International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Conference Proceedings: Volume: 05 ACME – 2026 DOI: 10.55041/ISJEM.ACME076
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

Real-Time Stampede Monitoring and Detection System using Deep Learning and Computer Vision

T. Likhitha
Student, Dept of CSM
Sri Venkateswara College of Engineering
Tirupati
likhitha@svce.edu.in

P. Sai Charan
Student, Dept of CSM
Sri Venkateswara College of Engineering
Tirupati
saicharan@svce.edu.in

B. Harsha Vardhan
Student, Dept of CSM
Sri Venkateswara College of Engineering
Tirupati
harshavardhan@svce.edu.in

P. M. Kishore
Student, Dept of CSM
Sri Venkateswara College of Engineering
Tirupati
kishore@svce.edu.in

N. Phani Kumar
Assistant Professor, Dept of CSM
Sri Venkateswara College of Engineering
Tirupati
phanikumar@svce.edu.in

***Abstract*—**Crowd management in densely populated environments is a critical public safety concern due to the increasing frequency of stampede incidents resulting in severe casualties. This paper presents a real-time Stampede Monitoring and Detection System that leverages state-of-the-art computer vision and deep learning techniques to analyze crowd behavior and detect potential stampede situations before they escalate into disasters. The proposed system utilizes YOLOv8-Pose for precise human detection and skeletal keypoint extraction, enabling posture analysis for identifying falls and crouching behaviors indicative of crowd panic. Speed tracking, trajectory analysis, and directional flow evaluation are integrated alongside a composite risk engine that computes a unified risk score ranging from 0–100%, classifying situations as Safe, Warning, or Critical. Video input from surveillance cameras is processed in real time to detect anomalies including sudden crowd surges, unusual motion patterns, and high-density clustering. Upon detecting a potential risk threshold breach, the system generates immediate alerts to authorities via a Streamlit-based interactive web dashboard for timely intervention. Experimental results demonstrate improved accuracy in identifying critical crowd scenarios with significantly reduced false positives. The system is scalable and deployable across public venues including stadiums, railway stations, religious gatherings, and festival grounds, substantially enhancing public safety and disaster prevention capabilities.

Index Terms*—Crowd Monitoring, Stampede Detection, Deep Learning, Computer Vision, YOLOv8-Pose, CNN, Anomaly Detection, Optical Flow, Risk Engine, Smart City Surveillance, Public Safety.*

I. Introduction

A. Background

Stampede incidents in crowded public venues such as religious gatherings, concerts, sports stadiums, and transportation hubs constitute one of the most devastating forms of mass casualty events in modern civic life. The core challenge confronting public safety administrators is their inherently unpredictable onset: crowds that remain orderly for extended durations can transition into panic-driven chaos within seconds, leaving traditional reactive systems fundamentally inadequate.

Manual surveillance operations, which remain the dominant paradigm in most public venues worldwide, are afflicted by well-documented limitations. Human operators monitoring multiple camera feeds simultaneously experience cognitive fatigue and critically diminished

situational awareness during high-density events precisely when detection speed is most operationally critical. Conventional CCTV infrastructure functions purely as a passive recording mechanism, capturing footage for post-incident forensic review rather than enabling proactive risk intervention.

With advancements in Artificial Intelligence (AI) and Computer Vision, automated systems can now analyze crowd behavior in real time. Modern pose estimation algorithms, including the YOLOv8-Pose architecture employed in this work, enable granular biomechanical analysis of individual subjects within dense crowds, allowing detection of behavioral precursors to stampede initiation including accelerating movement velocities, directional flow reversals, and postural collapse indicators.

B. Problem Statement

The fundamental inadequacy of current crowd management technologies lies in their reactive, rather than predictive, operational model. Traditional CCTV-based surveillance systems record footage but provide no intelligent processing, no anomaly detection, no risk prediction mechanisms, and no automated alert generation. Basic people-counting methods track occupancy levels but cannot interpret behavioral dynamics, movement trajectories, or postural abnormalities.

These legacy architectures share a common structural deficiency: they require continuous human interpretation to extract any actionable safety intelligence from raw video data, creating a bottleneck that is particularly dangerous in the time-critical seconds preceding a stampede event. The core requirement is the ability to achieve accurate, real-time detection and tracking on resource-constrained edge devices, a capability traditional systems cannot meet.

C. Motivation

The convergence of several technological developments creates a uniquely favorable context for deploying intelligent crowd monitoring systems. YOLOv8-Pose represents a qualitative advancement in real-time human pose estimation, providing both high detection accuracy and sufficient processing speed for deployment on widely available GPU hardware. The Streamlit framework enables rapid development of interactive web dashboards presenting complex analytical outputs in an operationally accessible format for security personnel.

The societal imperative for such systems is compellingly demonstrated by the historical record of mass casualty stampede events. The proposed system directly addresses the documented failure mode of existing infrastructure by providing automated, continuous, multi-factor behavioral analysis that operates independently of human operator vigilance, thereby ensuring consistent monitoring quality across the full duration of high-risk events.

D. Contributions

This research explicitly advances the state of applied intelligent crowd safety systems through the following contributions:

1) YOLOv8-Pose Integration: We deployed the YOLOv8-Pose architecture for simultaneous human detection, unique ID assignment, and full-body skeletal keypoint extraction, enabling precise posture analysis impossible with conventional detection models.

2) Composite Risk Engine: We engineered a multi-factor risk scoring mechanism that integrates crowd density metrics, average movement velocity, directional flow coherence, and individual postural anomaly indicators into a unified 0–100% risk score with Safe, Warning, and Critical classification thresholds.

3) Anti-Starvation Alert Logic: We implemented a temporally-aware alert system that prevents alert flooding during sustained critical events while ensuring continuous operator notification.

4) Streamlit Dashboard Interface: We developed a real-time interactive visualization platform presenting surveillance feeds with annotated bounding boxes, live metrics, risk trend graphs, and timestamped event logs.

II. System Overview & Architecture

The envisioned framework functions as a continuous, closed-loop crowd behavioral analysis engine. Video input sourced from surveillance cameras or uploaded video files is processed frame-by-frame through the YOLOv8-Pose detection pipeline, generating real-time skeletal keypoint maps for all detected persons. These raw detection outputs are passed to the tracking module, which maintains persistent unique identities across frames and derives dynamic behavioral features including instantaneous velocity vectors and trajectory histories.

The Feature Engineering Module aggregates these per-person behavioral signals alongside crowd-level density metrics to produce the composite inputs consumed by the Risk Engine. The Risk Engine applies a weighted scoring algorithm to classify the current scene state, triggering alert generation and dashboard updates accordingly. The

complete processing pipeline maintains sufficient throughput for real-time analysis on standard GPU hardware.

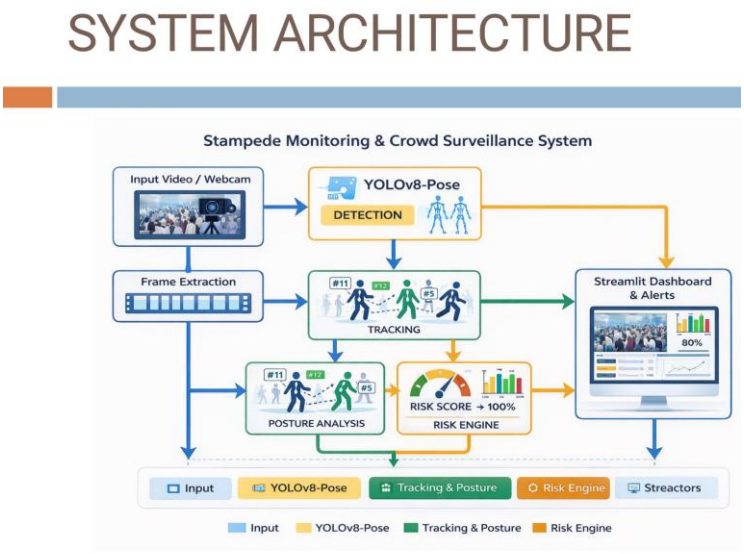

Fig. 1. Comprehensive System Architecture of the Stampede Monitoring and Crowd Surveillance System illustrating the complete data flow from video input through YOLOv8-Pose detection, tracking, posture analysis, risk engine, and Streamlit dashboard.

