

Design of an Autonomous Night Patrolling Bot with IoT-Based Remote Monitoring and Multi-Sensor Intrusion Detection

Disha Kute¹; Priya Longadge¹; Chetan Nimje¹; Harsh Gourikar¹; Malhar Borikar¹; Tanmay Mahankal¹; Uday Ghayade¹; Dr U. E. Hiwase²

¹B.Tech. Final Year Students, ²Assistant Professor

^{1,2} Department of Electrical Engineering, Priyadarshini College of Engineering, Nagpur, Maharashtra, India

Publication Date: 2026/09/30

Abstract: Securing residential colonies, campuses, warehouses and restricted industrial zones after dark still depends largely on human guards and fixed CCTV cameras, both of which suffer from fatigue, blind spots, poor low-light imaging and delayed reporting. This paper presents the design of a low-cost autonomous night patrolling bot that traverses a predefined route, detects obstacles, human movement and abnormal sound, and reports incidents to a remote user through an Internet of Things (IoT) platform. An Arduino Uno executes the patrol and sensing logic using an HC-SR04 ultrasonic sensor, an HC-SR501 passive infrared (PIR) sensor and a KY-038 sound sensor, while an L298N driver controls four DC gear motors. An ESP32-CAM module fitted with infrared illuminators provides night-vision snapshots and live video, and a Node MCU ESP8266 acts as the Wi-Fi gateway to a Blynk or Things Board dashboard. To reduce false alarms that affect single-sensor designs, a graded alert scheme fuses PIR and acoustic evidence and uses a stop-and-scan patrol cycle so that PIR readings are taken only when the platform is stationary. Analytical design checks indicate an obstacle-detection margin of more than 20 cm at a travel speed of about 0.34 m/s, an estimated operating time of roughly 2.4–3.3 h from a 3S Li-ion pack, and a total hardware cost of about ₹4,010. A structured validation plan with measurable performance indicators is also proposed.

Keywords: Autonomous surveillance, Blynk, ESP32-CAM, Internet of Things, Night Patrolling Robot, PIR Sensor, Sensor Fusion, Ultrasonic Obstacle Avoidance.

How to Cite: Disha Kute; Priya Longadge; Chetan Nimje; Harsh Gourikar; Malhar Borikar; Tanmay Mahankal; Uday Ghayade; Dr U. E. Hiwase (2026), Design of an Autonomous Night Patrolling Bot with IoT-Based Remote Monitoring and Multi-Sensor Intrusion Detection. *International Journal of Innovative Science and Research Technology* 11(9), 2175-2181. <https://doi.org/10.38124/ijisrt/26sep1044>

I. INTRODUCTION

Night-time is the period in which property crime, trespassing and vandalism are most difficult to prevent, because visibility is low and fewer people are present to notice unusual activity. Most Indian housing societies, educational campuses and small industrial units handle this risk with a combination of security guards on rotating shifts and a set of fixed CCTV cameras. Both measures have well-known weaknesses. A guard walking the same route for several hours gradually loses attention, and fixed cameras observe only the region inside their field of view, leaving corners, service lanes and open yards unobserved. Cameras without proper infrared illumination also produce dark, noisy images in which a person is hard to identify, and in many installations the footage is reviewed only after an incident has already taken place.

The Internet of Things (IoT) offers a practical way to close this gap. By attaching inexpensive sensors and wireless microcontrollers to a mobile platform, sensing can be carried to the places where it is needed and events can be pushed to a smartphone within seconds [1], [2]. A mobile robot does not replace human security staff; rather, it takes over the repetitive task of walking a fixed route and looking for anomalies, so that people are called in only when something requires a decision.

This paper describes the design of such a platform, referred to here as the Night Patrolling Bot. The main contributions are: (i) a complete, low-cost hardware architecture built from widely available modules; (ii) a stop-and-scan patrol strategy that makes PIR sensing usable on a moving robot; (iii) a graded, multi-sensor alert scheme intended to suppress false alarms; (iv) analytical checks of obstacle-detection timing, sensor coverage and battery endurance; and (v) a validation plan with quantitative

performance indicators. The rest of the paper is organized as follows. Section II reviews related work, Section III states the problem and objectives, Section IV presents the system architecture, Section V explains the methodology, Section VI gives the design analysis, Section VII outlines the validation plan, Sections VIII and IX discuss applications, limitations and future scope, and Section X concludes the paper.

II. LITERATURE SURVEY

➤ IoT Foundations for Remote Surveillance

Gubbi *et al.* [1] describe IoT as an architecture in which ubiquitous sensing, wireless communication and cloud-hosted analytics combine so that physical events become visible to remote users. Atzori *et al.* [2] survey the enabling technologies and point out that low-cost embedded nodes with standard internet protocols are the building blocks of such systems. Both surveys provide the conceptual basis for connecting a patrolling robot to a cloud dashboard. For the embedded side, Maier *et al.* [3] and Babiuch *et al.* [4] evaluate the ESP32 family and show that it combines adequate processing capability with integrated Wi-Fi at low cost, which makes it well suited to camera and telemetry nodes.

➤ Teleoperated Surveillance Robots

Early IoT surveillance robots were mostly operator-driven. Abdalla and Veeramanikandasamy [5] built a spy robot

on a Raspberry Pi that streams video and is steered by a remote user over an IP network. Anandravisekar *et al.* [6] presented an IoT-based surveillance robot that sends camera data to a remote user, and Akilan *et al.* [7] proposed an IoT robot for surveillance in hazardous locations where human entry is unsafe. These systems prove that live remote viewing is feasible with inexpensive hardware, but they depend on a person to drive the robot, so they do not remove the need for continuous human attention during a night shift.

➤ Night Patrolling Robots

Several works target night patrolling explicitly. Manasa *et al.* [8] and Kitchagiri *et al.* [9] proposed night-vision patrolling robots based on NodeMCU and Raspberry Pi respectively. Jullie Josephine *et al.* [10] used proximity sensing to detect and track human motion during night patrol. A group of studies motivated by women's safety on streets [11], [12], [13] combine motion and sound sensing with IoT reporting; for example, the robot in [13] follows a predetermined path, pauses when sound is heard and sends live updates to a control room. An Arduino- and NodeMCU-based design that uses ultrasonic obstacle avoidance, a sound sensor, a camera module and a buzzer with a web interface is reported in [14], and a recent chapter by Puvvada and Bodduluri [15] reviews autonomous night-patrol systems for comprehensive security. Table I summarises representative approaches.

Table 1 Summary of Representative Related Work

Ref.	Approach	Observed limitation
[5]	Raspberry Pi spy robot with IP-based video and remote steering	Needs a full-time operator; no autonomous patrol
[7]	IoT surveillance robot for hazardous locations	Operator-driven; not tailored to night-time imaging
[9]	Raspberry Pi night-vision patrolling robot	Higher processor cost and power draw
[10]	Night patrol robot with proximity-based human tracking	Short-range sensing; limited remote dashboard
[11]	IoT and AI-based night patrolling robot	AI processing raises cost and energy demand
[13]	Path-following robot; pauses on sound, live updates	Sound cue alone is prone to nuisance triggers
[14]	Arduino + NodeMCU robot with ultrasonic, sound sensor and camera	No PIR confirmation; alerting relies on single cue

➤ Research Gap

The review shows three recurring gaps. First, many platforms are either manually driven or depend on relatively expensive single-board computers. Second, event detection is often based on a single cue, typically sound or motion alone, which leads to nuisance alarms from animals, vehicles or wind. Third, the practical difficulty of using PIR sensors on a moving base, where the shifting thermal background itself can trigger the detector, is rarely addressed. The present design responds to these gaps with an Arduino-class controller, IR-assisted camera confirmation, fusion of PIR and acoustic evidence, and a stop-and-scan cycle for PIR sampling.

III. PROBLEM STATEMENT AND OBJECTIVES

The problem addressed is to provide continuous, low-cost night-time monitoring of a known area in which (a) human patrols are irregular because of fatigue, (b) fixed

cameras leave blind spots, (c) ordinary cameras perform poorly in darkness, and (d) owners learn of intrusions late because no real-time alert exists. The objectives of the work are to:

- Build a robot that patrols a predefined route without a human driver and avoids obstacles automatically;
- Detect obstacles, human movement and unusual sound using ultrasonic, PIR and acoustic sensors;
- Capture night-vision images and live video using an IR-illuminated camera module;
- Deliver real-time alerts, snapshots and status to a mobile application and web dashboard;
- Raise an immediate local alarm through a buzzer and indicator LED; and
- Keep the hardware cost low enough for adoption by housing societies and small institutions.

IV. PROPOSED SYSTEM ARCHITECTURE

➤ Overview

The architecture follows a sense–decide–act structure with an IoT layer for remote supervision, as shown in Fig. 1. The Arduino Uno is the real-time controller: it reads the three sensors, drives the motors and decides when an event must be reported. Time-critical tasks such as obstacle avoidance are handled locally, so the robot remains safe even if the Wi-Fi link is lost. Networking is delegated to two Wi-Fi modules. The NodeMCU ESP8266 receives compact event messages from the Uno over a serial link and forwards telemetry and notifications to the cloud, while the ESP32-CAM handles the bandwidth-intensive video stream independently. Separating control, telemetry and video in this way keeps the firmware of each processor simple and prevents video traffic from delaying safety-related control actions.

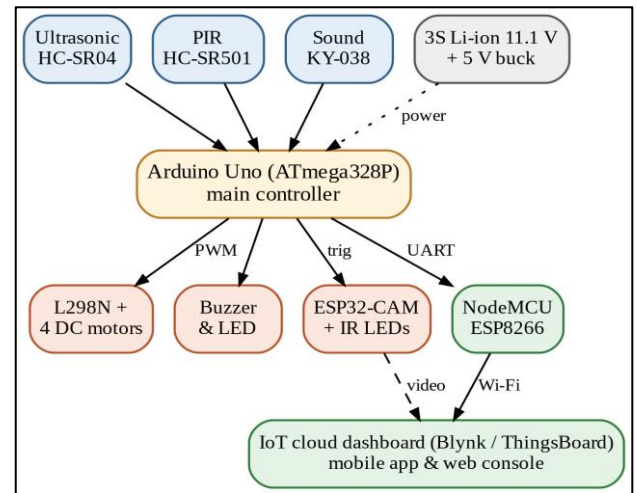


Fig 1 Hardware Block Diagram of the Proposed Night Patrolling Bot.

➤ Hardware Components

The selected components and their roles are listed in Table II. All parts are commonly available in the Indian hobby-electronics market, which keeps the design reproducible and easy to repair [16], [17], [18], [19].

Table 2 Hardware Components and Their Functions

Component	Key specification	Function
Arduino Uno R3	ATmega328P, 16 MHz, 14 digital I/O	Main controller: sensing, motion and decision logic
NodeMCU ESP8266	802.11 b/g/n Wi-Fi, 3.3 V logic	IoT gateway for telemetry and alerts
ESP32-CAM (OV2640)	2 MP sensor, Wi-Fi, external IR LEDs	Night-vision snapshots and live video
HC-SR04	2–400 cm range, 40 kHz	Obstacle detection and avoidance
HC-SR501 PIR	Up to about 7 m, adjustable delay	Human-body movement detection
KY-038	Microphone, analog and digital outputs	Unusual-sound detection
L298N	Dual H-bridge, up to 2 A per channel	Speed and direction control
Chassis + 4 BO motors	4-wheel drive, 6–12 V motors	Locomotion
Li-ion pack	3S 18650 (11.1 V) + buck converter	Rechargeable power supply
Buzzer and LEDs	5 V active buzzer	Local audible and visual alarm

➤ Interfacing

Table III gives the proposed signal mapping on the Uno. The ENA and ENB inputs of the L298N are driven from PWM-capable pins so that speed can be adjusted in firmware. The hardware serial pins D0/D1 are kept free for programming and debugging, and a software serial port on A1/A2 is used for the link to the NodeMCU. Because the NodeMCU operates at 3.3 V, the Uno transmit line is reduced through a 1 k Ω /2 k Ω resistive divider; the same arrangement is used for the capture-trigger line to the ESP32-CAM.

Table 4 Proposed Arduino Uno Pin Mapping

Uno pin(s)	Connected signal
D2	PIR output (external interrupt)
D3 / A0	Sound sensor digital / analog output
D4, D13	Buzzer, status LED
D5, D6 (PWM)	L298N ENA, ENB (motor speed)
D7–D10	L298N IN1–IN4 (motor direction)
D11, D12	HC-SR04 TRIG, ECHO
A1, A2	Software-serial RX/TX to NodeMCU (via divider)
A3	Snapshot trigger to ESP32-CAM (via divider)

➤ Power Subsystem

A three-cell series (3S) 18650 Li-ion pack with a nominal voltage of 11.1 V supplies the motor driver directly. A buck converter steps the pack voltage down to a regulated 5 V rail for the Uno, sensors, camera module and NodeMCU. Feeding the logic from a switching regulator rather than from the Uno's linear regulator reduces heat and prevents motor-induced voltage dips from resetting the controllers. A battery-management module protects the cells from over-charge, over-discharge and short circuit.

V. METHODOLOGY

The development follows the staged process of Fig. 2. Hardware and firmware are developed and tested module by module before IoT integration, after which the complete robot is evaluated in night scenarios and its parameters are tuned iteratively.

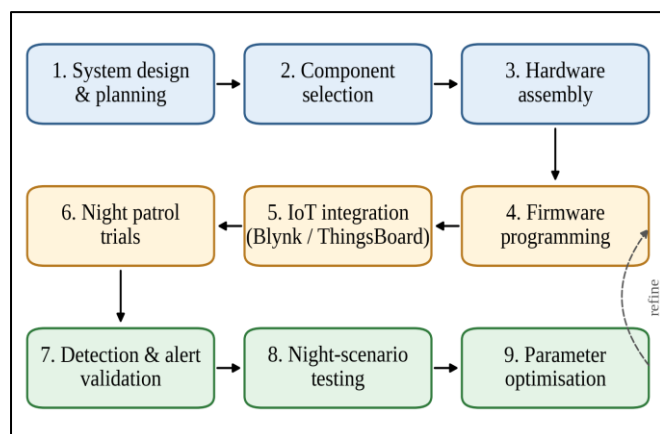


Fig 2 Development Methodology of the Proposed System.

➤ Patrol Route Strategy

The patrol route is stored in firmware as an ordered table of segments, each defined by a motion primitive (forward, turn left, turn right) and its duration or target count. The Uno executes the segments cyclically. At the end of each straight segment the robot halts for a short scan window, typically 3–5 s, during which the PIR and sound channels are sampled and a snapshot may be requested. This stop-and-scan cycle is central to the design: a PIR sensor responds to changes in received infrared energy, and when the sensor itself moves, the

changing thermal background can produce spurious detections. Sampling the PIR only while the platform is stationary removes this source of false alarms. Because the route is executed open loop, small heading errors accumulate; wheel encoders or line-following markers are therefore identified as future refinements.

➤ Obstacle Detection and Avoidance

The HC-SR04 emits a 40 kHz burst and reports the round-trip time t of the echo. The distance d to the obstacle is

$$d = v_s \cdot t / 2 \quad (1)$$

Where v_s is the speed of sound, which depends on air temperature T (°C) approximately as

$$v_s \approx 331.3 + 0.606 T \text{ m/s} \quad (2)$$

Ultrasonic ranging is a long-established technique for obstacle avoidance on mobile robots [20], [21]. In the firmware a measurement is taken every 60 ms and the median of three readings is used to reject isolated spurious echoes. When d falls below a threshold $d_{th} = 25$ cm, the robot stops, reverses briefly, turns towards the clearer side and then re-joins the current route segment.

➤ Human Motion and Sound Detection

The HC-SR501 detects the movement of warm bodies through a pyroelectric element behind a Fresnel lens. Its sensitivity and hold time are set with on-board potentiometers, and retrigger mode is selected so that continued movement keeps the output active. The KY-038 provides a comparator output whose threshold is set with a trimmer, together with an analog output. To prevent a single impulsive sound, such as a slammed door, from generating an event, the sound flag is raised only when the analog level stays above the threshold for at least 200 ms.

➤ Multi-Sensor Alert Logic

Instead of treating every trigger as an intrusion, the controller assigns one of three alert levels using the combination of evidence observed within a 5 s window, as summarised in Table IV. The camera snapshot provides visual confirmation to the remote user, and a 60 s cool-down between push notifications prevents the dashboard from being flooded during a continuing event.

Table 4 Graded Alert Decision Logic

PIR / Sound	Alert level	System response
0 / 0	Normal	Continue patrol; periodic status update
0 / 1	Level 1 – Advisory	Log event, capture snapshot, update dashboard; no push notification
1 / 0	Level 2 – Suspected	Halt, snapshot, push notification; re-sample PIR for 3 s
1 / 1, or PIR persists	Level 3 – Intrusion	Buzzer and LED on, push alert with snapshot, live stream enabled

➤ IoT Communication and Dashboard

The Uno sends short, comma-separated event frames to the NodeMCU at 9600 baud, for example an identifier, the alert level, the PIR and sound flags and the last obstacle distance. The NodeMCU maps these values to Blynk

datastreams (virtual pins), which drive widgets such as a status indicator, an event counter and a distance gauge, and it raises Blynk events that the platform delivers to the user's phone as push notifications [22]. Things Board can be used as an open-source alternative, with telemetry posted over MQTT or HTTP

and alarm rules defined on the server [23]. The ESP32-CAM runs a web server that serves an MJPEG stream and single-frame captures. For night imaging, 850 nm IR LEDs illuminate the scene; a lens without an IR-cut filter is required, since the standard OV2640 lens used on many modules blocks much of this wavelength.

➤ Operational Flow

The complete control flow is given in Fig. 3. After initialization, the robot repeatedly executes route segments, checks the obstacle distance, evaluates the detection channels during scan windows, and raises the appropriate alert before continuing. The loop ends when the route is complete or when the user stops the patrol from the dashboard.

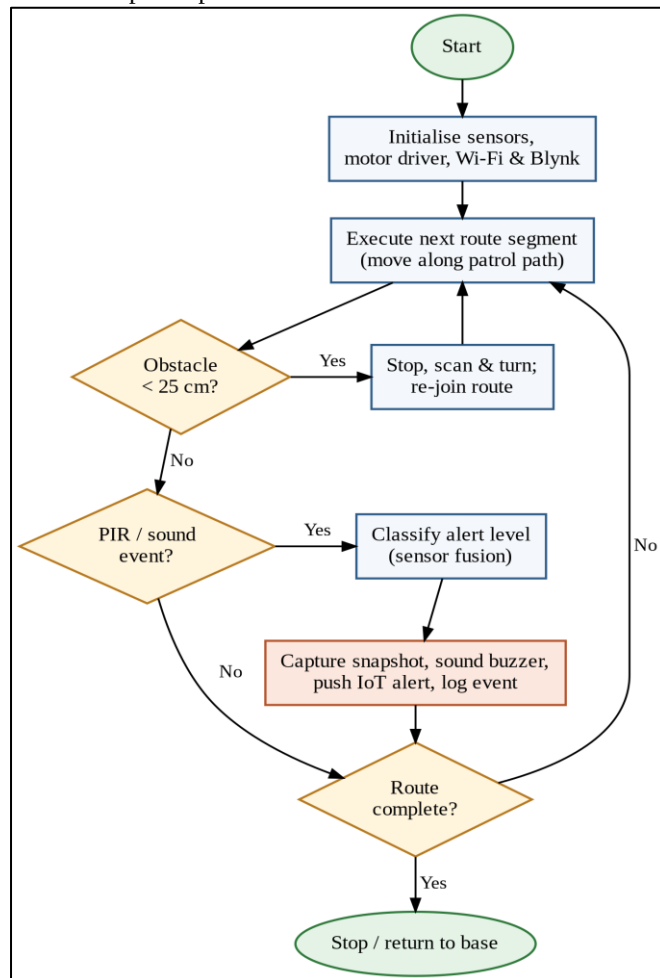


Fig 3 Control Flow of the Patrolling Algorithm.

VI. DESIGN ANALYSIS

➤ Travel Speed and Obstacle-Detection Margin

With 65 mm wheels driven at about 100 rpm, the forward speed is

$$v_r = \pi D N / 60 \approx 0.34 \text{ m/s} \quad (3)$$

For an obstacle at the 25 cm threshold, the echo returns in about 1.46 ms, and even an echo from the sensor's 4 m limit returns within about 23 ms, well inside the 60 ms measurement interval. Between two successive measurements the robot moves roughly 2 cm, so an obstacle that enters the beam just after one reading is still detected more than 20 cm away. This margin comfortably covers the braking distance of a small geared platform. Using a fixed speed of sound of 343 m/s at a night temperature of 10 °C introduces an error of only about 1.7%, or about 0.4 cm at the threshold, so temperature compensation through (2) is optional.

➤ PIR Coverage

Taking a detection range of 7 m and a horizontal cone of about 110°, the area covered from a single scan position is approximately $(110/360) \cdot \pi \cdot (7 \text{ m})^2 \approx 47 \text{ m}^2$. Scan points are therefore spaced so that successive coverage sectors overlap along corridors and yard boundaries.

➤ Power Budget and Endurance

Table V lists the estimated load based on typical datasheet values. With a buck-converter efficiency of about 85%, the 5 V electronics draw roughly 3.0 W from the pack, while the four motors draw about 0.6 A at the pack voltage, or about 6.7 W, when moving.

Table 5 Estimated Power Budget (Typical Values)

Load	Current	Power
Arduino Uno	50 mA @ 5 V	0.25 W
ESP32-CAM (streaming)	240 mA @ 5 V	1.20 W
IR LED array (4 × 20 mA)	80 mA @ 5 V	0.40 W
NodeMCU ESP8266	80 mA @ 5 V	0.40 W
Sensors, L298N logic, buzzer	60 mA @ 5 V	0.30 W
5 V rail from pack ($\eta \approx 85\%$)	—	$\approx 3.0 \text{ W}$
4 BO motors (moving)	0.6 A @ 11.1 V	$\approx 6.7 \text{ W}$
Total while moving	—	$\approx 9.7 \text{ W}$

A 3S pack of 2600 mAh cells stores about 28.9 Wh. Allowing 80% usable capacity gives about 23 Wh, which corresponds to roughly 2.4 h of continuous motion. With the stop-and-scan cycle, if the robot is moving 60% of the time, the average demand falls to about 7 W and the estimated endurance rises to about 3.3 h. Longer coverage can be obtained with a 3S2P pack or with the auto-docking scheme proposed in Section

IX. These figures are design estimates and will be verified by measurement on the prototype.

➤ Cost Analysis

The approximate component cost is summarized in Table VI. The complete platform costs about ₹4,010, which is comparable to one or two good-quality fixed cameras, yet the robot covers a much larger area because it move.

