International Journal of Trend in Scientific Research and Development (IJTSRD)
Special Issue on Smart Innovations in Computer Science and Applications
Available Online: www.ijtsrd.com e-ISSN: 2456 – 6470

Design and Deployment of an Intelligent Crowd Detection and Accident Prevention System

Siddhi Atkare
Department of Computer Application, School of Science,
G. H. Raisoni Skill Tech University, Nagpur, India

Annotation
Mass assemblies taken in open areas are usually safety hazards because of overcrowding, mismanagement, or delayed emergency response. Traditional surveillance tools are very manual-oriented and restrict the ability to capture severe conditions on time. This study proposes an Intelligent Crowd Detection and Accident Prevention System using OpenCV-controlled image processing integrated with YOLOv8 object detection and IoT communication. The system estimates crowd density through land visibility analysis, tracks movement velocity, and delivers threshold-based warnings to authorities. Zone-based monitoring detects localized overcrowding. The architecture offers an affordable, scalable, and real-time solution flexible to diverse settings like stadiums, transport stations, and community events, achieving 90–92% accuracy in crowd estimation with 2–3 second alert response time.

Keywords: Crowd Detection, Accident Prevention, OpenCV, YOLOv8, IoT, ThingSpeak, ESP32, Land Visibility, Zone Analysis.

I. INTRODUCTION

Worldwide, there is an increasing concern on the safety of the people during large gatherings. Concerts, religious gatherings, and sporting tournaments are events that have high population density, and unexpected occurrences may develop into stampede or crush injuries. Traditional surveillance depends on the human operator to look through the CCTV feeds, which is subject to time wastage and inaccuracies. In addition, the majority of existing systems are reactive and detect issues only once they arise.

As computer vision advances, tools like OpenCV and deep learning frameworks such as YOLOv8 can identify crowds effectively using regular hardware without specialized GPUs. Integrating these with IoT platforms enables real-time monitoring, alert generation, and remote visualization. This study is concerned with utilizing such tools to create a proactive, low-cost, and automated accident prevention system suitable for deployment in public spaces.

II. LITERATURE SURVEY AND RESEARCH GAP

The study of crowd detection has evolved significantly over the past decade, shifting from sensor-based and traditional image processing approaches toward AI-driven deep learning systems. Researchers have mainly concentrated on two approaches: computer vision-based detection and IoT-based monitoring. While vision systems provide visual intelligence for people counting and density estimation, IoT integration ensures data transmission, remote monitoring, and control actions.

Sensor-Based Approaches:

Early crowd monitoring systems relied on infrared sensors, pressure pads, or ultrasonic detectors to estimate human presence. While these techniques provided approximate crowd density, they lacked accuracy and scalability for large open spaces and are costly to install and maintain.

Image Processing-Based Approaches:

Traditional image processing techniques use background subtraction, motion tracking, or contour detection to estimate crowd density. Methods such as frame differencing and optical flow help analyze movement patterns. However, they are highly sensitive to environmental changes like illumination, shadows, and reflections.

Deep Learning-Based Approaches:

Modern crowd analysis has shifted toward Convolutional Neural Networks (CNNs) and YOLO-based frameworks. These models can identify and classify people with high accuracy even in dense environments. YOLOv8, used in this project, offers enhanced detection accuracy, faster training, and better performance on smaller datasets, processing 20–25 fps on standard laptops.

Research Gap:

There is little literature on pure OpenCV, software-only systems that combine density and velocity analysis with real-time threshold alerts and IoT integration on low-cost hardware. This paper addresses that gap by presenting an OpenCV + YOLOv8 + ThingSpeak framework deployable without GPU hardware.

III. SYSTEM ARCHITECTURE

The proposed system integrates both software and hardware components to perform detection, analysis, and communication. Fig. 1 shows the overall system block diagram, and Fig. 2 presents the detailed layered architecture of the system.

