Human Congestion Monitoring System Using YOLOv8, Deep SORT, and Re-Identification for Real-Time Crowd Analysis and Missing Person Detection

Prof. P. G. Ligade[1], Mr. Chaitanya Utekar[2], Mr. Ajinkya Naik[3], Mr. Prathmesh Bhatpure[4], Mr. Sahil Lukade[5]

[1]*Professor, Department of Information Technology, Sinhgad Institute of Technology and Science, Narhe, Pune.*

[2,3,4,5]*UG Student, Department of Information Technology, Sinhgad Institute of Technology and Science, Narhe, Pune.*

***Abstract*—Managing large crowds in public spaces such as railway stations, malls, and public events poses significant safety and security challenges. Overcrowding, sudden surges, and the need to locate missing individuals demand automated, real-time solutions that go beyond conventional surveillance. This paper presents a Human Congestion Monitoring System that integrates YOLOv8-based person detection, Deep SORT multi-object tracking, ResNet-50 Re-Identification (ReID), and FAISS-based vector search to enable live crowd counting, density thresholding, surge detection, and missing person tracking through a unified web-based dashboard. The backend is built with FastAPI and communicates with the frontend in real time via Web Sockets. Alerts for crowd threshold breaches, sudden surges, and missing person matches are generated automatically and persisted in a SQLite database. Experimental results on live webcam streams and uploaded video files confirm the system's effectiveness, achieving accurate person counts, responsive alerting, and reliable missing person re-identification across multiple video sources.**

***Index Terms*—Human Congestion Monitoring, YOLOv8, Deep SORT, Re-Identification (ReID), FAISS, FastAPI, Web-Socket, Missing Person Detection, Crowd Density, Real-Time Surveillance**

I. INTRODUCTION

The rapid growth of urban populations and the increasing frequency of mass gatherings at public venues have intensified the demand for intelligent crowd management systems. Traditional CCTV surveillance relies heavily on human operators, making it prone to fatigue, delayed response, and scalability limitations. Automated systems capable of counting people, estimating crowd density, detecting dangerous surges, and identifying missing individuals can significantly reduce public safety risks.

Recent advancements in deep learning have made real-time object detection and tracking viable on commodity hardware. Convolutional neural networks such as the YOLO family [1] provide high-speed, accurate person detection, while multi-object tracking algorithms like Deep SORT [3] maintain individual identities across video frames. Re-identification (ReID) techniques [4] further enable matching of individuals across camera views or over time, forming the backbone of missing person detection.

This paper describes the design, implementation, and evaluation of Crowd Watch, a Human Congestion Monitoring System that:

- Detects and counts persons in real time from webcam or uploaded video using YOLOv8n.
- Tracks individuals across frames using Deep SORT with a ResNet-50 ReID embedding module.
- Maintains a FAISS vector index of all observed person embeddings for fast similarity search.
- Alerts operators when the live count crosses a configurable threshold (manual, area-based WHO standard, or rolling statistical auto mode).
- Detects missing persons by matching a reference photo embedding against the live ReID index in real time.
- Broadcasts all events to a web dashboard via WebSocket with no polling latency.

The remainder of this paper is structured as follows. Section II surveys related work. Section III describes the proposed system architecture and methodology. Section IV details the implementation. Section V presents experimental results and testing. Section VI discusses future scope, and Section VII concludes the paper.

II. RELATED WORK

A. Object Detection for Crowd Analysis

YOLO (You Only Look Once) [1] revolutionized real-time object detection by framing it as a single regression problem. YOLOv8 [2] further improves speed and accuracy with an anchor-free head and a CSP-Darknet-based backbone, making it well suited for person detection in crowded scenes. Sindagi and Patel [6] provide a comprehensive survey of crowd counting methods, noting that detection-based approaches outperform regression-based ones when individual localisation is required.

B. Multi-Object Tracking

Deep SORT [3] extends the original SORT tracker by incorporating a deep appearance descriptor, significantly reducing identity switches compared to purely motion-based trackers. It uses a Kalman filter for state prediction and the Hungarian algorithm for data association, combined with a learned cosine distance metric for appearance matching. This combination makes it highly effective for tracking people in moderate to dense crowds.