Table 6 Estimated Component Cost

Component	Qty	Cost (₹)
Arduino Uno R3	1	550
NodeMCU ESP8266	1	300
ESP32-CAM + IR LEDs	1	700
HC-SR04 ultrasonic sensor	1	80
HC-SR501 PIR sensor	1	90
KY-038 sound sensor	1	60
L298N motor driver	1	180
Chassis, 4 BO motors, wheels	1 set	700
18650 cells, holder, charger, buck	1 set	950
Buzzer, LEDs, wires, PCB, misc.	—	400
Total (approximate)		4,010

VII. EXPERIMENTAL VALIDATION PLAN

The prototype will be evaluated in an indoor corridor and an outdoor campus yard under ambient illumination below about 1 lux. Table VII lists the test scenarios and the performance indicators that will be reported. Each scenario will be repeated a sufficient number of times, for example 30 trials, so that detection and false-alarm rates can be stated with reasonable confidence.

Table 7 Test Scenarios and Performance Indicators

Scenario	Indicator
Person walking across path at 2, 4 and 6 m	Detection rate (%) at each range
Empty route; animals, traffic and wind noise	False alarms per hour
Event to phone notification	Alert latency (s); target below 5 s
Hard, soft and angled obstacles	Avoidance success rate (%), collisions
Snapshot in darkness at 2–5 m	Recognisability of a person in the image
Full charge to cut-off	Measured endurance (h)
Wi-Fi link interrupted	Continued safe patrol; event recovery on reconnection

Soft materials such as curtains and foam absorb ultrasound and may be detected at shorter range, and the scenarios deliberately include such cases so that the limitation can be quantified rather than assumed.

modular design to which new sensors can be added with little change to the firmware structure.

VIII. APPLICATIONS AND ADVANTAGES

The platform is suited to residential societies and gated campuses, warehouses, godowns and factories, college and hospital premises, power substations and other restricted zones, farms, and parking areas or ATM surroundings. Its main advantages are fatigue-free, repeatable patrolling; lower cost than a dense multi-camera installation; real-time alerts to a phone from any location; elimination of fixed blind spots through mobility; reduced dependence on manpower; and a

IX. LIMITATIONS AND FUTURE SCOPE

The current design relies on Wi-Fi coverage along the route, uses open-loop route execution that can drift over long patrols, and performs no on-board recognition of people. Future work will address these points through: person or face recognition on the camera feed to further reduce false alarms; wheel encoders, GPS for outdoor sites, or LiDAR-based SLAM for larger areas; automatic docking with solar-assisted charging; GSM or LoRa fallback links where Wi-Fi is unavailable; coordination of several robots from a single dashboard; and fire, smoke and gas sensors so that the same patrol can also serve as a safety inspection.

X. CONCLUSION

This paper has presented the design of an autonomous, IoT-connected night patrolling bot built from low-cost, readily available modules. The architecture separates real-time control on an Arduino Uno from IoT telemetry on a NodeMCU and video on an ESP32-CAM, so that obstacle avoidance remains reliable regardless of network conditions. A stop-and-scan patrol cycle makes PIR sensing practical on a mobile base, and a graded fusion of PIR and acoustic evidence with camera confirmation is proposed to reduce false alarms. Analytical checks indicate an adequate obstacle-detection margin, an endurance of roughly 2.4–3.3 h and a hardware cost of about ₹4,010. The prototype will be built and evaluated against the indicators defined in Section VII, and the results are expected to show that dependable night-time monitoring can be made affordable for small institutions and residential communities.

ACKNOWLEDGMENT

The authors thank the Department of Electrical Engineering, Priyadarshini College of Engineering, Nagpur, for providing laboratory facilities and support for this work.

