ORIGINAL RESEARCH PAPER

Volume - 15 | Issue - 04 | April - 2026 | PRINT ISSN No. 2277 - 8179 | DOI : 10.36106/ijsr

INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH

A REAL-TIME IOT SYSTEM FOR CROWD DENSITY MONITORING AND SMS-BASED EMERGENCY ALERTS

Computer Science

Muthunagai. R	Assistant Professor, Jansons Institute of Technology
Akina S	UG Student, Jansons Institute of Technology
Gollavilli Saraswathi	UG Student, Jansons Institute of Technology
Priyadharshini D	UG Student, Jansons Institute of Technology

ABSTRACT

Rapid urbanization and increasing frequency of large public gatherings have made automated crowd management a critical public safety requirement. This paper presents an IoT-based real-time crowd flow monitoring and density analysis system that automatically detects and responds to overcrowding conditions in public spaces such as railway stations, shopping malls, stadiums, and event venues. A Python application captures continuous live video through a webcam and processes each frame using the YOLOv8n object detection model to identify and count individuals across four spatial zones (A, B, C and D). A temporal smoothing algorithm stabilizes zone-wise counts, and occupancy is classified as Low (below 18%), Medium (18–30%), or High (above 30%). When two consecutive frames confirm a High-density zone, an alert command is transmitted over UART serial communication at 115,200 baud to an ESP32 microcontroller, which simultaneously activates three response mechanisms: the SIM800L GSM module dispatches an SMS notification to designated authorities; a 16×2 LCD display shows the real-time person count and zone status; and an optional 5 V buzzer produces a local audible warning. When crowd density returns below the threshold, normal monitoring resumes automatically. System testing confirmed person detection accuracy of approximately 94%, zone classification accuracy of approximately 91%, and alert delivery latency under 1.5 seconds from threshold breach. The complete hardware costs between ₹2,500 and ₹4,000, making the system cost-effective and scalable for smart city deployment.

KEYWORDS

IoT, Crowd Monitoring, ESP32, YOLOv8n, GSM Alert, Real-time Density Analysis, Smart City

INTRODUCTION

The rapid growth of urban populations has made crowd management a critical public safety challenge. Overcrowding in places like railway stations, shopping malls, and stadiums can trigger stampedes, accidents, and delayed emergency responses. Conventional surveillance systems depend entirely on human operators watching CCTV footage, which is labor-intensive, error-prone, and incapable of generating automatic alerts [1].

Advances in computer vision and IoT embedded systems now enable low-cost, automated, and intelligent crowd monitoring. This work presents a system that integrates Python-based YOLOv8n human detection with an ESP32 microcontroller, GSM alerting, and LCD feedback to deliver real-time, zone-wise crowd density analysis without manual intervention.

Literature Survey

Kannan et al. [1] proposed a heatmap-based crowd monitoring system using computer vision, which was visually effective but computationally expensive. Kumar and Singh [2] employed ultrasonic IoT sensors for density estimation, achieving cost efficiency at the expense of accuracy in dense scenarios.

Gazis and Katsiri [3] introduced IoT cloud middleware for scalable remote monitoring, though performance was constrained by network dependency. Chen et al. [4] demonstrated YOLOv8-based crowd detection at 24 FPS on CPU, validating its suitability for embedded deployment.

Sharma and Kanwal [8] reviewed IoT-based smart surveillance and emphasised the necessity of automated real-time alerting for proactive crowd management. Unlike these works, the proposed system uniquely integrates edge-based YOLOv8n detection, zone-wise density smoothing, GSM alerting, and LCD feedback within a single low-cost platform.

System Design

3.1 Architecture

The system is organised into three layers. The Sensing Layer consists of a webcam that captures continuous live video at the monitoring site. The Processing Layer runs a Python application using OpenCV and YOLOv8n to detect and count individuals per zone. The Response Layer comprises the ESP32 microcontroller, SIM800L GSM module, 16×2 LCD display, and buzzer.

3.2 Zone-Wise Density Analysis

Each video frame is divided into four equal quadrants (Zones A, B, C, D). YOLOv8n (class ID 0: person, confidence ≥ 0.5) detects bounding boxes. For each detected person, centroid coordinates determine zone assignment. A temporal smoothing function stabilizes raw counts:

$$\text{count\_smooth} = 0.7 \times \text{prev} + 0.3 \times \text{current}.$$

Zone density state is classified as: LOW (<18% occupancy ratio), MEDIUM (18–30%), or HIGH (>30%). Two consecutive HIGH frames must be observed before an alert is confirmed, preventing false triggers.

3.3 Alert Mechanism

When any zone reaches confirmed HIGH state, the Python application transmits a message (e.g., "HIGH: A,C") over UART at 115,200 baud to the ESP32. The ESP32 simultaneously: (i) instructs the SIM800L GSM module to send an SMS to registered authority contacts; (ii) updates the LCD with "Crowd Alert" and the live person count; and (iii) optionally activates the buzzer. When density falls below threshold, "NORMAL" is transmitted and all alerts are cleared.

Hardware Components

The complete hardware stack is summarised in Table 1. The ESP32-S (240 MHz dual-core, onboard Wi-Fi/Bluetooth, 512 KB internal + 4 MB PSRAM) acts as the central controller. The SIM800L GSM module operates on the 900 MHz band and communicates with the ESP32 via TTL UART. The 16×2 LCD displays zone status and crowd count locally. An active 5 V buzzer (85 dB, 3 kHz) provides audible alerts. Power is supplied through a 7812/7805 dual-regulator circuit providing stable 12 V and 5 V DC from a 230 V AC mains source.

Table: 1 Specialization of Components

Component	Specialization
ESP32-S	240 MHz, 4 MB PSRAM, Wi-Fi+BT
Camera	Webcam / ESP32-CAM OV2640
GSM Module	SIM800L, 900 MHz, TTL UART
LCD Display	16×2, I2C interface
Buzzer	Active 5V, 85 dB, 3 kHz
Power Supply	7812/7805, 5V & 12V DC

Software Implementation

The software stack uses Python 3.x with OpenCV for frame capture and preprocessing. YOLOv8n (Ultralytics) handles person detection. Serial communication uses PySerial at 115,200 baud. The ESP32 is programmed in Embedded C via Arduino IDE, interfacing GSM via AT commands and driving the LCD through the LiquidCrystal library. Two parallel paths exist: webcam mode (YOLO zone counting) and video file mode (pixel-density analysis via Gaussian blur thresholding). Rate limiting (0.5 s minimum interval) prevents serial bus flooding.

RESULTS AND DISCUSSION

The system was tested under indoor conditions with varying crowd sizes (1–10 persons) across four zones and under different lighting conditions. Table 2 summarizes key performance metrics.

Table : 2 Key Performance Metrics

Metric	Value
Person detection accuracy	~94% (YOLOv8n)
Zone classification accuracy	~91% across zones
Alert generation latency	<1.5 s from breach
GSM SMS delivery time	~3–5 s
LCD refresh interval	1 s
Estimated hardware cost	₹2,500–₹4,000

Hardware implementation confirmed that the LCD displayed correct zone-wise status (“SMS Sent! Zone: A”) and the GSM module delivered alerts reliably under stable network conditions. Detection accuracy decreased slightly under extreme low-light conditions, consistent with YOLOv8n's known sensitivity to frame quality. The proposed system's hardware cost is approximately 5× lower than comparable commercial crowd management devices, confirming viability for public-sector deployment.

CONCLUSIONS

This paper presented an IoT-based real-time crowd flow monitoring and density analysis system that integrates YOLOv8n computer vision with ESP32 embedded control, GSM alerting, and LCD feedback. The system demonstrated reliable zone-wise density classification, low-latency alert generation under 1.5 seconds, and cost-effective hardware deployment at under ₹4,000, making it suitable for smart city environments, transportation hubs, and large-scale public events.

Future work will extend the system to multi-camera coverage, cloud-based analytics dashboards, and predictive crowd density forecasting using deep learning trend models.

REFERENCES

[1] P. Kannan, S. Kumar, and P. Devi, "Real-Time Crowd Monitoring and Management Using Computer Vision and Heatmap Visualization," Int. J. Computer Vision Applications, vol. 12, no. 3, pp. 45–58, 2025.
[2] A. Kumar and R. Singh, "Real-Time Crowd Density Estimation Using Ultrasonic Sensors and IoT," Int. J. IoT Systems, vol. 8, no. 2, pp. 101–110, 2021.
[3] V. Gazis and E. Katsiri, "Streamlined Intelligent Crowd Monitoring with IoT Cloud Middleware," IEEE Internet of Things J., vol. 11, no. 4, pp. 2301–2312, 2024.
[4] L. Chen et al., "Dense-Stream YOLOv8n for Real-Time Crowd Monitoring in Smart Environments," IEEE Access, vol. 13, pp. 56789–56800, 2025.
[5] O. Janssens et al., "Device-Free Crowd Size Estimation Using Wireless Sensing," Sensors, vol. 24, no. 6, pp. 1500–1512, 2024.
[6] K. Ramesh and P. Vardhan, "IoT-Enabled Smart Surveillance for Public Places," Int. J. Smart Systems, vol. 10, no. 1, pp. 75–85, 2022.
[7] M. Qaraqe et al., "Crowd Behavior Detection Using Video Swin Transformer," Applied Intelligence, vol. 54, no. 2, pp. 2205–2218, 2024.