C. Person Re-Identification

Person ReID aims to match individuals across different cameras or time intervals. Zheng et al. [4] established early benchmarks and demonstrated that CNN-based feature ex-traction substantially outperforms hand-crafted descriptors. ResNet-50 [5] pre-trained on ImageNet has been widely adopted as an effective backbone for ReID feature extraction, producing 2048-dimensional embeddings that encode rich appearance information.

D. Crowd Density Estimation and Alerting

Li et al. [7] proposed CSRNet, a dilated CNN for high-density crowd counting via density map estimation. While accurate in extreme density scenarios, detection-based meth-ods remain preferable when precise individual tracking and identity-aware alerts are required. WHO crowd safety guide-lines [8] define dangerous crowd density as exceeding 4 people per square metre, motivating the area-based threshold mode implemented in this system.

E. Missing Person Detection

Automated missing person systems using surveillance video are an active research area. Ye et al. [9] demonstrated that ReID networks trained on large-scale pedestrian datasets generalise well to unseen environments when combined with efficient nearest-neighbour search. FAISS [10], a library for efficient similarity search in high-dimensional spaces, provides the retrieval backbone needed for real-time ReID matching at scale.

III. SYSTEM ARCHITECTURE AND METHODOLOGY

A. Overall Architecture

The system follows a modular pipeline architecture, as illustrated in Fig. 1. Video frames from a webcam or up-loaded video are processed by a detection module, passed to a tracker, enriched with ReID embeddings, indexed in a vector store, and evaluated for alert conditions. All results are broadcast to connected dashboard clients via WebSocket.

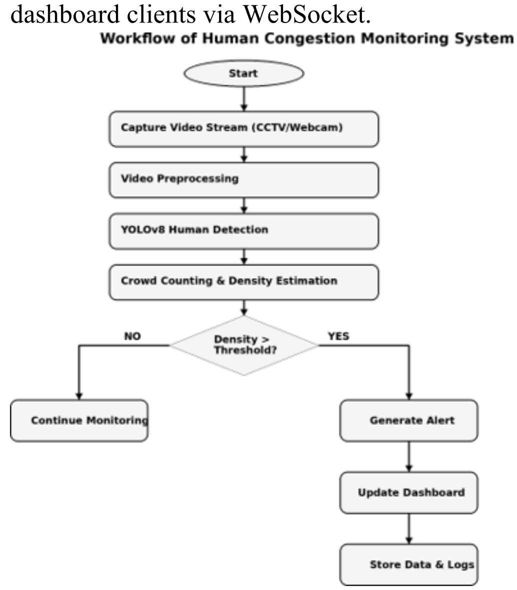

Fig. 1. Workflow of the Human Congestion Monitoring System.

B. Person Detection Module

YOLOv8n (nano variant) is used for person detection due to its balance of speed and accuracy on CPU and GPU. Before inference, each frame undergoes

CLAHE (Contrast Limited Adaptive Histogram Equalisation) pre-processing on the L channel in LAB colour space. This normalises uneven lighting conditions common in indoor and low-light surveillance scenarios. Detection is configured with:

- Confidence threshold: 0.40 (catches partially visible or distant people)
- IoU threshold: 0.40 (better separation in tight crowds)
- Class filter: person (class 0) only
- Aspect ratio filter: width < 2.5× height (eliminates non-person shapes)
- Minimum bounding box: 10 × 20 pixels

C. Multi-Object Tracking with Deep SORT

Confirmed detections are passed to a Deep SORT tracker configured with:

- max_age = 15: holds a lost track for 15 frames before dropping it, reducing re-identification overhead.
- n_init = 2: confirms a track after 2 consecutive detections, filtering spurious detections.
- nms_max_overlap = 0.5: suppresses duplicate tracks in crowded scenes.
- max_cosine_distance = 0.4: tight ReID matching constraint to minimise identity switches.

Only confirmed tracks contribute to the crowd count and ReID index. A count smoother with exponential moving average ($\alpha = 0.8$) reduces display jitter without sacrificing responsiveness.

D. Re-Identification Module

For each confirmed track, the person crop is extracted and passed to a ReID model built on ResNet-50 with the final fully connected layer replaced by an identity layer, yielding a 2048-dimensional feature vector. The vector is L2-normalised so that Euclidean distance approximates cosine similarity. The model is pre-trained on ImageNet and used as a strong generic appearance encoder without domain-specific fine-tuning.

The embedding transform pipeline is:

$$\hat{\mathbf{e}} = \frac{f_\theta(\text{Resize}(I_{128\times256}))}{\|f_\theta(\text{Resize}(I_{\ 128\times256}))\|_2 + \epsilon}$$

where fθ is the ResNet-50 encoder and ϵ = 10−6 prevents division by zero.

E. FAISS Vector Index

All normalized embeddings are stored in a FAISS IndexFlatL2 index alongside metadata (camera ID, times-tamp, track ID, bounding box). This supports two use cases:

1. Live missing person matching: On every webcam frame, each new embedding is compared to the stored reference embedding. A MISSING_PERSON_FOUND alert fires when the L2 distance is below 0.80.
2. One-shot index search: The user can upload a photo at any time to retrieve the top-k closest matches from the entire historical index.

F. Alert and Threshold System

The system supports three threshold modes:

1. Manual: Operator sets an absolute person count thresh-old.
2. Area-based: Threshold is auto-computed from the mon-itored area in km2 using the WHO crowd safety standard of 4 people/m2:

Tarea = min(5000, max(1, Akm2 × 4,000,000)) (2)

3. Rolling auto: A rolling window of the last 200 count samples sets the threshold dynamically as μ + 1.5σ, adapting to baseline crowd levels without manual cali-bration.

Three alert types are generated by the heuristic’s module:

- CROWD_DENSITY: live count ≥ active threshold.
- SUDDEN_SURGE: rate of increase > 5 people/second over the last 10 frames.
- PUSHING_LIKELY: bounding box overlap ratio among detected persons exceeds 20% of all possible pairs, indicating overcrowding.

IV. IMPLEMENTATION

A. Technology Stack

Table I Summaries the core technologies used.

Component	Technology
Detection Model	YOLOv8n (Ultralytics)
Tracking	Deep SORT Realtime
ReID Backbone	ResNet-50 (PyTorch / torchvision)
Vector Search	FAISS (IndexFlatL2, dim=2048)
Backend Framework	FastAPI + Uvicorn
Real-time Comms	WebSocket (native FastAPI)
Image Processing	OpenCV, NumPy, Pillow

Database	SQLite (via Python sqlite3)
Frontend	HTML5, CSS3, Vanilla JavaScript
Configuration	Python dotenv

B. Backend Design

The backend is a FastAPI application structured into the following modules:

- main.py: Application entry point, WebSocket hub, REST endpoints for threshold control, missing person management, webcam control, and static file serving.
- inference.py: Asynchronous video stream processor for uploaded video files.
- models/detector.py: CLAHE pre-processing and YOLOv8 inference wrapper.
- models/tracker_reid.py: Deep SORT tracker, ResNet-50 ReID model, and count smoother.
- models/action_heuristics.py: Per-camera alert logic for density, surge, and overlap events.
- store/reid_index.py: Thread-safe FAISS wrapper for embedding storage and retrieval.
- utils/db.py: SQLite initialization and alert persistence.

The main event loop stores a reference at startup, enabling background threads (webcam, video processor) to dispatch WebSocket alerts safely via asyncio.run_coroutine_threadsafe.

C. Webcam Processing Pipeline

The webcam thread runs at approximately 20 fps (50 ms sleep interval). For each frame:

1. CLAHE pre-processing is applied.
2. YOLOv8n detects persons.
3. Deep SORT updates track states; only confirmed tracks are counted.
4. ResNet-50 extracts a 2048-dim embedding for each confirmed crop.
5. Embeddings are added to the FAISS index with meta-data.
6. Missing person distance check fires an alert if L2 distance < 0.80.
7. Crowd density alert fires (throttled to once per 5 sec-onds) if count ≥ threshold.
8. Count is written to the shared camera count dictionary for broadcast.

D. WebSocket Broadcast

A background coroutine broadcasts a STATS_UPDATE message to all connected clients every 1 second, containing per-camera counts, total count, active threshold, mode, and missing person tracking status. Alerts are pushed immediately upon detection using the same WebSocket connections.

E. Frontend Dashboard

The single-page dashboard is built in plain HTML/CSS/JavaScript and provides:

- Live person counts card, active threshold, and threshold mode display.
- Real-time crowd density line chart updated on every WebSocket message.
- Webcam control (Start/Stop) with live detection count feedback.
- Threshold control panel (manual count, area in km2, rolling auto toggle).
- Missing person panel: photo upload, tracking activation, one-shot index search with timestamped results.
- Scrollable alert feed showing all real-time events with severity highlighting.