REFERENCES

- [1]. J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, 2013.
- [2]. L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.
- [3]. A. Maier, A. Sharp, and Y. Vagapov, "Comparative analysis and practical implementation of the ESP32 microcontroller module for the Internet of Things," in *Proc. Internet Technologies and Applications (ITA)*, 2017, pp. 143–148.
- [4]. M. Babiuch, P. Foltýnek, and P. Smutný, "Using the ESP32 microcontroller for data processing," in *Proc. 20th Int. Carpathian Control Conf. (ICCC)*, 2019, pp. 1–6.
- [5]. G. O. E. Abdalla and T. Veeramanikandasamy, "Implementation of spy robot for a surveillance system using internet protocol of Raspberry Pi," in *Proc. 2nd IEEE Int. Conf. Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, 2017, pp. 86–89, doi: 10.1109/RTEICT.2017.8256563.
- [6]. G. Anandrasekar, A. A. Clinton, T. M. Raj, L. Naveen, and M. Mahendran, "IoT based surveillance robot," *International Journal of Engineering Research & Technology (IJERT)*, vol. 7, no. 3, pp. 84–87, 2018.
- [7]. T. Akilan, S. Chaudhary, P. Kumari, and U. Pandey, "Surveillance robot in hazardous place using IoT technology," in *Proc. 2nd Int. Conf. Advances in Computing, Communication Control and Networking (ICACCCN)*, 2020, pp. 775–780, doi: 10.1109/ICACCCN51052.2020.9362813.
- [8]. P. Manasa, K. Sri Harsha, D. DM, R. Karthik, and O. Naveen Nichal, "Night vision patrolling robot using NodeMCU," *Journal of Xi'an University of Architecture & Technology*, vol. 12, no. 5, 2020.
- [9]. P. Kitchagiri, S. Vaddi, S. R. Rajanala, B. V. Manthena, S. Sanepalli, and I. A. Pindoo, "Night vision patrolling robot for security patrolling using Raspberry Pi," *International Journal of Engineering Research and Applications*, vol. 11, no. 5, pp. 1–7, 2021.
- [10]. D. C. Jullie Josephine, G. N. P. S. Frank, A. Manikandan, S. Bhavaneskumar, and T. Sakthivel, "Night patrol robot for detecting and tracking of human motions using proximity sensor," in *Proc. Int. Conf. Edge Computing and Applications (ICECAA)*, 2022.
- [11]. S. Bhatia, S. Verma, N. Singh, I. Saxena, and I. Verma, "IoT and AI based women's safety night patrolling robot," in *Proc. 2nd Int. Conf. Advancement in Electronics & Communication Engineering (AECE)*, 2022, doi: 10.2139/ssrn.4158918.
- [12]. A. R. Komatireddy, K. R. Reddy, B. Boini, and G. G. Kumar, "Women safety night patrolling robot," *International Journal for Research in Applied Science & Engineering Technology*, vol. 10, no. 6, pp. 1279–1284, 2022, doi: 10.22214/ijraset.2022.44043.
- [13]. R. Mahalakshmi, M. Kavitha, B. Gopi, and S. M. Kumar, "Women safety night patrolling IoT robot," in *Proc. 5th Int. Conf. Smart Systems and Inventive Technology (ICSSIT)*, 2023, pp. 544–549, doi: 10.1109/ICSSIT55814.2023.10060955.
- [14]. "IoT based smart night patrolling robot," *International Journal of Creative Research Thoughts (IJCRT)*, vol. 11, no. 4, 2023, Paper ID IJCRT2304133.
- [15]. K. S. Puvvada and A. J. Bodduluri, "Robotic vigilance: Autonomous night patrol systems for comprehensive security," Springer Nature, 2026, doi: 10.1007/978-981-95-0684-2_15.
- [16]. Arduino, "Arduino Uno Rev3 – Technical documentation," docs.arduino.cc.
- [17]. Espressif Systems, "ESP8266EX datasheet" and "ESP32 series datasheet."
- [18]. STMicroelectronics, "L298 dual full-bridge driver datasheet."
- [19]. ElecFreaks, "HC-SR04 ultrasonic ranging module user guide."
- [20]. R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to Autonomous Mobile Robots*, 2nd ed. Cambridge, MA, USA: MIT Press, 2011.
- [21]. J. Borenstein and Y. Koren, "Obstacle avoidance with ultrasonic sensors," *IEEE Journal on Robotics and Automation*, vol. 4, no. 2, pp. 213–218, 1988.
- [22]. Blynk Inc., "Blynk IoT platform documentation," docs.blynk.io.
- [23]. ThingsBoard, "ThingsBoard documentation," thingsboard.io/docs.