D-Shield: Digital Twin Stampede Hazard Identification and Detection System

Prof. Sadhana Kekan[1], Rushabh Bhalgat[2], Aaryan Bairagi[3], Tanmay Chandgude[4], Kalpesh Paste[5]
[1,2,3,4,5]*Department of Information Technology, Progressive Education Society's Modern College of Engineering, Pune, India*

Abstract*—Large public gatherings are highly vulnerable to catastrophic crowd disasters due to the limitations of traditional, reactive surveillance. Existing deep learning approaches for stampede prediction often rely on opaque classifiers that lack physical interpretability and require heavy computational resources. This paper presents D-SHIELD (Digital Twin-based Stampede Hazard Identification, Estimation and Live Detection), a proactive, edge-deployed framework that fuses real-time video analytics, physical motion modeling, and digital twin technology. Diverging from black-box heuristics, D-SHIELD introduces a novel, physics-aware Motion Metrics Engine. Operating at the network edge on hardware such as the Raspberry Pi, the system utilizes YOLOv8 to extract pose landmarks and computes mathematically defined, viewpoint-agnostic signals: motion impulse (I), orientation flatness ratio (F), and temporal post-impact stillness (S). These metrics are fused using optimized weighting parameters (λ) into a continuous probabilistic FallScore to accurately detect individual collapses when a predefined threshold (τ) is exceeded. For macro-level crowd monitoring, a Multi-Grid Decision Engine partitions the environment, computing an Instability Index (Ψ) by fusing local spatial density (ρ) and velocity variance ($\sigma v2$) to identify emerging stampede precursors. Extracted analytics are transmitted via lightweight Message Queuing Telemetry Transport (MQTT) protocols to an Eclipse Ditto Digital Twin, maintaining a live three-dimensional (3D) virtual replica for real-time situational awareness and predictive forecasting. By combining deterministic physical signal processing with distributed Internet of Things (IoT) messaging, D-SHIELD establishes a scientifically defensible, scalable, and proactive paradigm for crowd safety management.

***Index Terms*—Crowd monitoring, Digital twin, Edge computing, Physics-aware detection, Stampede prediction, YOLOv8.**

I. INTRODUCTION

The unprecedented growth of urban populations and the increasing frequency of large-scale public gatherings, such as religious festivals, concerts, and sporting events, have elevated crowd safety to a critical global concern. Historically, inadequate crowd management has resulted in catastrophic disasters, stampedes, and mass panic incidents that cause devastating loss of life. Traditional crowd management approaches have predominantly relied on reactive measures, such as deploying security personnel after observing dangerous conditions through closed-circuit television (CCTV) systems or enforcing static capacity limits. These conventional methods suffer from fundamental limitations: human operators monitoring multiple camera feeds are prone to attention fatigue, cognitive overload, and subjective decision-making, which often lead to delayed interventions. Furthermore, existing AI-based surveillance solutions frequently rely on opaque, computationally heavy black-box machine learning classifiers that lack physical interpretability and suffer from high latency due to their dependence on cloud computing architectures. By the time visible signs of danger—such as crowd crushing or falling individuals—emerge, the window for effective intervention has typically already closed.

To address these critical gaps and bridge the research-to-deployment divide, this paper introduces D-SHIELD (Digital Twin-based Stampede Hazard Identification, Estimation and Live Detection). D-SHIELD is a proactive, edge-deployed framework that shifts the paradigm of crowd safety from reactive response to anticipatory prevention. Operating entirely at the network edge on lightweight embedded hardware (such as the Raspberry Pi 4), the system

ensures sub-second latency and preserves public privacy, as no raw video footage is ever transmitted to the cloud. Diverging from traditional black-box classification heuristics, D-SHIELD introduces a novel, physics-aware Motion Metrics Engine. The system utilizes the YOLOv8 object detection model to extract skeletal pose landmarks in real-time, translating these coordinates into mathematically deterministic, viewpoint-agnostic signals: motion impulse, orientation flatness ratio, and post-impact stillness. These dynamic signals are fused into a continuous probabilistic FallScore to accurately identify individual physical collapses without relying on pre-trained fall datasets.

II. PROBLEM STATEMENT

Despite the increasing frequency of large-scale public gatherings, contemporary crowd management systems remain fundamentally reactive, frequently detecting hazardous conditions only after they have fully manifested into crowd crushes or stampedes. By the time visible signs of danger—such as blocked pathways, localized congestion, or falling individuals—become apparent, the critical 30 to 60-second window for effective intervention has typically already closed. This reactive paradigm results in devastating consequences, including catastrophic loss of life, severe long-term psychological trauma for survivors, and massive economic and legal liabilities for event organizers.

The primary limitations of existing crowd monitoring and management approaches can be categorized into three critical areas:

1. Limitations of Human-Dependent Surveillance and Static Thresholds

Traditional crowd management relies predominantly on human operators monitoring multiple closed-circuit television (CCTV) feeds. This method is highly susceptible to cognitive overload and attention fatigue; research indicates that human vigilance in monitoring tasks decreases by 50% after merely 20 to 30 minutes of continuous observation. Furthermore, manual crowd assessments provide only aggregate occupancy numbers and lack crucial spatial distribution data. Consequently, static capacity limits fail to account for localized overcrowding, allowing deadly crush conditions to develop in specific zones even when the overall venue remains well below its theoretical capacity.

2. Inadequacies in Current AI and Cloud Infrastructures

While advanced deep learning models have been applied to crowd analytics, practical deployment faces severe technical and architectural barriers. Existing high-end systems typically utilize cloud-dependent architectures that require heavy video data transmission, introducing unacceptable latencies of 2 to 5 seconds—a delay that is detrimental in fast-moving stampede scenarios. Moreover, current machine learning solutions frequently rely on opaque, "black-box" classifiers that lack deterministic, physics-based interpretability. Traditional systems also perform global scene analysis, treating the entire monitored area as a single entity, which ignores spatial heterogeneity, fails to model cross-zone risk propagation, and delays the detection of localized, early-stage instability.

3. Absence of Predictive Intelligence and Intervention Simulation

Existing systems merely report current conditions and provide no forward-looking intelligence to forecast crowd states 2 to 5 minutes into the future. Operators lack the analytical tools to predict if a specific sector will reach a dangerous density based on current inflow rates, or to test intervention strategies—such as closing gates or redirecting crowds—before implementation. Without multi-agent simulation or digital twin forecasting to validate "what-if" scenarios, emergency responses are based purely on intuition rather than data-driven analysis, which can sometimes inadvertently worsen bottleneck conditions.

Consequently, there is an urgent and critical need to bridge the research-to-deployment gap by developing an automated, edge-deployed, and physics-aware framework capable of performing localized risk forecasting, enabling venue operators to preemptively intervene before crowd instability escalates into a disaster.

III. OBJECTIVES

The primary goal of this research is to bridge the gap between reactive crowd surveillance and proactive

disaster prevention. To achieve a scalable, mathematically grounded, and real-time crowd safety management system, this study outlines the following core objectives:

- Develop a Physics-Aware Edge Analytics Pipeline:

To implement a real-time, edge-deployed computer vision architecture (utilizing hardware such as the Raspberry Pi 4 and the YOLOv8 model) that extracts skeletal pose landmarks at high frame rates. The objective is to replace opaque, black-box machine learning classifiers with mathematically deterministic, viewpoint-agnostic physical signals—specifically motion impulse, orientation flatness ratio, and post-impact stillness—to calculate a probabilistic "FallScore" for accurate individual collapse detection.

- Design a Multi-Grid Zone Decision Engine:

To overcome the limitations of global scene analysis by partitioning the monitored venue into a structured multi-grid system (e.g., creating 60 independent spatial zones). This engine will calculate localized spatial density and velocity variance to generate a continuous Instability Index (Ψ), enabling the early identification of localized stampede precursors and the modeling of dynamic cross-zone risk propagation.

- Establish a Low-Latency Digital Twin Framework:

To construct a highly scalable, cloud-based digital twin using the Eclipse Ditto platform, integrated with lightweight Message Queuing Telemetry Transport (MQTT) messaging protocols. This middleware framework aims to maintain a continuously synchronized, 3D virtual replica of the physical venue with sub-second latency, ensuring secure and immediate reflection of crowd counts, motion entropy, and alert statuses.

- Enable Predictive Forecasting and Simulation-Based Interventions:

To develop a predictive analytics and multi-agent disaster shielding simulation engine capable of forecasting crowd densities 1 to 5 minutes into the future. This objective focuses on allowing safety operators to evaluate "what-if" intervention strategies—such as gate closures, route redirections, and evacuation behaviors—providing actionable intelligence before physical implementation.

- Calibrate Detection Thresholds via Data-Driven Optimization:

To scientifically validate the system's accuracy by treating the mathematical weighting parameters (λ) of the FallScore as an optimization problem. The objective is to utilize a constrained grid search methodology to maximize the F1-score across simulated collapse, jumping, and non-collapse sequences, ensuring minimal false positive rates while maintaining high sensitivity to genuine hazards.

IV. LITERATURE SURVEY

Effective crowd management is highly complex, particularly in high-density environments where sudden surges can quickly escalate into catastrophic disasters. A review of the existing literature reveals a progressive shift from traditional, human-dependent surveillance toward automated systems integrating Computer Vision (CV), behavioral modeling, and Digital Twin (DT) technologies.

A. Deep Learning for Crowd Density and Detection

The widespread availability of closed-circuit television (CCTV) cameras has made computer vision a foundational method for crowd monitoring. Early crowd counting techniques relied on detection-based approaches that struggled with heavy occlusions in dense environments. Recently, Deep Learning (DL) and Convolutional Neural Networks (CNNs) have emerged as highly effective alternatives. One-stage object detectors such as YOLO and SSD, alongside two-stage detectors like Faster R-CNN, are heavily utilized for automated human detection. For extreme density estimation, researchers have introduced purely point-based frameworks like P2PNet, as well as models utilizing dilated convolutions, such as CSRNet, which extract spatial features without losing resolution.

In the context of disaster environments, Shukla and Shukla developed a framework integrating Unmanned Aerial Vehicles (UAVs) with CSRNet to accurately classify disaster zones (using the AIDER dataset) and estimate high-density population clusters. Similarly, Kaka Khel et al. demonstrated the efficacy of a hybridized YOLOv4 model for the real-time estimation of crowd counts, pedestrian speed, and directional flow. Despite these advancements, highly accurate deep learning models typically demand

significant computational resources, limiting their viability for ultra-low-latency, edge-deployed environments.

B. Modeling Crowd Dynamics and Stampede Precursors

Accurately predicting crowd disasters requires robust models that account for both physical interactions and psychological phenomena, such as herding and panic jamming. Researchers frequently employ microscopic agent-based models, such as the Social Force Model (SFM), which simulates pedestrian movement based on driving intentions and avoidance forces.

To transition from simulation to real-world prediction, recent studies have explored advanced algorithmic monitoring. For instance, Sun et al. proposed a stampede alert clustering algorithmic system based on Tiny-Scale Strengthened Deformable DETR to predict stampedes in crowded settings. While these algorithms provide powerful detection capabilities, they often function as opaque, "black-box" classifiers. They rely on learned visual patterns rather than mathematically deterministic, physics-based dynamic signals, making them difficult to audit in safety-critical applications.

C. Digital Twins and Predictive Simulation

The paradigm of the Digital Twin—a dynamic, real-time virtual replica of a physical entity—is revolutionizing data-driven crowd management by enabling continuous state monitoring and predictive "what-if" simulations. Shah et al. proposed a digital twin system specifically tailored for real-time crowd management in transportation hubs, utilizing IoT and edge analytics to reflect physical environments.

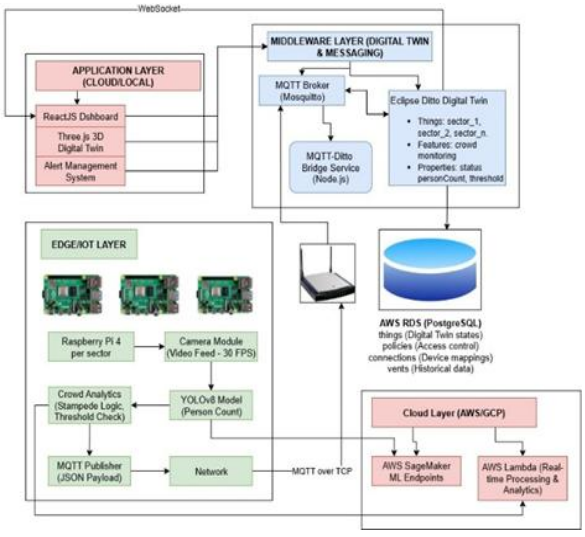
To enable forward-looking intelligence, Krishnakumari et al. introduced the "Crowd Safety Manager," an integrated 3D Digital Twin that utilizes the Bowtie model for risk assessment and the XGBoost machine learning framework to generate multi-day crowd forecasts. Similarly, Alba et al. developed the "Artax" framework, a Digital Twin designed to predict human crowd motion by tracking real-time smartphone locations and utilizing simulators (such as The ONE and Perses) to model dangerous scenarios. In the realm of interactive planning, the "CrowdCollab" system utilizes tabletop tangible user interfaces combined with handheld displays to provide multi-disciplinary teams with both top-down and first-person perspectives for evaluating simulated crowd dynamics.

D. Research Gap

While modern computer vision excels at macro-level density estimation and Digital Twins offer powerful forecasting platforms, a critical operational gap remains. Existing systems often rely heavily on cloud-processing architectures that introduce unacceptable latency, or they depend on machine learning heuristics that lack physical interpretability. D-SHIELD bridges this divide by proposing a patent-defensible, edge-deployed framework that extracts mathematically transparent, physics-based kinetic signals (motion impulse, orientation flatness) directly at the camera node, coupling deterministic early-warning detection with sub-second Digital Twin synchronization.

V. PROPOSED SYSTEM ARCHITECTURE

The D-SHIELD architecture is purposefully designed to bridge the gap between computationally heavy cloud-based machine learning and the ultra-low-latency requirements of real-time disaster prevention. To ensure data privacy and guarantee sub-second response times, no raw video footage is ever transmitted to the cloud. Instead, the core pipeline is organized into three vertically stacked edge-processing modules—the Perception Layer, the Reasoning Layer, and the Communication Layer—supported by a robust Cloud Middleware and Application Layer.



A. Edge Perception Layer

Operating directly on edge hardware (such as the Raspberry Pi 4), the Perception Layer ingests live video streams and serves as the system's "eyes". Rather than utilizing opaque deep learning classifiers to determine safety states, this layer employs a lightweight on-device inference engine (such as YOLOv8) to extract *n* skeletal landmark coordinates for every visible individual in the frame. Concurrently, a Spatial Density Engine estimates a continuous occupancy map. This layer effectively converts raw unstructured pixel data into a clean, mathematically defined representation of crowd coordinates, completely decoupling visual perception from safety policy logic.

B. Edge Reasoning Layer

The Reasoning Layer is the analytical brain of the system, applying physics-aware mathematical modeling to the structured coordinates provided by the Perception Layer. It is subdivided into two primary evaluation tracks:

- Motion Metrics Engine:

This module isolates individual physical collapses by temporally smoothing the extracted landmark coordinates and computing deterministic signals: motion impulse (sudden acceleration), flatness ratio (bounding box geometry indicating a horizontal posture), and post-impact stillness. These signals are normalized and fused into a continuous probabilistic "FallScore".

- Multi-Grid Decision Engine:

To monitor macro-level crowd instability, the monitored space is partitioned into a structured matrix (e.g., a 6×10 grid yielding 60 independent zones). For each zone, the system calculates an Exponential Moving Average (EMA) of spatial density alongside the local velocity variance. These features are fused into a localized Instability Index (Ψ). The Decision Engine then evaluates this index against calibrated thresholds ($T1$, $T2$) to classify each zone's state as Normal, Instability, or Stampede. A cross-zone propagation model (θ) is also applied to upgrade the risk status of adjacent zones if heavy crowd outflow is detected.

C. Communication and Digital Twin Middleware Layer

To guarantee resilient alert delivery even during transient network degradation, outward-facing data transfer is handled exclusively by the Communication Layer via the Message Queuing Telemetry Transport (MQTT) protocol. The Event Publisher module serializes risk classifications, individual counts, and triggering metric values into lightweight JSON payloads and publishes them to an MQTT broker (e.g., Mosquitto). In the cloud middleware, a dedicated Node.js MQTT-Ditto Bridge service subscribes to these anomaly topics. Upon receiving a payload, the bridge invokes REST APIs to instantly update the properties of the corresponding spatial "Things" maintained within the Eclipse Ditto Digital Twin platform. This establishes a continuously synchronized, live virtual replica of the physical venue.

D. Application and Visualization Layer

The top tier of the architecture comprises a cross-platform Command Center dashboard built with Electron and React. Rather than polling a database, the dashboard establishes a bidirectional WebSocket connection directly with the Eclipse Ditto platform. When the Digital Twin state changes, the dashboard instantly receives pushed updates, triggering a WebGL/Three.js scene re-render. This provides safety operators with a live 3D visualization of the venue, overlaying real-time, color-coded crowd heatmaps (e.g., Green = Safe, Yellow = Warning, Red = Critical) and an automated alert management panel. By completely separating the edge perception pipeline from the cloud visualization layer, D-SHIELD maintains high scalability and sub-second end-to-end event reporting

VI. SYSTEM WORKFLOW

The end-to-end execution flow of the D-SHIELD architecture is designed as a continuous, deterministic loop operating primarily at the network edge to guarantee sub-second response times. The per-frame processing cycle avoids opaque machine learning classification and instead strictly follows a sequential pipeline from raw optical input to 3D visualization. The complete system workflow is executed in the following steps:

1. Video Ingestion and Edge Perception
The processing cycle is initiated by the Raspberry Pi camera module, which captures a live video frame. This raw frame is ingested by the Edge Perception Layer, where a lightweight YOLOv8 model isolates human subjects. Subsequently, the Pose Landmark Extractor derives specific skeletal coordinates (such as shoulders and hips) for every visible individual in the frame.

2. Physics-Based Signal Extraction
Instead of relying on deep learning classifiers to detect falls, the system treats each tracked skeleton as a physics particle. The extracted landmark coordinates are temporally smoothed and passed into the Motion Metrics Engine. This engine computes deterministic physical signals for each individual, including centroid velocity, motion impulse (to detect sudden acceleration spikes), flatness ratio (to detect posture collapse), and post-impact stillness.

3. Multi-Grid Spatial Mapping and Density Smoothing
Concurrently, the system performs macro-level crowd tracking. The extracted agent centroids are mathematically mapped to specific sectors within a calibrated spatial matrix (e.g., a 6x10 grid yielding 60 independent zones). To prevent the system from reacting to instantaneous frame-level noise, the Spatial Density Engine applies an Exponential Moving Average (EMA) to the localized occupancy data, outputting a highly stable crowd density metric alongside the local velocity variance.

4. Threat Evaluation via Decision Engine
Once the physical signals and spatial metrics are computed, they are routed to the Decision Engine—the central analytical brain of the system.

- Individual Collapse:

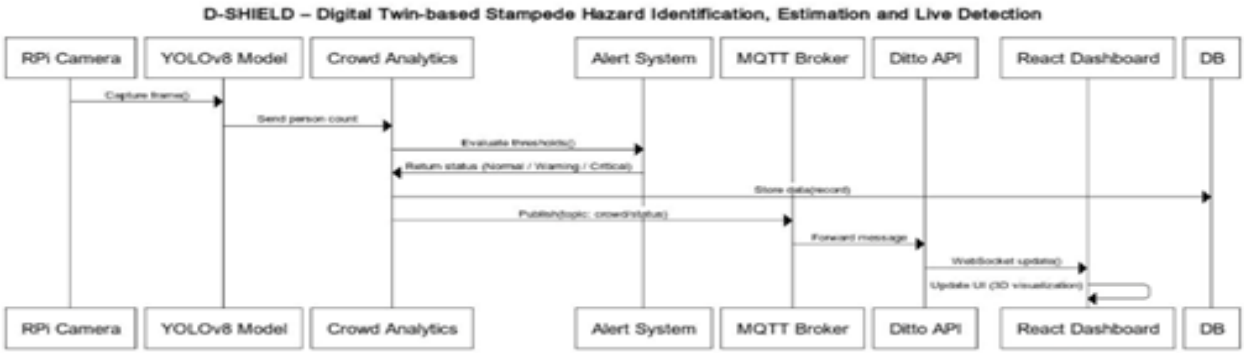
The engine fuses the normalized impulse, flatness, and stillness metrics into a continuous probabilistic FallScore. If this score exceeds a predefined threshold (τ), a fall event is confirmed.

- Crowd Instability:

The engine fuses the EMA-smoothed spatial density and velocity variance to compute an Instability Index (Ψ) for each grid zone. The zone is classified as Normal, Instability, or Stampede based on calibrated thresholds ($T1$, $T2$). The engine also applies a cross-zone propagation model to dynamically upgrade the risk status of adjacent zones if heavy crowd outflow is detected.

5. MQTT Telemetry and Digital Twin Synchronization
If the Decision Engine detects a hazard, it does not send raw video data. Instead, it generates a structured, lightweight JSON alert payload containing the sector ID, risk level, and the triggering metrics. The MQTT Publisher transmits this payload over TCP/IP to a Mosquitto broker. In the cloud middleware, a Node.js MQTT-Ditto Bridge service subscribes to these anomalies, instantly invoking REST APIs to update the properties of the corresponding spatial "Thing" within the Eclipse Ditto Digital Twin platform.

6. Real-Time 3D Visualization and Data Persistence
In the final step, the Eclipse Ditto platform detects the state change in the digital twin and pushes an update via WebSockets to the cross-platform Electron and React Dashboard. This push notification triggers a WebGL/Three.js scene re-render, providing the security operator with an instantly updated 3D venue map featuring color-coded crowd heatmaps and active alert panels. Simultaneously, the alert payload and event data are persisted to a cloud database (AWS RDS or MongoDB) to maintain an audit trail for post-incident analysis and future predictive modeling.



VII. IMPLEMENTATION SETUP

The implementation of the D-SHIELD system requires a highly optimized deployment strategy to bridge edge computing, Internet of Things (IoT) messaging, and cloud-based 3D visualization. The setup is divided across edge hardware configuration, modular software execution, and middleware integration to ensure sub-second latency and mathematical determinism.

A. Edge Hardware and Operating Environment

The edge perception and reasoning layers are deployed on a Raspberry Pi 4 (equipped with 4GB or 8GB of RAM) connected to a standard Raspberry Pi Camera Module or IP camera. To ensure maximum performance with minimal overhead, the edge device operates on the lightweight Raspberry Pi OS Lite (64-bit), specifically built on Debian GNU/Linux 13 ("trixie").

B. Software Stack and Virtual Environments

To manage system stability and prevent dependency conflicts, the edge node employs a dual-Python architectural setup. System-level operations, such as camera interfacing and MQTT broker management, run on the native Python 3.13 environment. Concurrently, the machine learning and computer vision pipelines are isolated within a dedicated Python 3.10 virtual environment (dshield-ml). This isolated environment hosts critical processing libraries, including opencv-python for frame manipulation, mediapipe for pose extraction, tflite-runtime for potential headcount models, and paho-mqtt for telemetry transmission.

C. Modular Edge Runtime and Scheduling

To maintain real-time processing speeds (≥ 5-10 FPS) and avoid thermal throttling on the Raspberry Pi, the system completely avoids monolithic execution. Instead, the architecture is implemented as a highly modular pipeline managed by an orchestrator (edge_runtime.py):

- Perception Layer (pose_pipeline.py & landmark_mapper.py):

The system utilizes MediaPipe to extract 33 holistic skeletal landmarks to ensure viewpoint-agnostic capture. Because the downstream physics engine is optimized for specific dynamic tracking, a mapping module converts these 33 landmarks into a targeted 17-point coordinate system, effectively removing unreliable lower-body keypoints heavily prone to occlusion in top-view crowd scenarios.

- Reasoning Layer (pose_analyzer.py & stampede_detector.py):

The structured 17-point skeletons are passed into individual analyzers that calculate the deterministic physical signals (motion impulse, flatness ratio, and temporal stillness). Concurrently, a spatial density engine applies an Exponential Moving Average (EMA) to filter out instantaneous frame noise, generating a stabilized density metric for the decision engine.

- Scheduler Loop:

To prevent CPU saturation, edge_runtime.py implements a task scheduler. While pose detection runs continuously on every frame, heavier macro-level tasks (such as crowd counting and disaster classification) are scheduled to execute intermittently (e.g., every 10 or 20 frames).

D. IoT Messaging and Cloud Middleware

All external communication is handled via the Message Queuing Telemetry Transport (MQTT) protocol to guarantee delivery under transient network conditions. An Eclipse Mosquitto MQTT broker is installed locally or on a cloud virtual machine. The broker is strictly configured (allow_anonymous false) to require encrypted password authentication, ensuring secure telemetry transmission.

When the decision_engine.py confirms a crowd hazard, an event_publisher.py script packages the anomaly classification, physical metric values, and confidence scores into a lightweight JSON payload and publishes it to the broker. A cloud-hosted Node.js service (the MQTT-Ditto Bridge) continuously subscribes to these topics. Upon receiving a payload, the bridge translates the data into HTTP REST API calls to update the corresponding spatial "Thing" properties within the Eclipse Ditto Digital Twin framework.

E. Application Frontend and Visualization

The top tier of the implementation is the Command Center, developed as a cross-platform desktop application using the Electron framework combined

with React.js. The frontend application establishes a bidirectional WebSocket connection directly with the Eclipse Ditto platform. When Ditto registers a state change from the edge node, it pushes the update to the React application, which subsequently triggers a WebGL/Three.js scene re-render. This dynamically updates the 3D venue map with color-coded heatmaps and active alert panels. Simultaneously, all generated alert payloads are persisted into a MongoDB or AWS RDS (PostgreSQL) database to facilitate historical logging, system auditing, and future predictive model tuning.

VIII. ADVANTAGES OF PROPOSED SOLUTION

The D-SHIELD architecture introduces several critical advantages over traditional closed-circuit television (CCTV) monitoring and conventional machine learning systems, establishing a scientifically grounded and highly scalable approach to crowd safety:

A. Proactive Hazard Prevention

Unlike traditional surveillance systems that rely on human operators to react to disasters after they visibly escalate, D-SHIELD shifts the safety paradigm to proactive prevention. By identifying early stampede precursors and forecasting crowd densities 1 to 5 minutes into the future, the system provides security operators with the crucial lead time required to intervene before hazardous conditions become catastrophic.

B. Interpretable and Physics-Aware Logic

A major limitation of existing artificial intelligence solutions is their reliance on opaque, black-box deep learning classifiers that are difficult to audit. D-SHIELD overcomes this by employing mathematically deterministic, physics-based signals—specifically motion impulse, geometric flatness ratio, and post-impact stillness. This ensures that every alert is scientifically defensible, physically interpretable, and highly patent-defensible.

C. Ultra-Low Latency via Edge Computing

By deploying the perception and reasoning layers directly on lightweight edge hardware (such as the Raspberry Pi 4), the system eliminates the latency typically associated with transmitting heavy video data to cloud servers. This edge-first execution guarantees sub-second event reporting, which is essential for surviving fast-moving crowd crushes where every second dictates the outcome.

D. Strict Privacy Preservation

To adhere to public surveillance and data privacy regulations, D-SHIELD ensures that no raw or sensitive video footage ever leaves the edge device. Only structured, lightweight JSON telemetry payloads containing localized risk classifications and anonymized mathematical metrics are published via the MQTT protocol to the cloud middleware.

E. Viewpoint-Agnostic Detection

The dynamic kinetic signals utilized within the Motion Metrics Engine are designed entirely around relative landmark geometry and spatial displacement. Consequently, the system functions robustly across both top-view and side-view camera angles without requiring pre-trained fall datasets, complex perspective scaling, or strict camera calibrations.

F. Localized Multi-Grid Spatial Analysis

Traditional crowd management systems often perform global scene analysis, which obscures early localized hazards and ignores spatial heterogeneity. D-SHIELD's Multi-Grid Decision Engine partitions the monitored venue into an independent spatial matrix (e.g., 60 isolated zones). By analyzing localized density and velocity variance, the system pinpoints the exact coordinates of emerging instability and actively models cross-zone hazard propagation.

G. Predictive Digital Twin Simulation

The integration of the Eclipse Ditto Digital Twin enables not just real-time 3D visualization, but forward-looking intelligence. Operators can utilize the multi-agent Disaster Shielding module to simulate "what-if" intervention strategies—such as gate closures or crowd redirections—evaluating quantitative risk metrics like evacuation times and bottleneck locations before physically executing the response.

IX. LIMITATIONS AND FUTURE SCOPE

While the D-SHIELD architecture provides a robust, physics-aware, and edge-deployable solution for

proactive crowd management, there are inherent constraints in its current implementation, along with several avenues for future enhancement.

A. Limitations

- Occlusion in High-Density Environments:

In extremely dense crowd scenarios, computer vision algorithms, including YOLOv8 and skeletal landmark extractors, suffer from severe occlusion. While D-SHIELD utilizes spatial density mapping to mitigate this, precise individual-level tracking and counting may underreport the true crowd size when lines of sight are heavily obstructed.

- Edge Hardware Compute Bottlenecks:

The system relies on the Raspberry Pi 4 for continuous, real-time edge processing. Running concurrent AI models alongside optical flow analytics limits the ability to process ultra-high-resolution video or extremely high frame rates without risking thermal throttling or frame drops.

- Environmental Dependencies and Model Drift:

The accuracy of object detection and geometric orientation metrics heavily depends on environmental factors such as poor lighting, low-resolution camera feeds, and extreme camera angles. Furthermore, AI models may experience "model drift" as real-world conditions change (e.g., varying clothing patterns or new camera setups), necessitating periodic calibration to maintain accuracy.

- Network Dependency for Cloud Synchronization:

Although edge processing ensures that local analytical decisions are made instantly, the system relies on stable network connectivity to publish MQTT payloads and update the Eclipse Ditto Digital Twin. During severe network outages, cloud dashboard synchronization may experience lag, even though local edge-level counting continues to function.

- Simulation Behavioral Simplifications:

The Multi-Agent System (MAS) used for disaster shielding and evacuation simulations employs simplified behavioral rules for crowd dynamics. Because human panic and complex psychological responses are highly unpredictable, the simulated risk outcomes serve as approximations rather than absolute certainties.

B. Future Scope

- UAV and Autonomous Drone Integration:

To overcome blind spots and provide coverage in areas without fixed CCTV infrastructure, future deployments can integrate autonomous Unmanned Aerial Vehicles (UAVs). These drones could execute YOLOv8 edge analytics directly on onboard hardware and stream dynamic spatial data into the Digital Twin.

- Multi-Modal Sensor Fusion:

To address visual occlusion in extreme density scenarios, future iterations of D-SHIELD will aim to combine optical video data with non-visual sensors. Integrating pressure mats, Bluetooth-based crowd density beacons, and environmental sensors can provide a highly accurate, fused metric for spatial occupancy.

- Advanced Evacuation Optimization via Machine Learning:

The predictive simulation engine can be enhanced by integrating reinforcement learning algorithms. This would allow the system to automatically optimize exit routing, dynamic crowd signalling, and evacuation strategies across a multitude of simulated disaster scenarios.

- City-Scale Digital Twin Deployments:

While the current system excels at venue-level monitoring, the architecture can be scaled to support city-level event management. By integrating public transit data, traffic systems, and broader smart-city infrastructure, D-SHIELD can enable coordinated, multi-layered emergency responses for massive public festivals or urban demonstrations.

- Automated Multilingual Interventions:

To improve direct communication with the crowd during an escalating hazard, future versions can integrate text-to-speech systems tied to the Digital Twin's alert engine. This would allow D-SHIELD to automatically issue clear, multilingual audio guidance via connected venue sound systems to safely redirect foot traffic before a crush occurs

REFERENCES

[1] M. Shah, S. Goyal, and S. Salvi, “Crowd density estimation and real-time monitoring using digital twins in transportation hubs,” in *Proc. PuneCon 2024*, 2024.

[2] P. Krishnakumari, S. Hoogendoorn-Lanser, J. Steenbakkers, and S. Hoogendoorn, “Crowd Safety Manager: Towards data-driven active decision support for planning and control of crowd events,” TRB Preprint, 2023, doi: 10.48550/arXiv.2308.00076.

[3] P. Shukla and S. Shukla, “UAV based disaster detection and crowd sensing using deep learning models,” in Proc. 26th Int. Conf. Distributed Computing and Networking (ICDCN), Jan. 2025, pp. 414–419, doi: 10.1145/3700838.3703685.

[4] M. Sun, Y. Wang, and Z. Zhao, “Stampede alert clustering algorithmic system based on tiny-scale strengthened DETR,” arXiv Preprint, 2024.

[5] M. H. K. Khel, K. A. Kadir, S. Khan, M. N. M. M. Noor, H. Nasir, N. Waqas, and A. Khan, “Realtime crowd monitoring – Estimating count, speed and direction of people using hybridized YOLOv4,” IEEE Access, vol. 11, 2023.

[6] S. van den Bogaard, T. Maessen, E. van Hilten, Y. Jin, H. van Iterson, S. Houben, and R.-H. Liang, “CrowdCollab: A multi-user, multi-perspective collaborative tool for crowd planning,” in Proc. 19th Int. Conf. Tangible, Embedded, and Embodied Interaction (TEI ’25), Mar. 2025, pp. 1–7, doi: 10.1145/3689050.3706007.

[7] I. Alba, J. Troya, and C. Canal, “Towards a digital twin system for human crowd motion prediction,” Universidad de Málaga – ITIS Software, 2024.

[8] B. