AI-Based Real-Time Crowd Density Alert System

Suresh Kumar S
Professor of Artificial Intelligence and Data Science
Rajalakshmi Engineering College
Chennai, India
sureshkumar.s@rajalakshmi.edu.in

Benjamin Nicolas S
Artificial Intelligence and Data Science
Rajalakshmi Engineering College
Chennai, India
221801005@rajalakshmi.edu.in

Charless Binny K
Artificial Intelligence and Data Science
Rajalakshmi Engineering College
Chennai, India
221801007@rajalakshmi.edu.in

Harshitha R
Artificial Intelligence and Data Science
Rajalakshmi Engineering College
Chennai, India
221801018@rajalakshmi.edu.in

Abstract— In densely populated environments such as transportation terminals, religious gatherings, concerts, and public festivals, real-time crowd management is critical to ensuring safety and preventing hazardous incidents like stampedes and bottlenecks. Manual monitoring methods, typically involving security personnel observing CCTV footage, are labor-intensive, prone to human error, and often reactive rather than preventive. To address these limitations, this paper proposes an AI-Based Real-Time Crowd Density Alert System that leverages deep learning and computer vision to automate the detection, monitoring, and alerting process when crowd density exceeds predefined safety thresholds.

The system is built around the YOLOv8 (You Only Look Once, Version 8) deep learning model, a state-of-the-art object detection framework known for its speed and accuracy in identifying objects within real-time video streams. Using OpenCV, the system captures video input from either a USB-connected webcam or a smartphone camera (via IP streaming tools like Iriun or DroidCam) and processes each frame through YOLOv8 to detect individuals present in the scene. The total person count is then computed and compared against a user-defined threshold, beyond which automated alert mechanisms are activated.

Two primary forms of alerts are incorporated: (1) an audible alarm produced using the winsound library that emits a continuous beep for 15 seconds to draw immediate on-site attention, and (2) an email notification system that dispatches alert messages to pre-registered authorities using Python's built-in smtplib for remote intervention. To enhance operational efficiency and accessibility, a web-based interface developed using the Flask framework displays the live video stream with bounding boxes and detection labels, current person count, and alert status in real-time, accessible through any modern browser.

The proposed system has been implemented entirely in Python, leveraging additional libraries such as NumPy for numerical operations and Ultralytics for YOLOv8 model deployment. Experimental evaluation demonstrates that the system is capable of high-speed processing, low-latency detection, and reliable alert generation under varying lighting and environmental conditions. It offers a scalable, cost-effective, and easily deployable solution for integration into existing public surveillance infrastructures.

Future work may explore integration with edge computing devices for localized processing, multi-camera synchronization for large-area coverage, cloud-based data analytics for pattern recognition, and the use of IoT-driven automated control responses such as gate locking or crowd redirection. Additionally, incorporating advanced analytics such as movement prediction, behavior anomaly detection, and historical trend mapping can further enhance the intelligence and responsiveness of the system. With continued development, the system presents a promising contribution to AI-driven public safety and surveillance solutions, capable of supporting disaster prevention and efficient emergency response in critical environments.

Index Terms—Real-time surveillance, crowd density detection, YOLOv8, computer vision, Flask interface, public safety, AI monitoring, OpenCV, alert systems, deep learning, smart infrastructure, automated crowd control

I. INTRODUCTION

In ensuring that public safety mechanisms evolve in step with increasing population densities and urbanization, issues pertaining to real-time monitoring, proactive alerting, and autonomous decision-making remain largely unaddressed. Traditional surveillance systems, although widespread, depend heavily on manual observation and delayed response, which compromises their effectiveness during high-risk situations such as overcrowding, stampedes, or emergency evacuations. While CCTV coverage exists in most public settings, the absence of intelligent behavior analysis, human-like situational awareness, and automated notification protocols leaves significant gaps in critical response operations. Compounding these issues are the inefficiencies in manual crowd estimation, the lack of instant alerts, and the limited engagement of security personnel during real-time events, which together hinder the ability to prevent crowd-related disasters effectively.

Fig. 1. Tech Stacks

II. RELATED WORKS

Current Systems and Limitations in Crowd Management

The field of crowd management, particularly through the use of AI-based real-time systems, faces several challenges. The following sections explore the limitations of existing platforms and how these can be addressed by the **AI-Based Real-Time Crowd Density Alert System**.

