multi-core processing enhances system responsiveness.
2. **GraphicsProcessingUnit(GPU):**
   - **Requirement:** NVIDIA GTX 1060 or higher (recommended for training; not mandatory for inference).
   - **Purpose:** Accelerates deep learning processesliketrainingconvolutionalneural networks (CNNs).
3. **RAM (Memory):**
   - **Requirement:** Minimum 8 GB (16 GB or more recommended).
   - **Purpose:** Supports handling large datasets, image preprocessing, and real-time processing during deployment.
4. **Storage:**
   - **Requirement:**500GBHDDorSSD(SSD recommended).
   - **Purpose:** Stores images, model weights, datasets, and results, ensuring quick access during operations.
5. **Power Supply:**
   - **Requirement:** Rechargeable battery pack.
   - **Purpose:** Provides stable power for extended operational hours.
6. **Sensors and Peripherals:**
   - **Components:**
     - Arduino Uno microcontroller: Central processing for navigation.
     - Motor drivers: Enables robot movement (forward, reverse, and direction).
     - Ultrasonic sensor: Obstacle detection for collision avoidance.
     - ESP32 CAM module: Real-time image capture.
     - Buzzer: Immediate threat alerts.
7. **Network:**

- **Requirement:** High-speed internet connection.
- **Purpose:** Facilitates real-time data transmission and alerts.

2. Software Requirements

The software environment integrates development and operational tools to manage the robot's functionality:

1. **Operating System:**
   - Requirement: Windows 10/11 or Linux (Ubuntu recommendation).
   - Purpose: Provides a stable platform for development and deployment.
   - I functionalities.
2. **Programming Languages:**
   - Requirement: Python 3.7 or higher.
   - Purpose: Primary language for AI and surveillance modules.
3. **IDE (Integrated Development Environment):**
   - Requirement: Arduino IDE and Python IDE.
   - Purpose: Develop and deploy navigation and A.
4. **Libraries & Frameworks:**
   - OpenCV: For image processing.
   - PyTorch (YOLOv11): For real-time object detection.
   - Telegram Bot API: For sending real-time alerts.
   - NumPy and Pandas: Data manipulation and preprocessing.
   - Matplotlib: Visualization and metrics plotting.
5. **Network Connectivity:**
   - Requirement: Reliable internet connection.
   - Purpose: Enables real-time communication and cloud-based updates.

By combining robust hardware and software, the Night Patrol Robot ensures efficient and reliable performance for night-time surveillance tasks.

VI. CONCLUSION

The AI-Driven Night Patrol Robot offers a comprehensive solution to the challenges of night-time security and surveillance by integrating autonomous navigation with real-time AI-powered threat detection.

The system's ability to navigate dynamically using ultrasonic sensors and Arduino, coupled with its advanced detection capabilities through theYOLOv11 AI model and ESP32 CAM module, ensures reliable, proactive, and efficient monitoring.

This hybrid approach significantly reduces human dependency, enhances response times to potential threats, and provides a scalable framework for modern surveillance needs. By combining hardware and software innovations, the project successfully addresses gaps in traditional security systems, setting a new standard for automation in patrolling and monitoring tasks.

VII. ACKNOWLEDGMENT

This project received support from the East West Institute of Technology. We express our sincere appreciation guide, Mrs. Likhitha S, Assistant Professor Department of Information Science and Engineering for her valuable guidance and significant contributions. We also acknowledge Mrs. Shruthi T V, Prof & Head of ISE Department, for her support and mentorship throughout the project. Special thanks go to Principal Dr.Channakeshavalu for wavering encouragement. We are indebted to presence for their constructive feedback on earlier versions. Any shortcomings in the manuscript remain our and should not reflect negatively for the professional mentioned above.

REFERENCES

[1] Redmon,J.,&Farhadi,A.(2018).YOLOv3:An Incremental Improvement. Retrieved from https://arxiv.org/abs/1804.02767
[2] Espressif Systems. (2020). ESP32-CAM Datasheet. Retrieved from https://www.espressif.com/sites/default/files/documentation/esp32-cam_datasheet_en.pdf
[3] ArduinoDocumentation.(2023).ArduinoUnoTechnical Specifications. Retrieved from https://docs.arduino.cc/hardware/uno-rev3
[4] LabelStudioDocumentation.(2023).Open-SourceData Labeling Tool. Retrieved from https://labelstud.io/
[5] Ultralytics.(2023).YOLOv11CustomTrainingGuide. Retrieved from https://ultralytics.com/yolov11