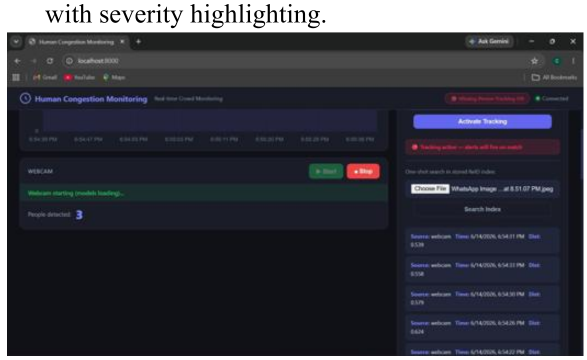

Fig. 2. Dashboard showing live person count, crowd density chart, and threshold control panel

V. RESULTS AND EVALUATION

A. System Testing

The system was evaluated on live webcam streams and pre-recorded video files. Table II summarises the functional test cases and their outcomes.

B. Performance Observations

Table III records the key performance characteristics observed during testing.

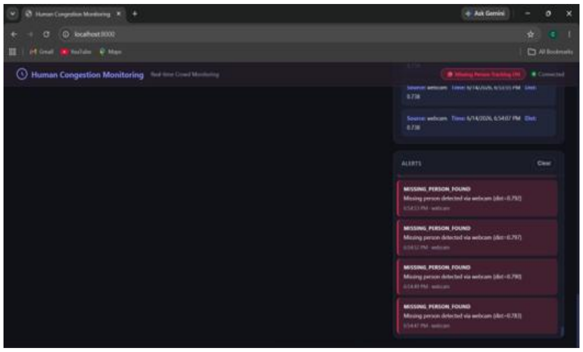

Fig. 3. Missing person tracking panel showing ReID search results with distance scores and timestamps

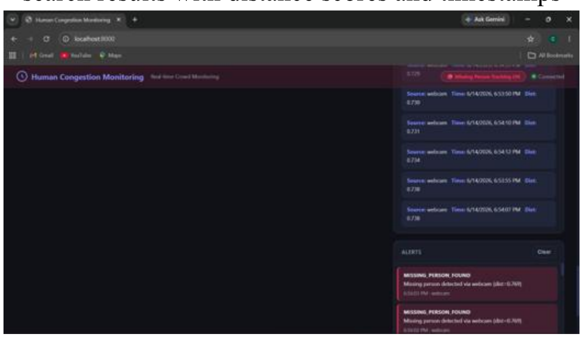

Fig. 4. Alert feed showing MISSING_PERSON_FOUND events with detection distances.

C. Missing Person Detection

During live webcam testing, the missing person tracking module successfully detected matches with L2 distances in the range 0.729–0.797 for the same individual appearing in different frames, as visible in the dashboard alert feed (Fig. 4). The one-shot search returned ranked results from the historical ReID index, enabling retrospective identification of when and where a person was last seen.

D. Crowd Density Alerting

With manual threshold set to 1 person for demonstration purposes, the dashboard correctly displayed crowd count increments and generated threshold alerts in real time. The rolling auto mode adjusted the threshold dynamically, reducing false positives during periods of stable crowd levels.

VI. FUTURE SCOPE

1. Multi-camera aggregation: Extend the system to handle simultaneous RTSP streams from multiple IP cam-eras with a unified per-zone crowd count.
2. Domain-specific ReID fine-tuning: Fine-tune the ResNet-50 encoder on surveillance-specific datasets (e.g., Market-1501, DukeMTMC) to improve re-identification accuracy beyond ImageNet pre-training.