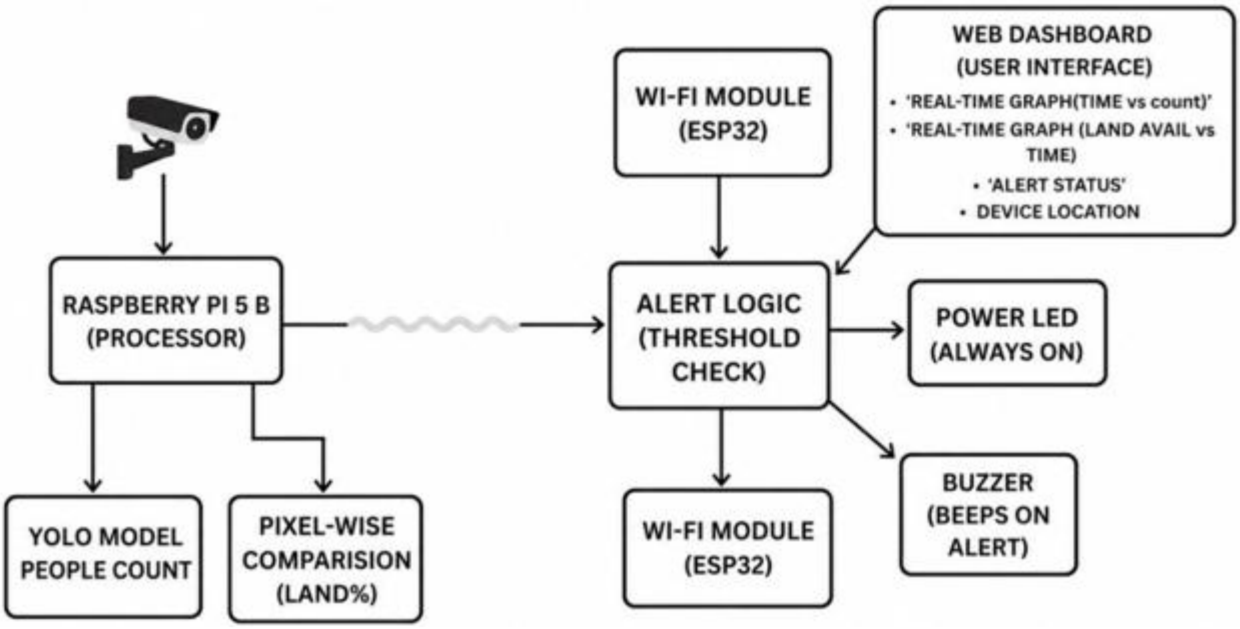

Fig. 1: System Block Diagram — Camera → Processing Unit → Alert Logic → ESP32 → ThingSpeak Dashboard