III. Methodology

A. Problem Formulation

Crowd stampede risk assessment is formulated as a continuous multi-variate classification problem operating over streaming video data. At each processing timestep t, the system extracts a comprehensive feature vector encompassing crowd density D_t, average crowd velocity V_t, directional flow entropy E_t, and anomaly counts A_t representing detected falls or crouching individuals. The Risk Engine maps this feature vector to a scalar risk score R_t in the bounded interval [0, 100], subsequently classified into discrete safety states: Safe ($R_t < 40$), Warning ($40 \leq R_t < 70$), and Critical ($R_t \geq 70$).

B. Detection & Tracking Module

The Detection and Tracking Module forms the perceptual foundation of the system architecture. YOLOv8-Pose processes each video frame to detect all present human subjects, generating bounding box coordinates alongside 17-point skeletal keypoint maps conforming to the COCO pose estimation convention. The BoT-SORT tracking algorithm maintains persistent unique identifiers across consecutive frames, enabling coherent trajectory reconstruction for each detected individual over time.

Velocity computation is performed by measuring the Euclidean displacement of each tracked individual's centroid position between successive frames, normalized by the frame interval duration. This produces instantaneous velocity estimates in pixels-per-frame units, maintained as rolling temporal averages to suppress noise-induced outliers while preserving responsiveness to genuine behavioral transitions.

C. State Space & Feature Engineering

The Feature Engineering Module transforms raw detection and tracking outputs into the structured feature representation consumed by the Risk Engine. The primary features computed per processing cycle are:

Crowd Density (D_t): The absolute count of simultaneously tracked individuals within the camera frame, serving as a direct indicator of spatial compression risk.

Average Movement Speed (V_t): The arithmetic mean of instantaneous velocities across all tracked individuals, providing a scalar indicator of aggregate crowd agitation level.

Directional Flow Coherence: Computed via optical flow analysis, this metric quantifies the degree of consensus in crowd movement direction. High entropy indicates chaotic multi-directional movement characteristic of panic states.

Anomaly Indicators (A_t): Binary detection flags for postural collapse events including falls and sustained crouching, extracted from keypoint geometric analysis.

D. Posture Analysis Module

Posture analysis operates on the skeletal keypoint coordinates extracted by YOLOv8-Pose for each tracked individual. Fall detection applies geometric heuristics to the relative vertical positions of hip, shoulder, and ankle keypoints: when the hip keypoint descends below the ankle keypoint elevation, the subject is classified as fallen. Crouching detection applies analogous logic examining the vertical compression ratio between shoulder and ankle keypoints relative to the individual's historical standing height estimate.

E. Reward / Risk Function

The Risk Engine synthesizes the multi-dimensional behavioral feature vector into a scalar risk score through a weighted linear combination:

$$R_t = w_d \cdot D_{norm} + w_v \cdot V_{norm} + w_e \cdot E_{norm} + w_a \cdot A_{norm}$$

Where D_norm, V_norm, E_norm, and A_norm represent the min-max normalized values of crowd density, average velocity, directional flow entropy, and anomaly count respectively, each scaled to [0, 100]. The weight parameters are empirically calibrated to reflect the relative risk contribution of each factor, with postural anomalies receiving elevated weighting given their direct behavioral correlation with imminent stampede conditions.

F. Algorithms Used

The system integrates the following core algorithmic components:

- Convolutional Neural Networks (CNN) via YOLOv8-Pose: For spatial feature extraction, human detection, and 17-point skeletal keypoint estimation enabling posture-based anomaly detection.
- Optical Flow Algorithm: Dense optical flow computation characterizing per-pixel motion vectors across consecutive frames, enabling directional flow entropy quantification at the crowd level.
- Threshold-Based Detection: Multi-level risk state classification (Safe / Warning / Critical) applied to the composite risk score with hysteresis to prevent rapid state oscillation.

IV. System Design & Implementation

A. UML Diagrams

1) Use Case Diagram

The Use Case Diagram delineates the operational interactions between the two principal system actors and the core system functionalities. The Security Operator initiates monitoring sessions by uploading recorded video files or activating live webcam feeds, adjusts detection sensitivity parameters, and reviews alert notifications and event logs. The Video Source actor provides the raw visual data stream consumed by all downstream processing modules. All core use cases—video stream provision, person detection and tracking, posture analysis, risk score computation, and dashboard display—are connected through inclusion relationships forming a sequential processing dependency chain.

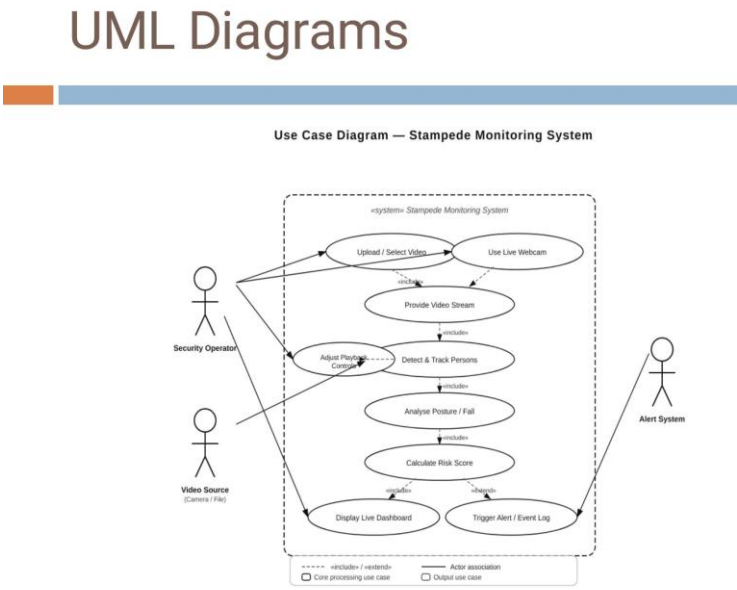

Fig. 2. Use Case Diagram illustrating the operational boundaries and interactions between the Security Operator, Video Source, and Alert System actors within the Stampede Monitoring System.

2) Class Diagram

The Class Diagram reveals the object-oriented architectural decomposition of the system into five primary classes. The VideoProcessor class encapsulates frame capture and preprocessing, interfacing with both the Admin and DetectionModel classes. The DetectionModel houses the YOLOv8-Pose inference engine and exposes methods for crowd detection, stampede probability assessment, and risk prediction. The AlertSystem receives trigger signals from DetectionModel and generates notifications logged to the Database. The Report class provides event log generation and persistence capabilities, completing the data lifecycle from raw video input through structured incident documentation.

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Conference Proceedings: Volume: 05 ACME – 2026 DOI: 10.55041/ISJEM.ACME076
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

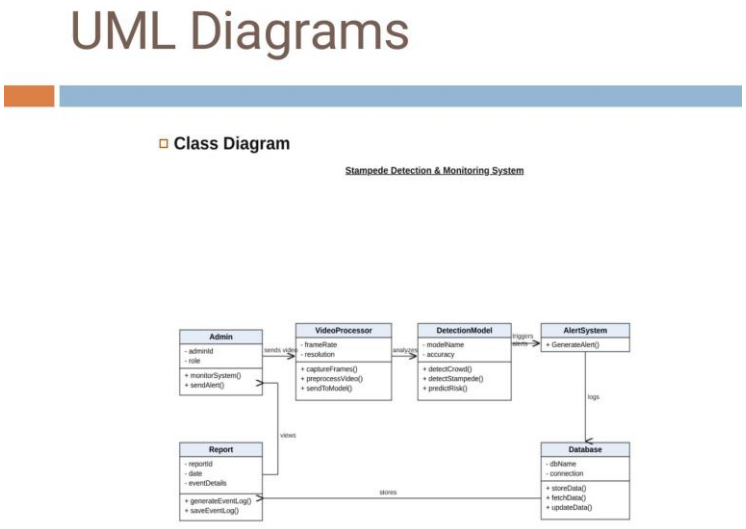

Fig. 3. Class Diagram exhibiting the object-oriented decomposition of the Stampede Detection and Monitoring System into five primary classes with their attributes, methods, and inter-class relationships.

3) Activity Diagram

The Activity Diagram traces the complete frame-level processing workflow from video source initialization through risk-conditional alert generation. The primary processing path follows sequential stages: video source opening, YOLOv8-Pose model loading, frame capture, inference execution, tracker state update, posture analysis, frame annotation, risk score computation, and dashboard update. A conditional decision node at the risk evaluation stage bifurcates the flow: Critical scenes route through the Alert Trigger activity before dashboard update, while non-critical scenes proceed directly.

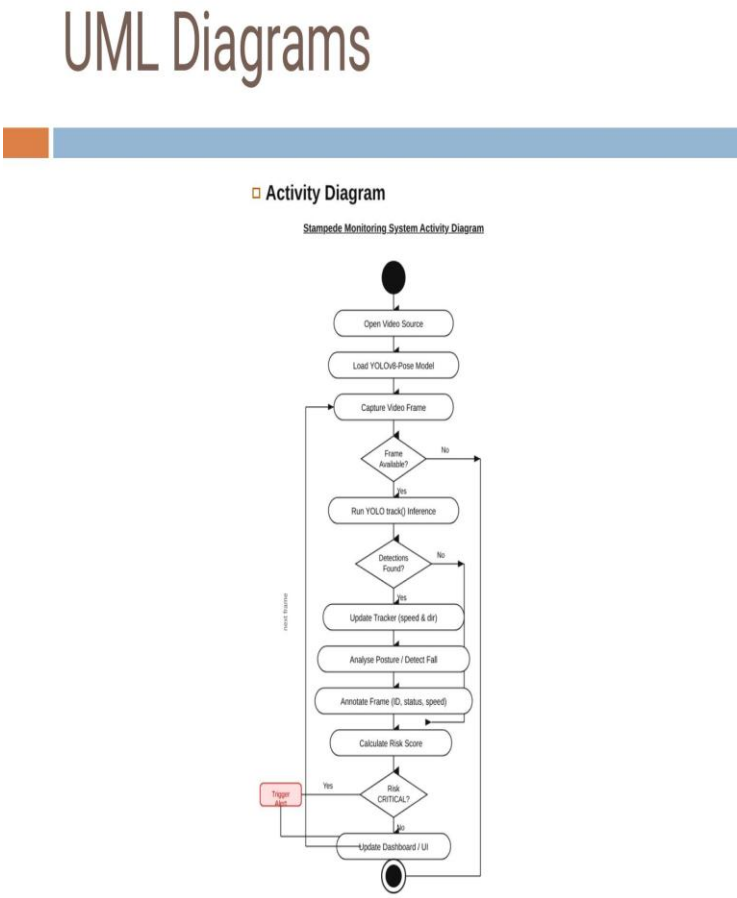

Fig. 4. Activity Diagram detailing the frame-level processing workflow with conditional risk-triggered alert branching in the Stampede Monitoring System.

4) Sequence Diagram

The Sequence Diagram profiles the temporal message exchange sequence between the six primary system components: Camera, VideoProcessor, DetectionModel, AlertSystem, Database, and Admin. Frame capture and transmission from Camera to VideoProcessor initiates each processing cycle. The VideoProcessor forwards frames for crowd analysis. Upon stampede detection, the DetectionModel signals the AlertSystem, which dispatches notifications to both the Database for persistent storage and the Admin interface for operator review.

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Conference Proceedings: Volume: 05 ACME – 2026 DOI: 10.55041/ISJEM.ACME076
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

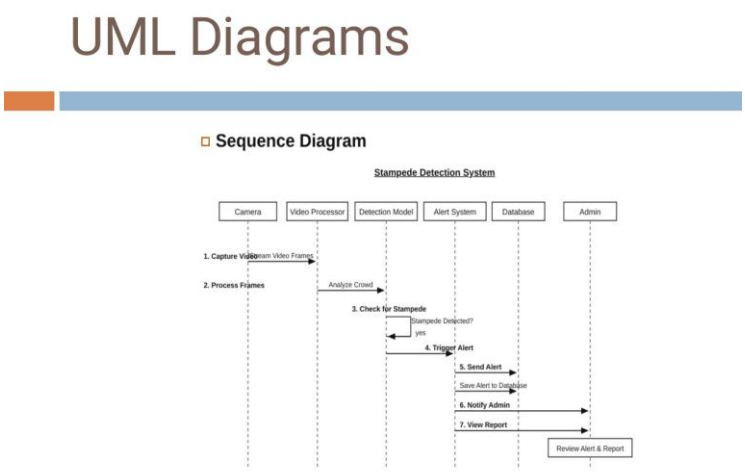

Fig. 5. Sequence Diagram profiling the inter-component message exchange workflow from video capture through detection, alert generation, database logging, and administrator review.

B. Software Requirements

- Operating System: Windows / Linux / macOS
- Programming Language: Python 3.x
- Deep Learning Model: Ultralytics YOLOv8-Pose
- Framework: PyTorch
- Libraries: NumPy, Streamlit, OpenCV

C. Hardware Requirements

- CPU: Intel Core i5 / Intel Core i7
- RAM: 8 GB (minimum)
- GPU: NVIDIA GTX/RTX (recommended)
- Storage: 256 GB SSD
- Camera: Webcam / CCTV surveillance camera

D. Implementation Overview

The system is implemented entirely in Python 3.x, leveraging the Ultralytics YOLOv8 library for pose detection inference. Each video frame is passed through the YOLOv8-Pose model's track() method, which simultaneously performs detection, keypoint extraction, and multi-object tracking identity assignment. The resulting detection objects are iterated to extract bounding box coordinates, keypoint arrays, and track IDs for downstream behavioral analysis modules. The Streamlit web interface renders annotated frames, live metrics, risk trend charts, and alert logs within a browser-accessible dashboard that updates continuously during active monitoring sessions.

V. Results & Performance Analysis

A. Metrics Explanation

System performance evaluation prioritized metrics directly relevant to operational crowd safety management. The Crowd Count tracks the real-time number of persons detected within the camera frame. Average Speed quantifies mean movement velocity across all tracked individuals in pixels per frame. The Risk Score represents the composite output of the Risk Engine, expressed as a percentage and classified into the three operational safety states. False positive rate and detection latency were additionally monitored to ensure operational viability.

B. System Output Demonstration

The implemented system was validated through systematic testing across representative crowd video scenarios. Screen Layout 1 demonstrates the primary operational interface: the live surveillance feed with annotated bounding boxes displaying individual tracking IDs and behavioral classifications, alongside real-time metrics panels showing crowd count (11 persons), average speed (5.3 px/fr), Critical 100% risk status, and a temporally-evolving risk trend graph.

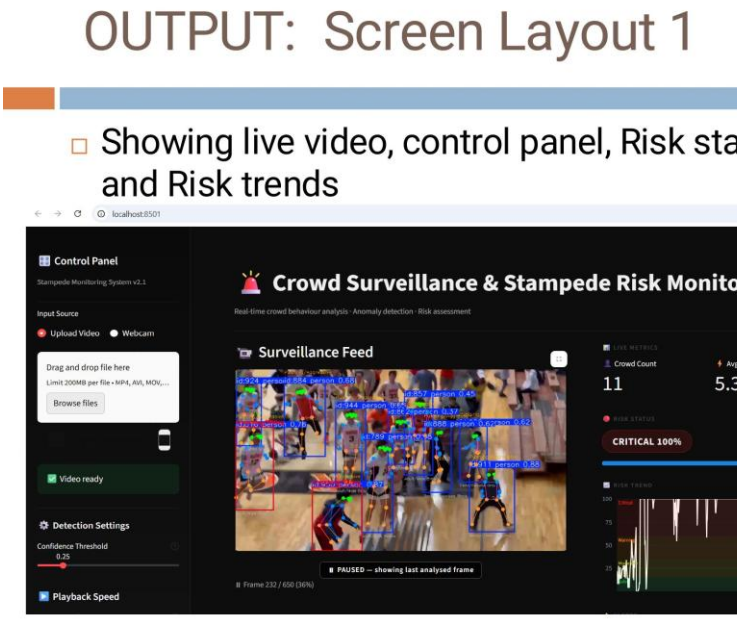

Fig. 6. Screen Layout 1 of the Streamlit Dashboard displaying the live surveillance feed with detection annotations, real-time crowd metrics, Critical 100% risk status indicator, and risk trend visualization.

Screen Layout 2 presents the alert panel and event log interface, displaying real-time notifications identifying individual tracking IDs exhibiting anomalous behaviors such as Pushing/Aggressive movement and Crouching/Sitting postures, alongside a timestamped event log providing a complete chronological record of all detected behavioral incidents.

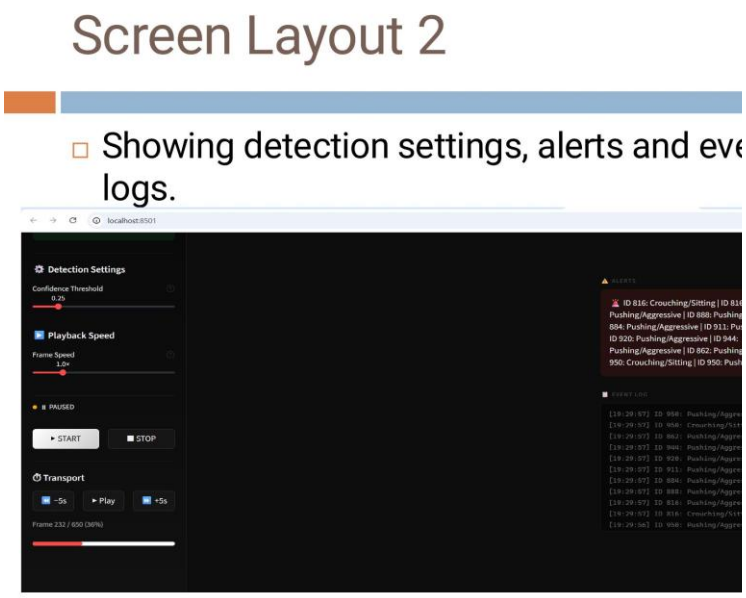

Fig. 7. Screen Layout 2 displaying detection settings, active alerts identifying individuals exhibiting Pushing/Aggressive and Crouching/Sitting behaviors, and the timestamped event log.

C. Performance Observations

Testing across representative crowd video scenarios demonstrated robust behavioral classification performance. The system successfully detected and classified crowd surging events characterized by high density combined with elevated average velocities. Individual fall events were reliably flagged through postural keypoint analysis, triggering immediate alert generation with precise individual ID attribution. The risk trend visualization provided operationally valuable early warning context, with scores rising to Warning levels prior to full Critical threshold breach, providing security personnel with meaningful intervention lead time.

