Infosys Springboard Virtual Internship 6.0

Completion Report

Crowd Count: Real-Time People Counting and Crowd Monitoring System

Submitted by:

Name: Shanta Angadageri

Mentor: Asma S

Batch: 3

Start Date: 22 September 2025

End Date: 21 November 2025

ABSTRACT

This project presents a real-time crowd counting and zone-based analytics system using YOLOv8 object detection and OpenCV. The aim is to detect people, count them, analyze zone occupancy, and generate actionable insights using data visualization techniques. The system processes live video from a webcam, detects individuals, identifies crowd zones, and updates crowd metrics continuously. It combines deep learning, computer vision, and analytics to provide a reliable solution for surveillance, safety, and crowd management applications.

CHAPTER 1 – INTRODUCTION

Crowd counting has become an essential technology in modern surveillance systems. With the rapid increase in urbanization, monitoring public spaces such as malls, airports, railway stations, and event venues has become crucial. Traditional methods rely heavily on human monitoring, which is prone to error and requires significant manpower. Automated crowd counting provides a scalable, accurate, and real-time solution.

This project uses YOLOv8—one of the most advanced deep learning object detection models—to identify and count people in live video streams. The system is capable of processing each frame, identifying the total number of people, and calculating how many individuals are inside specific zones. These results are further analyzed using charts and heatmaps, enabling deeper insights into movement patterns.

CHAPTER 2 – OBJECTIVES

The main objectives of this project are:

- To build an automated system for real-time crowd detection and analysis.
- To detect and count the number of people present in a video frame.
- To define specific zones and determine zone-wise occupancy.

- To generate visual analytics such as line charts, bar graphs, and heatmaps.

- To improve crowd monitoring using artificial intelligence and computer vision.

CHAPTER 3 – SYSTEM REQUIREMENTS

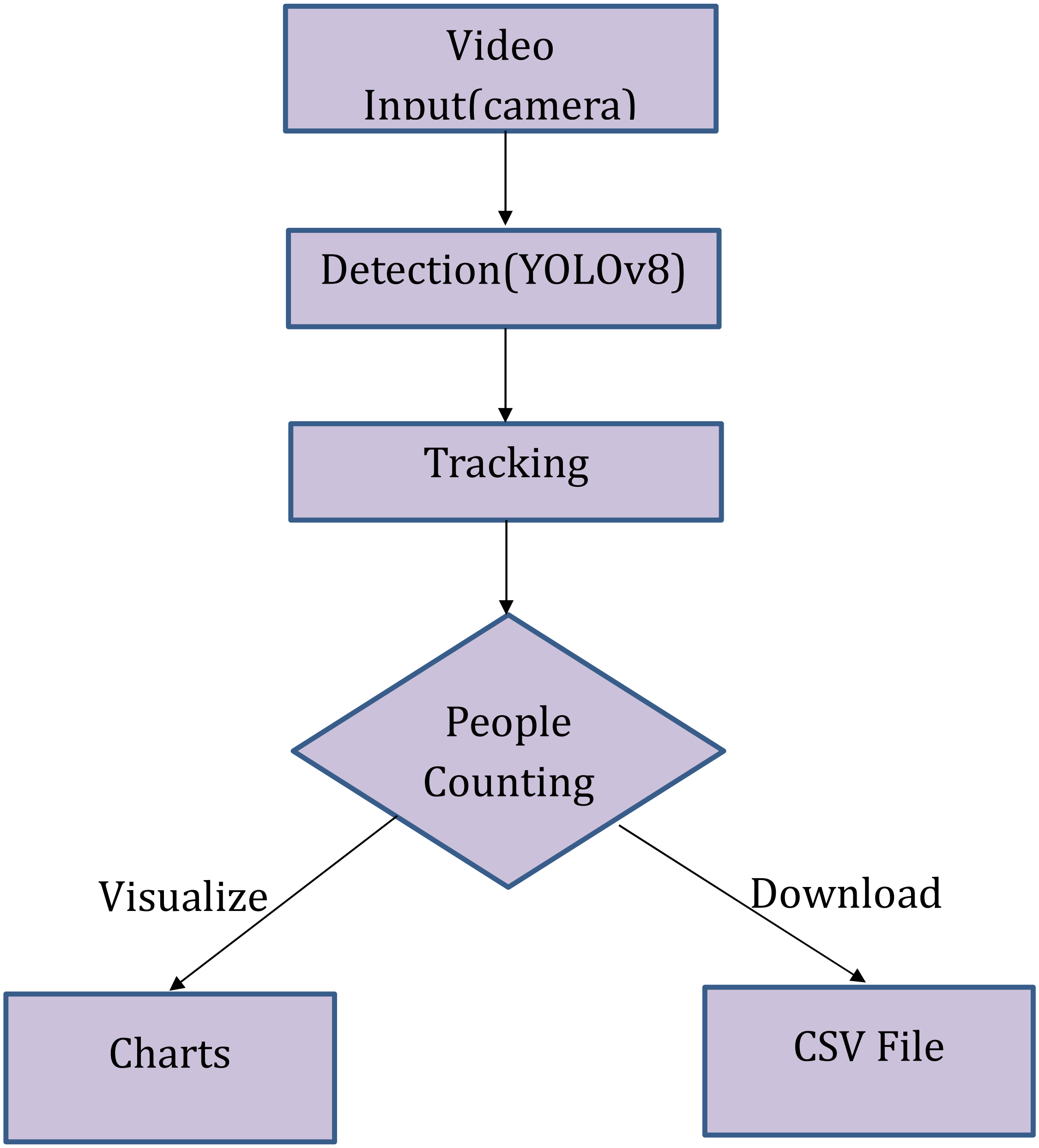
3.1 Hardware Requirements:

- Laptop/PC
- Webcam/External camera
- Processor

3.2 Software Requirements:

- Python 3.10+
- OpenCV
- Ultralytics YOLOv8
- Flask (for web interface, optional)
- Pandas and NumPy
- Matplotlib/Seaborn

CHAPTER 4 – PROPOSED MODEL



CHAPTER 5 - TOOLS AND TECHNOLOGIES

Component	**Technology**	**Role in the Project**
Web Framework	Flask	Main backend web server for routing, authentication, and UI logic
Database	SQLite	Secure storage of user credentials, roles, and sessions
Authentication	JWT (PyJWT)	User authentication; role-based access; secure session management
Object Detection	YOLOv8 (Ultralytics)	The Deep Learning Model used to Detects people in video and image frames
Object Tracking	ByteTrack	Tracks detected people across frames for consistent zone counting
Video/Image Input Processing	OpenCV (cv2)	Reads video/image files, processes frames, draws bounding boxes

Component	Technology	Role in the Project
Data Visualization	Matplotlib, Pandas	Generates charts, graphs, heatmaps, and CSV reports
Front-End	HTML, CSS, javascript	Provide structure and styling for web pages
Real-Time Video Streaming	OpenCV, Flask (Response)	Provides live object detection video streams in the browser
Live & Historic Analytics	Python + Visualization	Enables live zone counts, population trends, and report downloads
User Management	Flask, SQLite	Registration, login, user-role checks, and admin features

The architecture consists of the following modules:

1. Video Input Module – captures video frames from a webcam or file.

2. Detection Module – YOLOv8 performs object detection on each frame.

3. Zone Detection Module – checks if detected persons fall within predefined zones.

4. Tracking Module – maintains unique IDs for each frame.

5. Analytics Module – uses NumPy and Pandas for data handling.

6. Visualization Module – generates charts and heatmaps.

CHAPTER 6 – METHODOLOGY

6.1 Video Capture

The system uses OpenCV’s VideoCapture() function to retrieve frames from the webcam.

6.2 Person Detection

Each frame is processed using YOLOv8. Bounding boxes are extracted, converted to NumPy arrays, and filtered by class ID (person).

6.3 Zone Monitoring

The central point of each detected person is calculated. Persons whose center falls inside the zone are counted.

6.4 Data Storage

Counts are stored using Pandas DataFrames for graph generation.