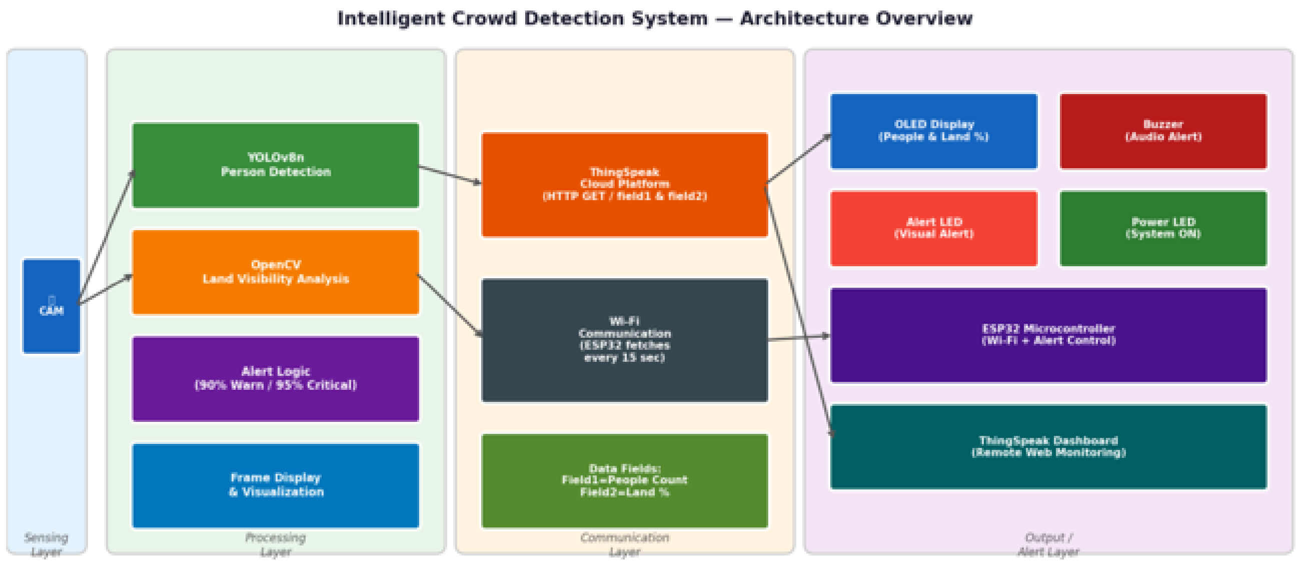

Fig. 2: Intelligent Crowd Detection System — Layered Architecture Overview

The system architecture is organized into four layers:

- Sensing Layer: A fixed overhead camera captures continuous live video frames from the monitored area.
- Processing Layer: A Raspberry Pi or laptop runs YOLOv8 for person detection and OpenCV for land visibility analysis. Alert logic compares occupancy against defined thresholds (Warning: 90%, Critical: 95%).
- Communication Layer: Processed data (people count and land availability percentage) is transmitted to the ThingSpeak cloud platform via HTTP GET requests every 16 seconds. ESP32 fetches this data via Wi-Fi every 15 seconds.
- Output / Alert Layer: The ESP32 drives an OLED display showing crowd data, an alert LED, and a buzzer. The ThingSpeak dashboard provides centralized remote monitoring.

IV. METHODOLOGY AND DESIGN

The design methodology illustrates the consecutive steps of detecting, analyzing, and preventing unsafe crowd conditions. The process is divided into two main sides: the Sender (camera and processing unit) and the Receiver (ESP32 and alert hardware).

4.1. Sender Side Process Flow

Fig. 3 shows the complete sender-side flowchart. The sender continuously captures video, runs YOLOv8 person detection, computes land visibility, checks thresholds, and transmits data to the cloud.

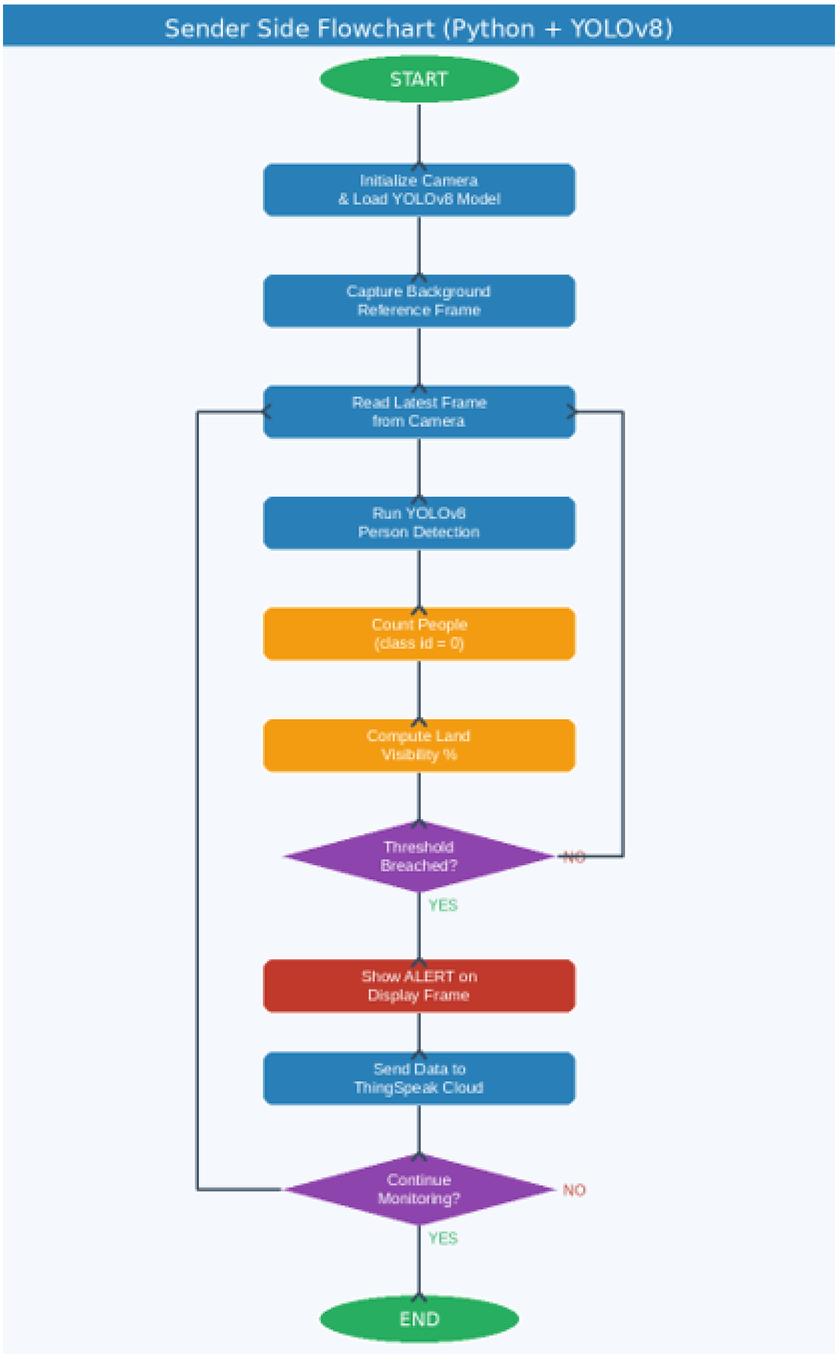

Fig. 3: Sender Side Flowchart — Python + YOLOv8 + OpenCV Processing Pipeline

4.2. Receiver Side Process Flow

Fig. 4 illustrates the ESP32 receiver flowchart. The ESP32 fetches cloud data, displays it on the OLED, and triggers alert mechanisms when unsafe thresholds are detected.