A. Traditional Crowd Monitoring Platforms:

Current crowd monitoring systems often rely on manual methods or fixed, rule-based algorithms, which are inefficient and prone to human error. These systems typically offer basic alerts for crowd congestion, but they lack real-time adaptability and do not provide sufficient feedback or insights into the reasons behind crowd density changes. As a result, operators struggle to respond proactively to potential crowd-related incidents. The **AI-Based Real-Time Crowd Density Alert System** aims to overcome this limitation by offering dynamic, real-time crowd density analysis, continuously adjusting its monitoring based on live video feeds. This ensures a timely response to potentially hazardous situations.

B. AI-Assisted Tools in Crowd Management:

AI technologies like **GitHub Copilot** and **Amazon CodeWhisperer** have shown promise in automation, but they fall short when applied to real-time crowd management. These tools focus on code completion rather than providing context-sensitive, domain-specific guidance, which limits their effectiveness in complex, real-world scenarios. Denny et al. [11] highlighted that while AI assistants can boost productivity, they often fail to contribute to deeper understanding unless they explain their suggestions. Similarly, current AI-based crowd management tools do not provide explanatory insights regarding crowd density patterns or actionable feedback for human operators. Our system, on the other hand, integrates a **YOLOv8** model for real-time person detection, combined with **Flask** for live video streaming, ensuring that security personnel receive not just alerts, but also visual context and actionable data to aid their decision-making process.

C. Collaborative Tools in Crowd Management:

Existing collaborative platforms, such as **Replit** and **CodePen**, offer tools for simultaneous editing, but they lack specific AI features that would enhance teamwork in crowd management contexts. These platforms often miss the necessary feedback loops for effective collaboration. In contrast, our system supports real-time collaboration by streaming live video to a web interface via **Flask**, where security teams can monitor the situation together and take prompt action based on the shared video feed and AI-generated insights. This allows for a more coordinated approach to crowd management, similar to pair programming, as highlighted by Williams and Kessler [16].

D. Weaknesses of Existing Crowd Monitoring Strategies:

Many traditional crowd monitoring systems suffer from a lack of contextual understanding and insufficient feedback mechanisms. **Binary feedback** (such as a simple “crowd limit exceeded” notification) provides little insight into the situation, making it difficult for personnel to respond effectively. Moreover, many platforms do not integrate **real-time feedback** from the environment, which could help operators adjust their responses based on immediate data. Our system addresses this by offering real-time crowd density analysis and customizable alert features, ensuring that alerts are not only triggered when thresholds are exceeded, but also accompanied by meaningful insights such as the location of the density hotspot, enabling more precise intervention.

Furthermore, many platforms lack support for different learning styles or types of crowd situations, often relying on one-size-fits-all solutions. The **AI-Based Real-Time Crowd Density Alert System** is designed to support various use cases, from festivals to transportation hubs, ensuring flexibility and scalability in diverse environments.

Relevance to Our Project:

The weaknesses identified in existing systems demonstrate the need for an AI-powered solution that not only detects crowd density but also provides explanatory feedback, real-time collaboration, and insights for proactive intervention. By integrating AI with a **YOLOv8** model, **OpenCV**, and a **Flask-based web interface**, our system offers real-time analysis and contextual information that goes beyond simple alerts. This approach directly addresses the gaps in current crowd management solutions, providing a more effective, adaptive, and intelligent solution for crowd safety.

Our system aims to bridge these gaps by offering real-time, dynamic crowd monitoring that adapts to changing crowd behavior, provides insights into density hotspots, and empowers security teams with actionable feedback, ultimately contributing to safer, more efficient crowd management.

