Smart Occupancy Monitoring System

(Project Report)

Team Name: Shadow Count

Team Number: 8

Batch: 30

Team Members :

- Vinayak Dubey (Team Lead) – S25CSEU0893

- Harsh Ahlawat – S25CSEU0883

- Md Rahbar Anwar – S25CSEU0897

- Himanshu Yadav – S25CSEU0894

Abstract :

Smart occupancy monitoring systems play a crucial role in modern infrastructure by enabling automated crowd tracking and efficient space management. This project presents a real-time occupancy monitoring solution using YOLO (You Only Look Once) and OpenCV. The system detects people, tracks their movement through a designated entry/exit line, and calculates the number of individuals inside a defined space. It also logs data for further visualization and analysis. The project aims to offer a simple, scalable, and cost-effective solution that can be deployed in classrooms, labs, offices, or public spaces.

1. Introduction :

Monitoring the number of people inside a room or facility is essential for safety, crowd control, and resource management. Manual counting is inefficient and error-prone, especially in high-traffic areas. With advancements in computer vision, automated real-time occupancy monitoring has become achievable using lightweight AI models.

This project leverages the YOLO object detection model integrated with OpenCV to detect and track individuals in a video feed. The system identifies crossing events, distinguishes between "IN" and "OUT" movements, and maintains a live occupancy count. Additionally, data is saved in a CSV file to generate meaningful graphs and insights.

2. Problem Statement :

Traditional methods of tracking occupancy require human supervision and are not suitable for continuous monitoring. Existing solutions may be expensive or require complex hardware. The need is for a low-cost, real-time, automated system that can accurately track occupancy and provide visual analytics.

3. Objectives :

- Detect and track human movement in real time using YOLO.
- Identify entry and exit patterns by tracking crossing over a virtual line.
- Maintain a live count of total people inside a room.
- Log all events (timestamp, direction, occupancy) in a CSV file.
- Generate visual graphs for analysis of footfall and occupancy trends.
- Create an efficient and easily deployable system suitable for various environments.

4. Proposed Solution :

The project uses a webcam or video input processed frame-by-frame. YOLO identifies persons in the video, and each person is assigned a unique ID for tracking. When a person crosses the defined virtual line, the system classifies it as an entry or exit. The count is updated live, displayed on the video feed, and logged for analysis. At the end of execution, graphs are automatically generated using Matplotlib.

5. System Architecture :

1. Video Input – Webcam or external camera feed.

2. Object Detection – YOLO detects persons (class 0).

3. Tracking Module – Assigns unique IDs and tracks movement.

4. Line Crossing Logic – Determines whether a person entered or exited.

5. CSV Logging – Saves timestamp, ID, direction, and occupancy count.

6. Visualization – Generates graphs.

7. Output Display – Live annotated feed showing counts and detections.

6. Technologies Used :

Python, OpenCV, YOLO (Ultralytics), cvzone, CSV Module, Datetime, Matplotlib, Collections.

7. Methodology :

Detection and Tracking: YOLO identifies each person in the frame and assigns IDs.

Line Crossing Logic: Vertical line used to classify IN/OUT movement.

Occupancy Count Calculation: inside_count = total_in - total_out.

CSV Logging: Saves all events for analysis.

Graph Generation: Plots occupancy vs time and entries per interval.

8. Results & Discussion :

The system successfully monitored real-time occupancy with high accuracy. Entry and exit events were correctly detected, and the occupancy count updated instantly. Graphs help visualize movement patterns.

9. Applications :

- Classroom and lab occupancy monitoring
- Office room utilization
- Queue management
- Retail footfall tracking
- Public space crowd analysis
- Library automation

10. Limitations :

- Accuracy decreases in poor lighting
- Occlusion may affect tracking
- Works best when camera faces the entry directly
- Fast-moving crowds can challenge tracking IDs

11. Future Enhancements :

- Add heatmaps
- Use YOLOv8 for higher accuracy
- Integrate IoT sensors
- Build real-time dashboards
- Multi-camera support

12. Individual Contributions :

- Vinayak Dubey (Team Lead): Testing, coordination, debugging, analysis.

- Harsh Ahlawat: Main coding, YOLO integration, tracking logic, data logging.

- Md Rahbar Anwar: Testing support, documentation, workflow preparation.

- Himanshu Yadav: Report preparation, structure, PPT support.

13. Conclusion :

The Smart Occupancy Monitoring System demonstrates how computer vision can automate real-time crowd tracking effectively. By integrating YOLO with OpenCV, the system offers a practical solution for institutions and commercial setups. The project provides a strong foundation for further enhancements.

14. References :

- Ultralytics YOLO Documentation

- OpenCV Python Guide

- Python Official Documentation

- Matplotlib Reference