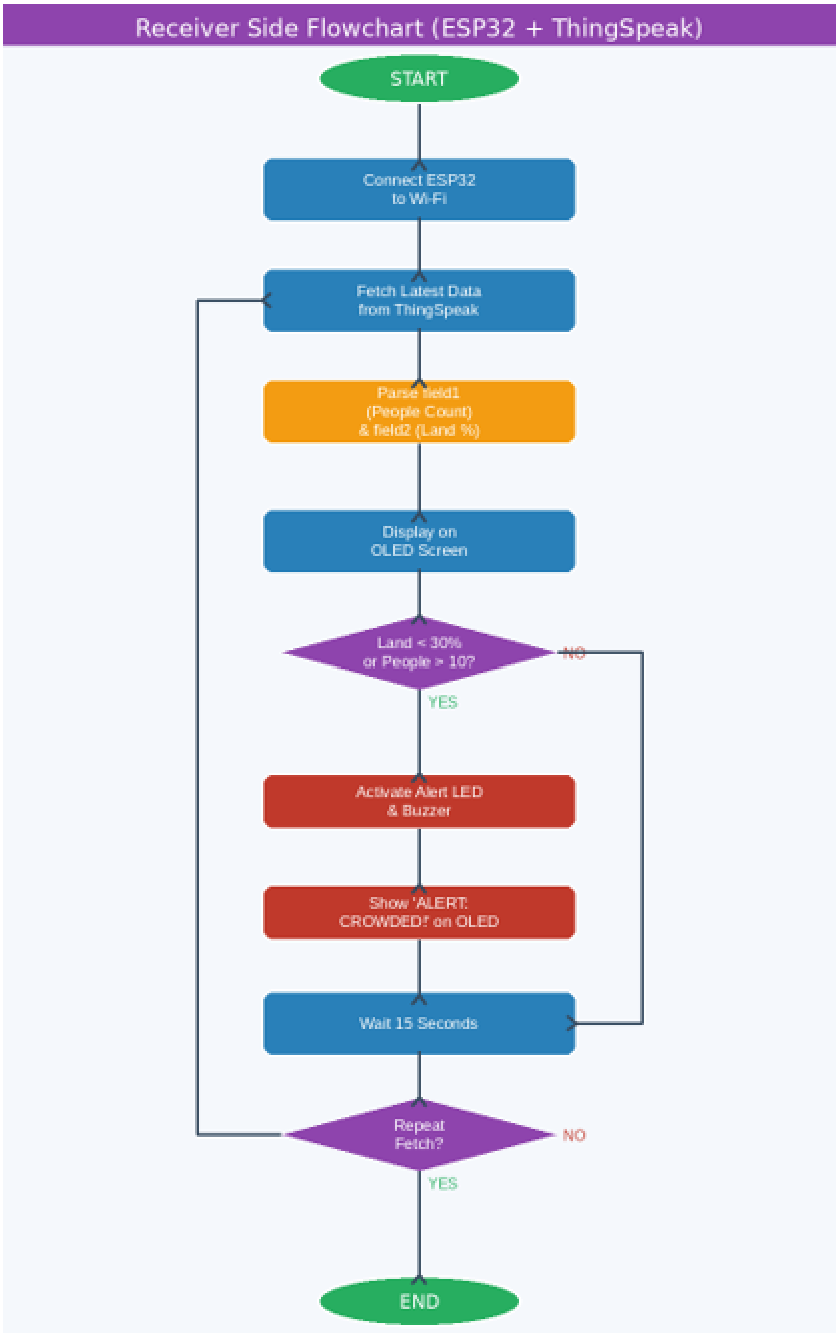

Fig. 4: Receiver Side Flowchart — ESP32 + ThingSpeak + Alert Mechanism

4.3. Key Processing Steps

The core processing pipeline on the sender side involves the following steps:

- Camera Initialization: cv2.VideoCapture(0) with buffer size reduced to 1 for minimal latency.
- Background Capture: A reference grayscale frame with Gaussian blur (21×21 kernel) is captured before detection begins.
- YOLOv8 Detection: model.predict() extracts person-class bounding boxes (class ID = 0) to compute people count.
- Land Visibility: Absolute difference between current frame and background is thresholded; non-zero pixels determine occupied percentage. Land available = 100 − occupied%.
- IoT Transmission: Payload with field1 (people count) and field2 (land %) is sent via HTTP GET

to ThingSpeak every 16 seconds.

- Alert Display: 'ALERT: CROWDED!' is overlaid on the frame when land < 30% or people count > 10.

V. HARDWARE COMPONENTS

The receiver unit consists of the following hardware components selected for low cost, low power consumption, and ease of deployment:

Component	Specification	Function
ESP32 Microcontroller	240 MHz, 4MB Flash, Wi-Fi 802.11 b/g/n	Main receiver controller; fetches cloud data, drives display and alerts
OLED Display (SSD1306)	0.96", 128×64 px, I2C	Shows people count, land %, and alert status in real time
Alert LED	GPIO 25, 220Ω resistor	Visual alert when crowd threshold exceeded
Power LED	GPIO 33, 220Ω resistor	Indicates system is powered and operational
Active Buzzer	GPIO 13, 3.3V–5V, ~85 dB	Audible alarm on critical crowd detection
Push Button	GPIO 34, 10KΩ pull-down	Page navigation on OLED display
Resistors	220Ω and 10KΩ	Current limiting for LEDs, pull-down for button
USB Webcam / IP Camera	HD 720p, 30 fps, 70° FOV	Captures live overhead video feed

Table 1: Hardware Components and Specifications

V. RESULTS AND DISCUSSION

The system was tested in three different environments: an open hall with medium lighting, an outdoor corridor with high brightness and shadows, and a controlled classroom setup. Performance was evaluated based on detection accuracy, land visibility estimation, response time, and IoT stability.

6.1. YOLOv8 Person Detection Accuracy

Test Scenario	Actual Count	Detected Count	Accuracy (%)
Indoor (5–8 people)	7	7	100%
Outdoor Corridor (10–12 people)	11	10	90.9%
Dense Standing Crowd	18	17	94.4%
Moving Crowd	15	14	93.3%

Table 2: YOLOv8 Person Detection Accuracy

YOLOv8n showed high accuracy (90–100%) across conditions. Minor accuracy drops occurred due to occlusion and bright sunlight in outdoor tests.

6.2. Land Visibility Calculation Accuracy

Scenario	Actual Free Area (%)	Computed (%)	Deviation (%)
Low Crowd	85	83	±2%
Medium Crowd	55	52	±3%
High Crowd	18	21	±3%
Very High Crowd	8	11	±3%

Table 3: Land Visibility Calculation Accuracy

Land visibility estimation remained within ±3% deviation, confirming it as a reliable secondary metric for crowd density estimation.

6.3. System Response Time

Operation	Time Taken
YOLO Detection per Frame	55–70 ms
Land Visibility Computation	8–12 ms
ThingSpeak Data Upload	1.2–1.6 sec
ESP32 Data Fetch Interval	Every 15 seconds (configurable)
Overall Alert Latency	2–3 seconds

Table 4: System Response Time Analysis

6.4. Comparison of Expected vs. Actual Results

Metric	Expected	Achieved
Detection Accuracy	> 85%	90–100%
Land Visibility Deviation	< ±5%	±3%
IoT Stability (Uptime)	90–100%	100%
Alert Accuracy	High	High
System Latency	< 5 sec	2–3 sec

Table 5: Comparison of Expected vs. Actual Results

The dual detection strategy (YOLO + Land Visibility) significantly improved accuracy and reduced false alarms. ThingSpeak IoT integration proved seamless, with 100% upload success rate and no connection drops during testing. Alerts were triggered at correct thresholds, demonstrating proper synchronization between sender and receiver.

VI. CONCLUSION

This paper presented an Intelligent Crowd Detection and Accident Prevention System combining YOLOv8 person detection, OpenCV land visibility analysis, and IoT-based alert communication. The system achieved approximately 90–100% detection accuracy and ±3% land visibility estimation accuracy, with an overall alert latency of 2–3 seconds, making it suitable for real-world deployment.

The proposed system is cost-effective, lightweight, and easily adaptable to various environments including stadiums, religious gatherings, transport hubs, shopping malls, and university campuses. It provides a reliable and automated solution for ensuring safety in high-density crowd situations without requiring GPU hardware or expensive infrastructure.

VII. FUTURE SCOPE

- Adaptive Lighting Handling: Applying adaptive histogram equalization and shadow-removal for better performance in low-light and outdoor conditions.
- Edge Device Integration: Deploying on Raspberry Pi or Jetson Nano for distributed smart city monitoring.
- Hybrid AI Models: Combining OpenCV with lightweight ML models to increase robustness under heavily occluded conditions.
- Privacy-Preserving Design: Anonymizing video data to meet ethical and legal privacy requirements.
- Large-Scale Deployment Testing: Validating scalability and reliability in real-world settings such as stadiums, transport terminals, and festival grounds.

REFERENCES

1. J. Ma, Y. Dai, and K. Hirota, "A Survey of Video-Based Crowd Anomaly Detection in Dense Scenes," Journal of Advanced Computational Intelligence, vol. 21, no. 2, pp. 235–246, 2017.

2. S. Elbishlawi et al., "Deep Learning-Based Crowd Scene Analysis Survey," Journal of Imaging, vol. 6, no. 9, p. 95, 2020.
3. Y. Wang and F. Luo, "A Survey of Crowd Counting Algorithm Based on Domain Adaptation," Academic Journal of Science and Technology, vol. 5, no. 2, pp. 35–37, 2023.
4. A. Khan et al., "Crowd Monitoring and Localization Using Deep Convolutional Neural Network: A Review," Applied Sciences, vol. 14, no. 14, p. 4781, 2020.
5. P. Kannan et al., "Real-Time Crowd Monitoring and Management," IRJAEH, 2025.
6. M. A. Khan, H. Menouar, and R. Hamila, "LCDnet: A Lightweight Crowd Density Estimation Model for Real-Time Video Surveillance," Pre-print, 2023.
7. A. Redmon et al., "YOLOv4: Optimal Speed and Accuracy of Object Detection," arXiv, 2020.
8. V. Ramabai et al., "Using Existing CCTV Network for Crowd Management Using AI/ML," IJRASET, 2025.