6.5 Visualization

Different visualization techniques include:

- Line Chart – showing population changes over time.
- Bar Chart – comparing zone and total counts.
- Heatmap – showing density and frequency of crowd accumulation.

CHAPTER 7 – IMPLEMENTATION

The implementation includes Python code for:

- Frame processing
- YOLO detection
- Unique ID generation
- Zone-based counting
- Text annotation
- Data plotting

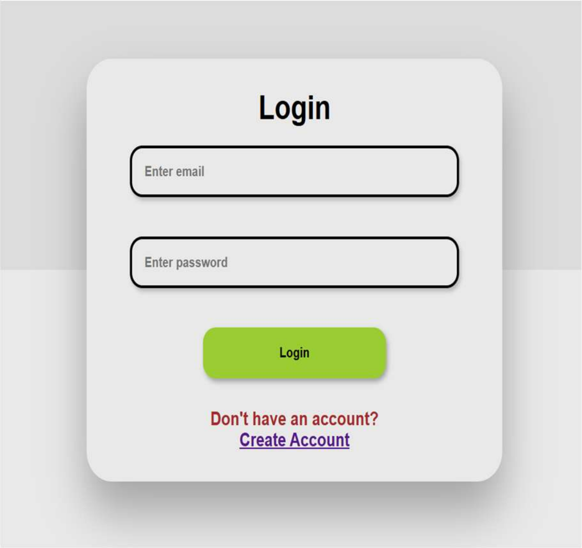
CHAPTER 8 – RESULTS

The system successfully detected and counted people in real time. Zone occupancy was accurately calculated. Visual charts clearly represented crowd variations over time.

Sample outcomes :

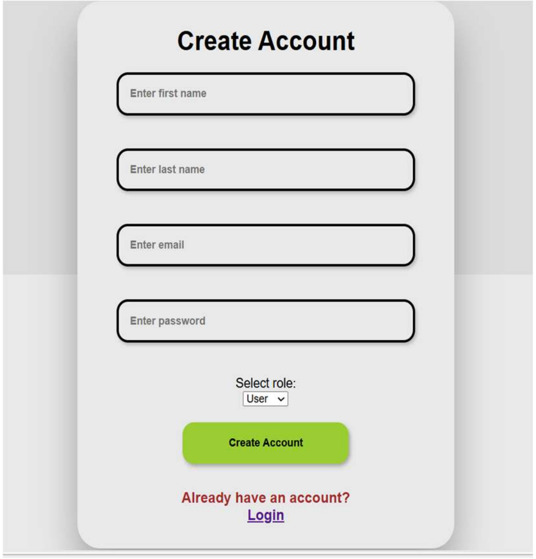
- Zone-wise crowd count
- Total headcount
- Real-time video with bounding boxes
- Line charts showing population trends

Login Page



Don't have an account?