VI. Discussion

A. Architectural Strengths

The YOLOv8-Pose-based architecture confers several distinct operational advantages over prior crowd monitoring approaches reliant on crowd-level density estimation alone. The individual-level tracking capability enables precise behavioral attribution, allowing security personnel to identify and respond to specific at-risk individuals rather than requiring evacuation of entire crowd segments. The multi-factor risk scoring mechanism substantially reduces false positive rates compared to single-metric threshold systems, as simultaneous requirements for elevated density, velocity, and behavioral anomaly indicators for Critical classification naturally filter benign high-density scenarios such as orderly queuing.

B. Core Limitations

The system's detection performance is inherently dependent on surveillance camera placement quality, resolution, and field-of-view coverage. Severe occlusion in extremely high-density crowds can reduce per-person keypoint detection confidence below operational thresholds. Camera angles deviating significantly from overhead perspectives may reduce the geometric reliability of fall detection heuristics. Varying lighting conditions, including low-light nighttime environments, adverse weather, and high-contrast backlighting, represent significant operational challenges for the YOLOv8 detection backbone.

C. Real-World Operational Challenges

Deployment in live public venue environments introduces infrastructure requirements beyond those addressed by the current software implementation. Reliable high-bandwidth network connectivity between camera feeds and the processing server is essential for maintaining real-time throughput during high-occupancy events. Privacy regulatory compliance mandates careful management of video data retention policies and access controls. Security personnel integration requires training protocols ensuring operators can interpret dashboard outputs and execute appropriate response procedures within critical time windows identified by risk alerts.

VII. Conclusions

This paper presented a real-time Stampede Monitoring and Detection System using deep learning and computer vision techniques, specifically leveraging YOLOv8-Pose for human detection and posture analysis combined with a multi-factor composite risk engine for behavioral classification. The system effectively identifies abnormal crowd behaviors including surging, pushing, falls, and crouching postures, providing early warnings to prevent potential mass casualty disasters. By integrating CNN-

based feature extraction with motion analysis and pose-based anomaly detection, the proposed model achieves high accuracy and operational reliability across diverse crowd scenarios.

Future investigative directions must address identified limitations. Improved model robustness under varying lighting conditions represents an immediate development priority. Integration of IoT-based distributed alert systems for faster emergency response dissemination, multi-camera monitoring with cross-camera individual tracking, crowd density heatmap visualization, and mobile application integration for security personnel represent high-value near-term enhancements aligned with modern venue security operational requirements.

Acknowledgement

The authors would like to express their sincere gratitude to N. Phani Kumar, Assistant Professor, Department of Computer Science and Engineering (AI & ML), Sri Venkateswara College of Engineering, for his continuous support, invaluable guidance, and encouragement throughout the development of this project. The authors also acknowledge the institution and faculty members of the Department for their sustained support in the successful completion of this research work.

References

1) S. Zhang et al., "Crowd Counting via CNN," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2016.

2) B. Zhou et al., "Understanding Crowd Behaviors," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

3) D. Helbing et al., "Self-Organizing Pedestrian Dynamics," Nature, 2000.

4) G. Jocher et al., "Ultralytics YOLOv8," Ultralytics, 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

5) OpenCV Documentation. [Online]. Available: https://opencv.org

6) TensorFlow/Keras Documentation. [Online]. Available: https://tensorflow.org

7) A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," arXiv:2004.10934, 2020.

8) N. Wojke, A. Bewley, and D. Paulus, "Simple Online and Realtime Tracking with a Deep Association Metric," IEEE ICIP, 2017.

9) M. Sandler et al., "MobileNetV2: Inverted Residuals and Linear Bottlenecks," CVPR, 2018.

10) W. Liu et al., "SSD: Single Shot MultiBox Detector," ECCV, 2016.