Table II Functional test cases and results

Test Case	Expected Result	Status
Webcam start/stop	Stream starts and stops cleanly	Pass
Live person count	Accurate live count displayed	Pass
Manual threshold set	Alert fires on count ≥ threshold	Pass
Area-based threshold	WHO density auto-computed correctly	Pass
Rolling auto threshold	Adapts to baseline crowd level	Pass
Missing person upload	Embedding extracted and stored	Pass
Missing person tracking	Alert fires on match (dist < 0.80)	Pass
One-shot ReID search	Top-30 matches returned with distances	Pass
Sudden surge detection	Alert on >5 people/sec increase	Pass
Overlap/pushing detection	Alert on ≥20% bbox overlap ratio	Pass
Alert persistence	Events saved to SQLite DB	Pass
WebSocket broadcast	UI updates within 1 second	Pass
Multi-video source	Counts aggregated per camera	Pass
Large video processing	Processed without crash	Pass

Table III Performance observations

Parameter	Observation
Detection model	YOLOv8n
Tracking algorithm	Deep SORT Realtime
ReID backbone	ResNet-50 (2048-dim)
Webcam capture rate	~20 fps
Alert broadcast latency	≤1 second
Missing person dist. threshold	0.80 (L2, normalised)
FAISS index type	IndexFlatL2
Database	SQLite
Threshold modes	Manual, Area-based, Rolling Auto
Tested video formats	MP4 (H.264)

3. Trajectory analysis: Track individual movement paths to detect loitering, counter-flow in evacuation corridors, or abnormal trajectory patterns.
4. Edge deployment: Optimise the pipeline for

deployment on NVIDIA Jetson or Raspberry Pi using Ten-sorRT or ONNX quantisation of both YOLOv8 and the ReID model.
5. SMS/email notification: Integrate alert forwarding via Twilio or SMTP so security personnel are notified even when not watching the dashboard.
6. Density heatmap visualization: Overlay spatial crowd density heatmaps on a floor plan or camera view to identify congestion hotspots.
7. Face-anonymised privacy mode: Apply face blurring before display to comply with privacy regulations while retaining body-based ReID functionality.

VII. CONCLUSION

This paper presented a real-time Human Congestion Monitoring System that integrates YOLOv8 object detection, Deep SORT multi-object tracking, ResNet-50 Re-Identification, and FAISS-based similarity search into a unified, web-accessible platform. The system provides live crowd counting, configurable density alerting with three threshold modes aligned to WHO crowd safety standards, automatic surge and pushing detection, and real-time missing person identification. All components were implemented in Python using FastAPI and communicated to the browser via Web Socket, achieving sub-second alert latency. Comprehensive functional testing confirmed that all modules detection, tracking, ReID, alerting, and persistence operate correctly and reliably. The system demonstrates that a compact deep learning pipeline can serve as a practical crowd safety tool for venues ranging from university campuses to public events and transportation hubs.

REFERENCES

[1] Redmon J., Divvala S., Girshick R., and Farhadi A., “You Only Look Once: Unified, Real-Time Object Detection,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779–788.
[2] Jocher G., Chaurasia A., and Qiu J., “Ultralytics YOLOv8,” 2023. Available: Ultralytics YOLOv8 GitHub Repository
[3] Wojke N., Bewley A., and Paulus D., “Simple Online and Realtime Tracking with a Deep Association Metric,” in Proceedings of the IEEE International Conference on Image Processing (ICIP), 2017, pp. 3645–3649.
[4] Zheng L., Shen L., Tian L., Wang S., Wang J., and Tian Q., “Scalable Person Re-identification: A Benchmark,” in Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2015, pp. 1116–1124.
[5] He K., Zhang X., Ren S., and Sun J., “Deep Residual Learning for Image Recognition,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.
[6] Sindagi V. A. and Patel V. M., “A Survey of Recent Advances in CNN-based Single Image Crowd Counting and Density Estimation,” Pattern Recognition Letters, vol. 107, pp. 3–16, 2018.
[7] Li Y., Zhang X., and Chen D., “CSRNet: Dilated Convolutional Neural Networks for Understanding the Highly Congested Scenes,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 1091–1100.
[8] World Health Organization, Crowd Management, Crowd Safety and Crowd Survival: WHO Technical Guidelines, Geneva, Switzerland: World Health Organization, 2010.
[9] Ye M., Shen J., Lin G., Xiang T., Shao L., and Hoi S. C. H., “Deep Learning for Person Re-identification: A Survey and Outlook,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 44, no. 6, pp. 2872–2893, 2021.
[10] Johnson J., Douze M., and Jégou H., “Billion-Scale Similarity Search with GPUs,” IEEE Transactions on Big Data, vol. 7, no. 3, pp. 535–547, 2021.