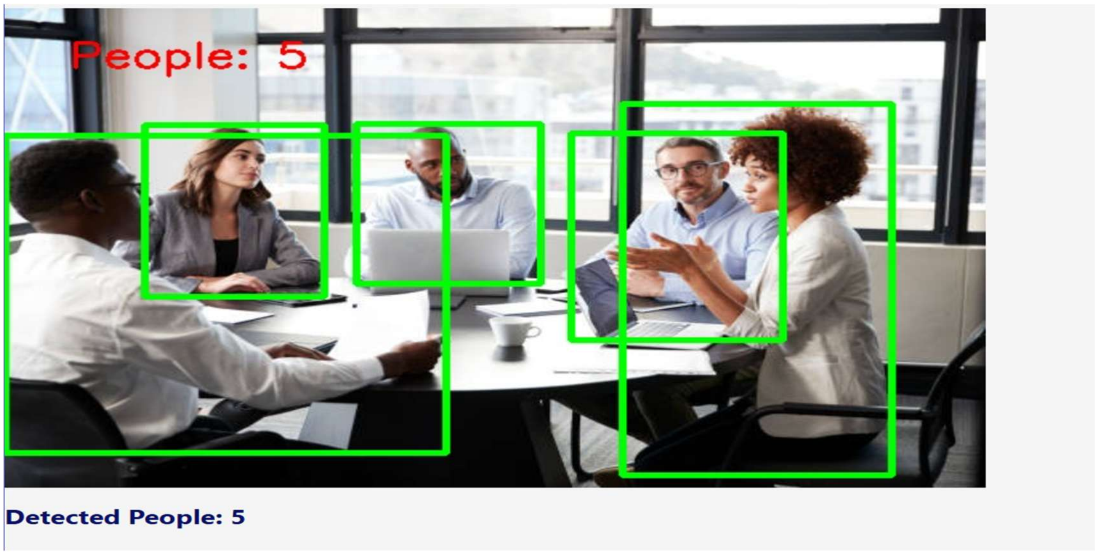
Registration Page



Select role: User

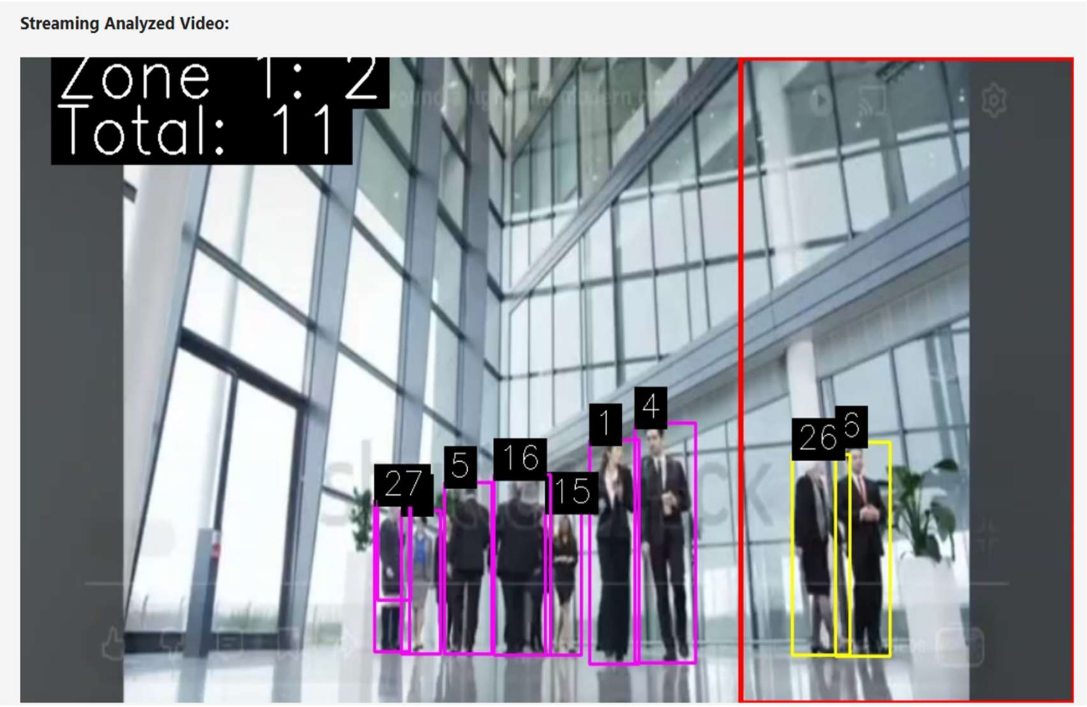
Already have an account?

People Count in Image

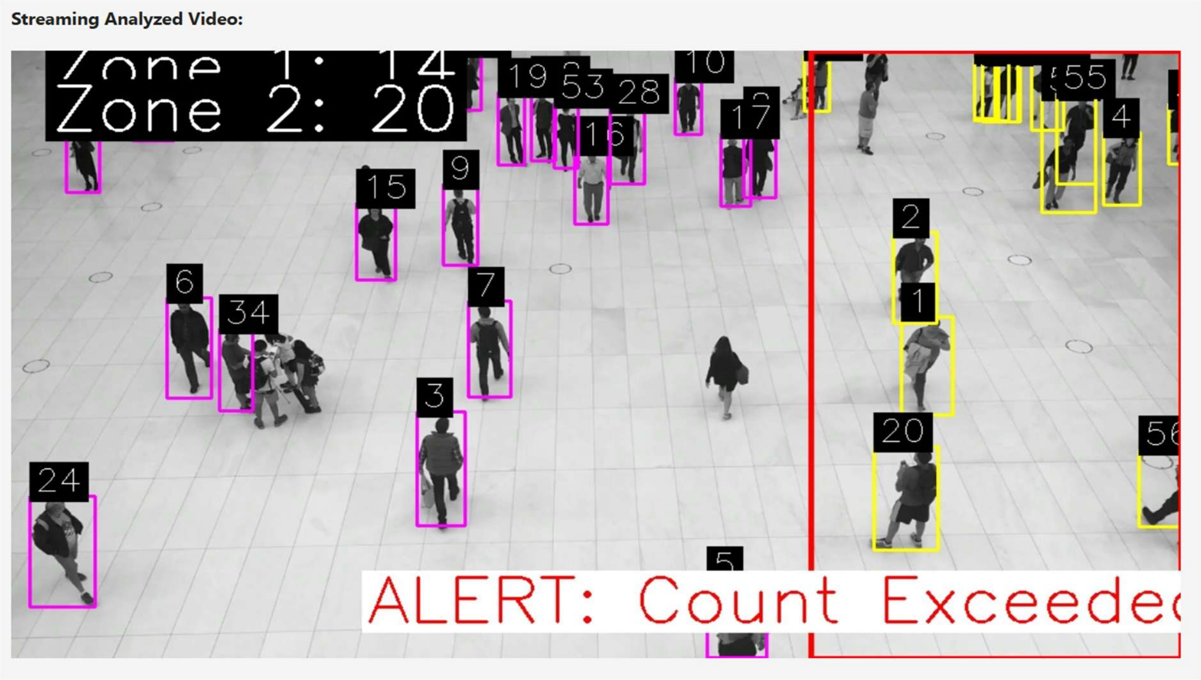


Detected People: 5

Zone Management in video stream

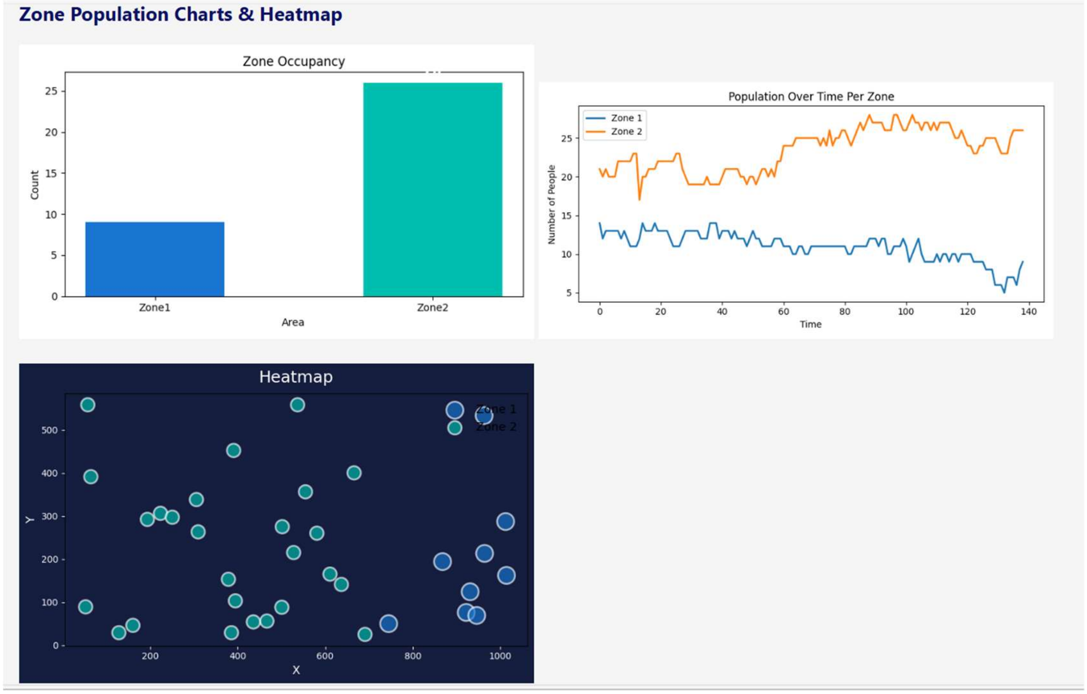

Streaming Analyzed Video:

Alert for count exceeded



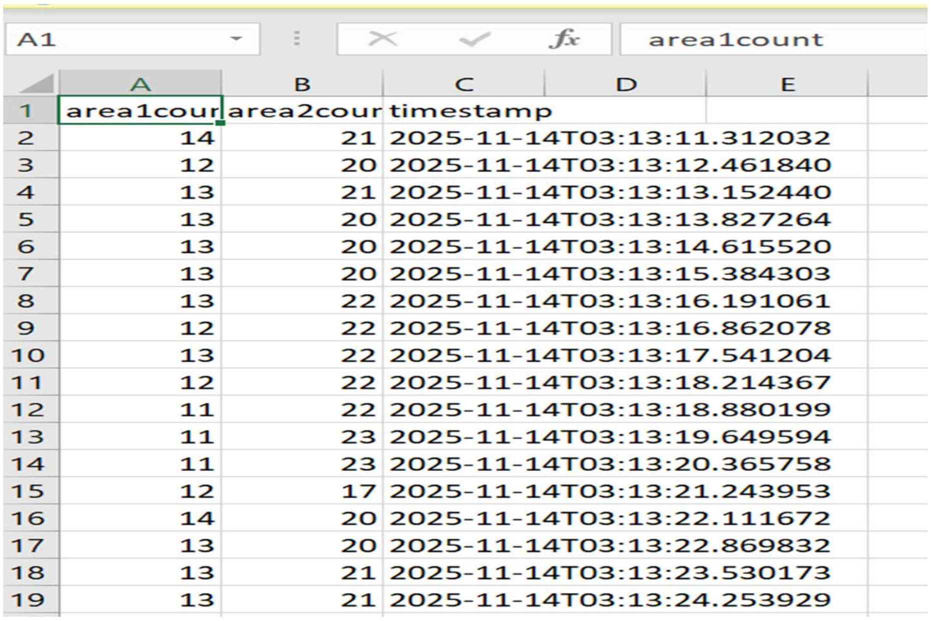
Streaming Analyzed Video:

Visualization



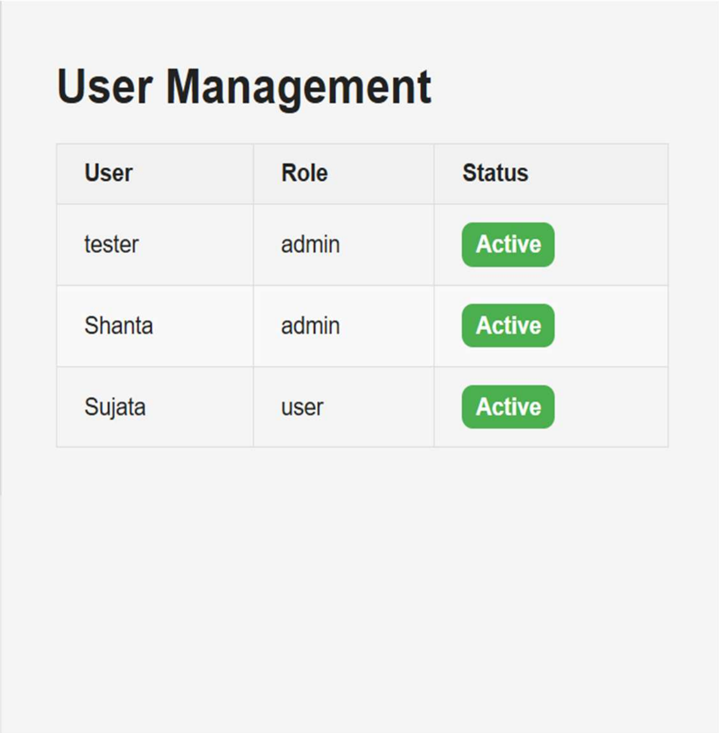
Zone Population Charts & Heatmap

Downloaded Report



	A	B	C	D	E
1	area1cour	area2cour	timestamp		
2	14	21	2025-11-14T03:13:11.312032		
3	12	20	2025-11-14T03:13:12.461840		
4	13	21	2025-11-14T03:13:13.152440		
5	13	20	2025-11-14T03:13:13.827264		
6	13	20	2025-11-14T03:13:14.615520		
7	13	20	2025-11-14T03:13:15.384303		
8	13	22	2025-11-14T03:13:16.191061		
9	12	22	2025-11-14T03:13:16.862078		
10	13	22	2025-11-14T03:13:17.541204		
11	12	22	2025-11-14T03:13:18.214367		
12	11	22	2025-11-14T03:13:18.880199		
13	11	23	2025-11-14T03:13:19.649594		
14	11	23	2025-11-14T03:13:20.365758		
15	12	17	2025-11-14T03:13:21.243953		
16	14	20	2025-11-14T03:13:22.111672		
17	13	20	2025-11-14T03:13:22.869832		
18	13	21	2025-11-14T03:13:23.530173		
19	13	21	2025-11-14T03:13:24.253929		

User Management



User Management

User	Role	Status
tester	admin	Active
Shanta	admin	Active
Sujata	user	Active

CHAPTER 9 – ADVANTAGES

- Works in real time
- High accuracy due to YOLOv8
- Easy-to-interpret visual analytics
- Useful for public safety, planning, and surveillance
- Can be extended to multi-camera environments

CHAPTER 10 – APPLICATIONS

Real-time crowd detection and analytics systems, like the one you've built, have a wide range of applications across public safety, operational management, urban planning, and live event coordination. Here are some of the key application areas, as supported by current industry use cases and technology trends:

- **Retail and Commercial Environments**: Analyze visitor density in malls, stores, and commercial venues to improve customer experience, optimize staffing, and reduce bottlenecks. Manage queues and adjust entry/exit flows based on live occupancy data.
- **Public Safety and Event Management:** Monitor crowd density at large gatherings such as concerts, sporting events, and religious festivals to prevent overcrowding and stampedes. Detect unusual or risky behaviors and generate real-time alerts for security teams
- **Transport Hubs and City Spaces:** Track crowd movement and occupancy at airports, train stations, and bus terminals for

efficient resource allocation and congestion prevention Assist city planners in analyzing pedestrian flow and optimizing urban design for safer and more accessible public spaces.

- **Threat Detection and Security:** Identify potential threats like overcrowding, stampedes, and abnormal crowd movements using AI and machine learning models. Facilitate quick incident response by integrating alerts and logs with emergency services systems.

CHAPTER 11 – FUTURE ENHANCEMENTS

- Sensor Fusion and IoT Integration: Connect multiple sensor types—such as cameras, LiDAR, thermal imaging, Wi-Fi, and RFID—to deliver more accurate crowd analysis, location tracking, and environmental awareness in real time.
- Behavior Analysis Beyond Counting: Enhance crowd analytics to recognize specific behaviors, sentiments, or anomalies (e.g., panic, aggression, distress) by integrating deep learning models capable of complex activity recognition.
- Edge Computing & Real-Time Response: Employ edge computing with 5G connectivity to process video and sensor data directly at the source, reducing latency and enabling instant alerts and interventions during large-scale events.
- AI-Powered Predictive Analytics: Integrate advanced AI models that use historical and real-time data to forecast crowd behaviors, movement patterns, and potential risks, enabling proactive safety measures and optimized event planning.

CHAPTER 12 – CONCLUSION

This project demonstrates an efficient, AI-based approach for real-time crowd counting using YOLOv8 and OpenCV. By analyzing video frames, detecting people, and generating analytics, the system provides a robust solution for modern surveillance and public monitoring systems. The combination of machine learning, analytics, and visualization makes this project powerful and highly practical.

REFERENCES

1. Ultralytics YOLO Documentation

2. OpenCV Library Documentation

3. Research papers on crowd analytics

4. Deep learning books and online resources