III SYSTEM ARCHITECTURE AND DESIGN

AI-Based Real-Time Crowd Density Alert System is based on a client-server architectural model with a focus on real-time, modular integration, and proactive alerts. The architecture of the system is segregated into three main functional layers: video acquisition, AI-based analysis, and alert & monitoring services. The frontend interface, which is created with Flask, benefits both administrators and operators through the presentation of the actual real-time video stream, present crowd number, and system status via a browser-enabled dashboard. This dashboard contains control settings for defining crowd thresholds, examining system logs, and triggering or terminating monitoring sessions.At the core of the system is the backend processing engine, where live video frames from a USB webcam or smartphone camera (via Iriun or DroidCam) are processed with OpenCV. Every frame is sent to the YOLOv8 model, which is integrated through the Ultralytics library, to detect persons very quickly. Filtered objects are detected for the "person" class, and a count for each frame is generated. The Crowd Monitoring Service, which is a specific Python module, checks if the count that is detected exceeds the set safety threshold.As soon as the threshold is surpassed, two alert mechanisms are activated together: a local sound alert produced through the winsound module (emitting a 15-second beep), and an email alert sent through the SMTP protocol by using Python's smtplib. These alerts are sent to the responsible authorities and act as instant crowd control messages. In the meantime, the labeled video stream with bounding boxes and crowd number is sent through Flask to the live web interface for real-time visualization and decision-making.

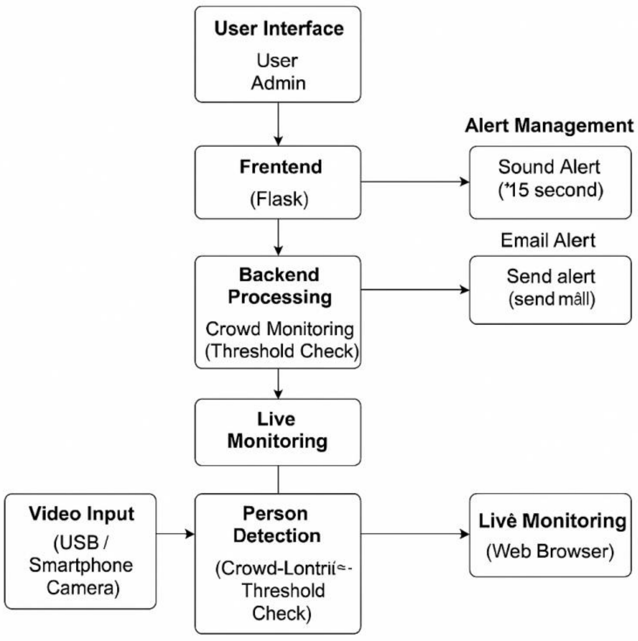

Fig. 2. Overview of the System

Moreover, a light-weight configuration management module enables the administrator to set thresholds, monitor logs, and modify credentials, while a status manager monitors the operation state of the system and logs alert timestamps. The design focuses on isolation of concerns, with each module performing a distinct task like detection, threshold analysis, notification, and video streaming—hence facilitating scalability, simpler maintenance, and future integration with cloud services or edge devices. This structure ensures that the system remains robust and extensible for deployment in high-density environments requiring active monitoring and fast response capabilities.

The main components are as follows:

1.User Interface:A centralized dashboard offers real-time visualization of crowd data, heatmaps, alerts, and control options. It enables security personnel to track areas and react to incidents effectively.
2.Backend Services:In charge of IoT sensor data ingestion (e.g., CCTV, thermal sensors, and motion sensors), storage, and processing. It also manages user authentication, system settings, and incident logging.
3.AI Integration:Crowd density estimation via computer vision.Predictive analytics to predict crowd build-up and recommend control measures. These models are constantly trained and updated with real-world data to enhance accuracy.
4.Real-Time Communication:
Provides immediate alerts and notifications to stakeholders through mobile phones, display boards, or PA systems. It also allows communication between ground staff and the control center.

B. Frontend Implementation

The frontend of the real-time crowd control system is developed using React.js, leveraging a component-based design for enhanced maintainability and scalability. It includes an interactive map interface to visualize crowd density in various zones, updated in real time. Key components include a Live Feed Panel for displaying surveillance footage, a Heatmap Visualization that overlays crowd concentration, and an Alert Console for displaying system-generated warnings. Administrative users can access a control panel to manage zones, view analytics, and trigger emergency protocols. The frontend communicates with the backend through RESTful APIs for data exchange and utilizes WebSocket or Socket.IO for receiving real-time updates. The interface is designed to be fully responsive and optimized for various device types, ensuring usability in both field and control room environments.

C. Backend Implementation

The backend is constructed using Node.js and Express.js to handle API routing, data aggregation, and integration with AI modules. It features a layered architecture including controllers for routing logic, services for business logic, and data access layers for interacting with the database. A core component is the Real-Time Data Aggregator, which collects information from IoT sensors, cameras, and user devices. The backend also includes an AI Decision Engine that analyzes input data to detect abnormal crowd behavior and trigger alerts. Role-based authentication is implemented using JWT tokens to secure access to administrative functionalities. Real-time capabilities are managed through WebSocket/SSE protocols, allowing the system to push live alerts and crowd metrics to connected clients efficiently.

D. Database Schema

The system employs MongoDB to manage its unstructured and semi-structured data. Collections include Zones, which store metadata about monitored areas; CrowdData, which logs real-time density and movement metrics; Users, containing access credentials and roles; and Alerts, which record AI-generated warnings and administrator actions. The database design focuses on performance and scalability, using indexing on frequently queried fields such as zone ID and timestamp. Additionally, embedded documents are used to minimize latency when querying related data such as sensor history within a specific zone.

E. AI Integration

At the core of the system lies an AI module that enables intelligent crowd analysis using computer vision and predictive analytics. Real-time video feeds and sensor data are processed using models trained to detect anomalies such as overcrowding, stampedes, or loitering. The AI module performs two primary tasks: (1) Crowd Behavior Detection—classifying normal vs. abnormal patterns based on density, movement velocity, and formation; and (2) Predictive Modeling—forecasting congestion using historical and real-time data. These models are fine-tuned using domain-specific datasets and continuously improve through feedback loops. Alerts generated by the AI are context-aware, considering environmental factors like time of day, event schedules, and weather to reduce false positives.

F. Security Implications

Security is a critical aspect of the system, particularly because it processes sensitive video feeds and user data. The platform enforces strict access control using JWT-based authentication and role segregation. Data streams are encrypted both in transit and at rest to prevent interception and unauthorized access. The codebase follows secure development lifecycle principles, including input validation, audit logging, and error handling. Additionally, the system uses sandboxed environments to process potentially vulnerable data and employs anomaly detection techniques to flag suspicious access patterns. These measures ensure the integrity, confidentiality, and availability of the system in high-risk, real-time environments.

IV. IMPLEMENTATION

A .Real-Time Data Processing and Evaluation

The system processes live data from surveillance cameras, IoT sensors, and mobile devices to evaluate crowd density and movement in real time. Each data stream is routed through an isolated processing unit to ensure system resilience and maintain data integrity. These units support multiple data types including video, geolocation, and telemetry. The evaluation module analyzes data based on predefined thresholds, such as crowd density limits, abnormal group movement, and stagnation. In response to evaluations, the system dynamically generates visual alerts and real-time feedback to the dashboard, supporting timely decisions by security personnel.

B. Real-time Collaboration

To enable multi-agency coordination and swift incident response, the system employs Socket.IO to facilitate real-time communication. Each monitored zone is associated with a virtual collaboration channel, allowing security teams, event managers, and emergency services to communicate instantly. Key features of the collaboration module include:

1. **Live Incident Sharing**: Users can share photos, videos, and incident reports within the virtual zone.
2. **Presence Tracking**: Team members logged into a specific zone are visible to others, promoting accountability and team awareness.
3. **AI Chat Assistant**: The chat interface integrates an AI-powered assistant capable of answering procedural questions or suggesting response strategies.
4. **Historical Logs**: All communications are stored for audit purposes and post-event analysis.

C. AI-Assisted Learning

Artificial intelligence is central to the system's decision support mechanism. Two core functionalities are provided.

1. **Anomaly Detection**: The AI continuously scans input data for anomalies like stampedes, unauthorized gatherings, or bottlenecks.
2. **Predictive Analytics**: It forecasts potential crowd build-ups using real-time and historical data, enabling proactive intervention.

D. User Interface Design

The user interface is crafted to provide quick situational awareness and simplicity of use under stress. The main design principles are:

1. Unified Control Dashboard: Everything—live feed, maps, alerts, AI recommendations—is consolidated into one view.

2. Real-Time Visual Feedback: Heatmaps and alerts update in real-time to represent the current crowd conditions.

3. Minimal Click Operations: Common operations like broadcasting alerts or changing views take few steps.

4. Collaborative Workflows: Messaging, reporting, and decision support are integrated within the UI to enable effortless task completion.

5. Adaptive Layouts: The interface adapts dynamically depending on user role (admin, on-ground officer, supervisor) and device type.

E. Admin and Content Management

The AI-based real-time crowd control system's administrative module is the operational center for management and content setup. System administrators are able to define geographic areas where surveillance and control are to be applied through a specific admin panel. Each area can be set up with distinct sensor parameters, including camera angles, resolution settings, and data sampling rates, to maximize detection accuracy depending on the environment. The platform constantly checks system health, allowing real-time monitoring of node uptime, sensor status, and data transmission integrity. Admins are able to dynamically set alert sensitivity thresholds based on crowd density trends, projected footfall, and security levels needed for a particular event. This renders the platform extremely responsive to changing conditions like concerts, protests, festivals, or emergency evacuations.Aside from real-time configuration, the admin panel accommodates strong user role and access control. Various administrative levels (e.g., supervisor, analyst, security head) can be assigned customized access to features for ensuring operational security and efficiency. The system further records and stores crowd movement data, enabling administrators to create historical analytics, identify repeating congestion patterns, and determine the effect of crowd control interventions. Event reports, such as timestamps of anomalies, maximum density records, and AI-driven interventions, are available for review and export. The admin panel, in turn, serves as the command center for proactive management to ensure safety and situational awareness in multicultural event.

System Workflow

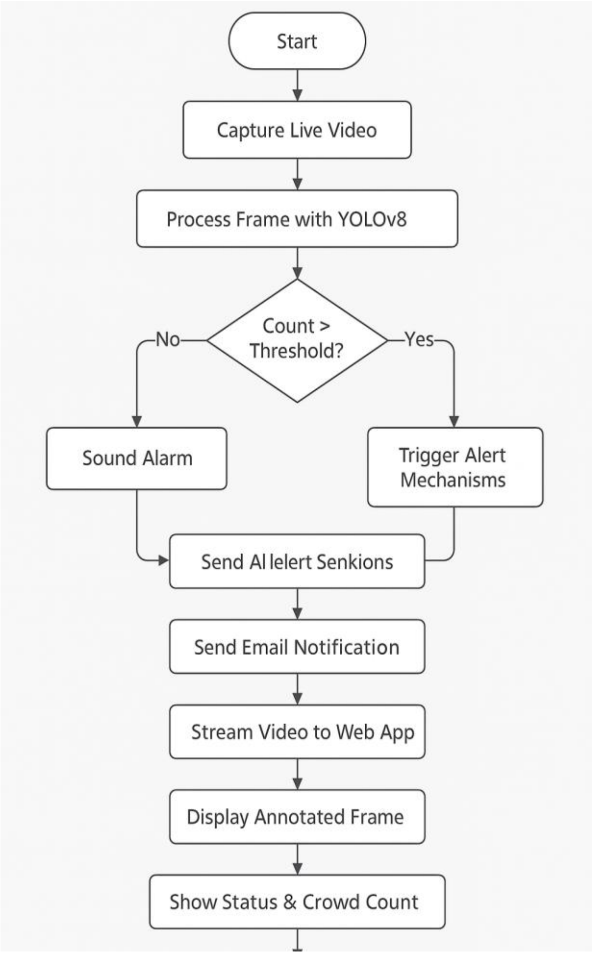

Fig. 3. DFD of the Proposed System

V. WORKING PRINCIPLE

Introduction to System Workflow

The AI-based real-time crowd control system follows a structured algorithm to enable intelligent monitoring, real-time alerts, and crowd behavior analysis. The algorithm is broken down into several steps.

Algorithm

Step 1: Integration of Sensors and Initialization of Zones

Input: Live video streams, crowd density sensors, and environmental inputs.

Process:

1. Authenticate system and initialize specified surveillance zones.

Ingest real-time feeds from CCTV, thermal cameras, or IR sensors.

Specify virtual geofences and zone boundaries for active monitoring.

Initiate continuous crowd flow tracking using AI models.

Output: Zones initialized and sensor feeds turned on for monitoring.

Step 2: Real-Time Crowd Detection and Analysis

Input: Frames of video streams and sensor data.

Process:

1. Apply AI/ML algorithms (YOLO, OpenCV) to identify people and crowd density.

2. .Estimate crowd flow direction, density, and speed.

3. Detect unusual patterns such as stagnation, overload, or panic movement.

4. Activate alerts upon dynamic threshold triggers and heatmap analyses.

Output: Real-time crowd analysis and anomaly alerts.

Step 3: AI-Driven Alert System

Input: Suspicious crowd patterns and risk trends.

Process:

1. Monitor context of crowd movement (e.g., unusual density spike).

2. Trigger contextual warnings through rule-based and predictive AI reasoning.

3. Alert based on threat level and probability of risk outcomes.

4. Notify admin dashboards and security personnel in real-time.

Output: Alerts and recommendations by AI for crowd control actions.

Step 4: Automated Response and Visualization
Input**:** Active alert or threshold breach event.
Process:

1. Imagine the crowd movement in heatmaps and live cam overlays.
2. Advise optimal crowd dispersal routes and intervention tactics.
3. Activate automated announcements or signage via connected systems.
4. Log incident information for post-event analysis. Automated safety interventions and real-time visual monitoring.

Step 5: Admin Control and Historical Analysis

Administrative UI request for data and analytics.

Process:

1. Access dashboard to view system health, sensor status, and alerts.
2. Study timelines of past events and crowd movement patterns.
3. Change zone definitions, alarm levels, or system parameters.
4. Create incident reports for strategy review and compliance.
5. Customizable system control and actionable event intelligence.

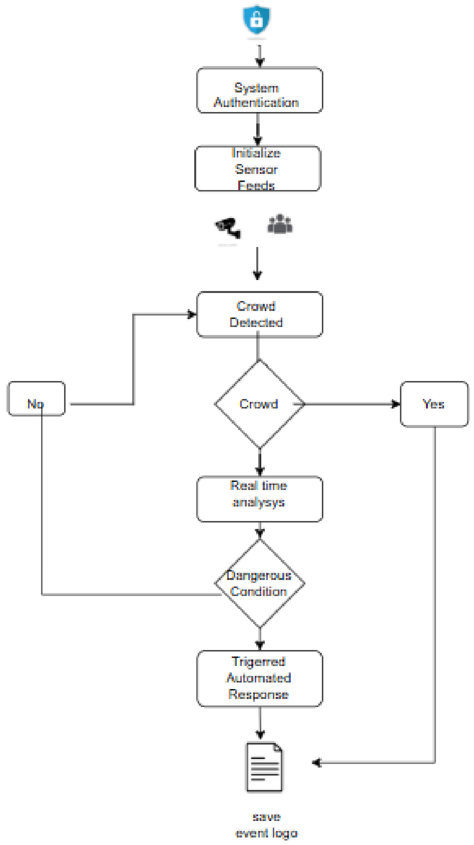

Fig. 4. Algorithm of System

VI. CONCLUSION

In this paper, we have presented the development and deployment of an AI-driven real-time crowd management system adapted to dynamic, high-density public spaces. The solution utilizes advanced artificial intelligence methods, real-time data acquisition, and centralized management features to respond to the new issues of crowd monitoring and assurance of safety. By combining surveillance streams, sensor networks, and AI-driven analytics, the system is able to detect abnormally behaving crowds, predict congestion hotspots, and facilitate timely responses.

The modular design of the system ensures scalability, flexibility, and effortless deployment. The frontend interface provides real-time visualizations and responsive admin features, while the backend handles data ingestion, AI-based analysis, and event-driven decision-making with high efficiency. The system can be deployed in various sites such as sports stadiums, city intersections, festival grounds, transportation hubs, and public events. The use of AI models, particularly in anomaly detection and predictive modeling, allows the platform to forecast crowd movement patterns and generate context-aware alerts independently.

One of the most prominent characteristics of this system is real-time decision support provided by AI-driven engines with live input to classify behavior and offer action plans accordingly. Another functionality of the system is the alert mechanism—governed by adjustable sensitivity limits—so that timely and well-directed interventions become possible.

Such alerts may be directed to on-site authorities, event promoters, or public address systems for optimum reduction of chances of stampede, bottleneck formation, or excessive crowd densities.

Admin console provides full control of system settings including zone definitions, sensor integration, role-based access, and historical report generation. Web-based, centralized dashboard provides decision-making operators with visual analysis, heatmaps, and statistical reports. Historical data are stored in a NoSQL database securely for after-event analysis, pattern detection, and performance audit of implemented crowd control measures.

Security and privacy have also been core concerns in this system's design. Data collected is encrypted and access limited by rigorous authentication procedures. Anonymized video analysis is facilitated by the system and adheres to data protection legislation to enable ethical deployment into public spaces.

The system was tested on simulated scenes with synthetic crowd data and real video streams. The performance was accurate in crowd density estimation, quick event detection, and low response times. The AI models used feedback to learn and update themselves through retraining cycles.

Overall, the proposed AI-driven crowd control system shows an efficient, intelligent, and secure model of managing crowds for mass public gatherings. It reduces reliance on human observation, enhances situational awareness, and provides for faster, data-driven action on crowd-based threats. Possible areas of research for the future involve the deployment of edge computing to reduce latency, the utilization of drone-based mobile surveillance, and the multimodal fusion of data streams such as GPS, RFID, and social media patterns to enhance situational awareness. The system is an enabler to safer, smarter cities and event management facilities

Fig. 5. Login interface of the AI-based real-time crowd control system, providing secure access for authorized personnel to monitor, analyze, and manage crowd dynamics effectively.

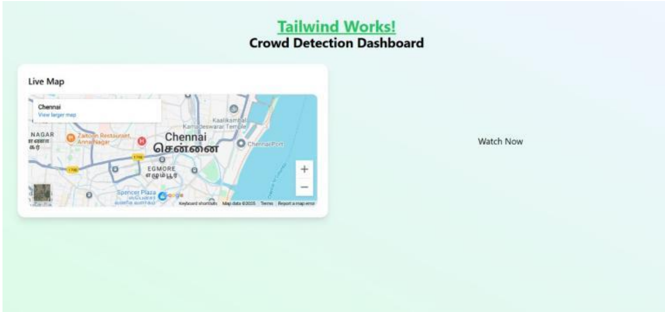

Fig. 6. Real-time crowd detection dashboard showcasing a live map of Chennai, enabling location-based monitoring and instant visualization of crowd density for efficient urban management.

VII. REFERENCES

[1] S. Ali and M. Shah, “A Lagrangian Particle Dynamics Approach for Crowd Flow Segmentation and Stability Analysis,” *IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–6.

[2] D. Helbing and P. Molnár, “Social force model for pedestrian dynamics,” *Physical Review E*, vol. 51, no. 5, pp. 4282–4286, 1995.

[3] M. Rodriguez, I. Laptev, J. Sivic, and J. Audibert, “Data-driven crowd analysis in videos,” *IEEE International Conference on Computer Vision*, 2011, pp. 1235–1242.

[4] T. Xiang and S. Gong, “Video behaviour profiling for anomaly detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 5, pp. 893–908, May 2008.

[5] H. Idrees, I. Saleemi, C. Seibert, and M. Shah, “Multi-source multi-scale counting in extremely dense crowd images,” *IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2547–2554.

[6] W. Ge, R. T. Collins, and R. B. Ruback, “Automatically detecting the mistreatment of people with a video camera,” *International Journal of Computer Vision*, vol. 82, no. 2, pp. 163–181, 2009.

[7] M. Rahmani and M. Shah, “Crowd counting using structural similarities in density maps,” *IEEE International Conference on Computer Vision*, 2015, pp. 2103–2111.

[8] V. Reddy, C. Sanderson, and B. C. Lovell, "Improved anomaly detection in crowded scenes via cell-based analysis of foreground speed, size and texture," *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2011, pp. 55–61.

[9] J. Chen, H. Liang, and L. Hanzo, "Artificial intelligence in intelligent transportation systems: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 5663–5688, Jul. 2022.

[10] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," *IEEE Conference on Computer Vision and Pattern Recognition*, 2005, pp. 886–893.

[11] H. Yao, F. Wu, and X. Zhu, "Deep learning for crowd analysis: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 11, pp. 7780–7798, Nov. 2022.

[12] P. Molchanov, S. Gupta, K. Kim, and K. Pulli, "Multi-sensor system for driver's hand-gesture recognition," *IEEE International Conference on Automatic Face and Gesture Recognition*, 2015, pp. 1–8.

[13] S. Zhan, H. Lin, and Y. Xu, "An edge AI-based real-time crowd monitoring system for smart cities," *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 6489–6497, Apr. 2021.

[14] J. Liu, Y. Li, and T. Mei, "Automatic video synopsis via spatiotemporal optimization," *IEEE Transactions on Multimedia*, vol. 18, no. 12, pp. 2254–2264, Dec. 2016.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.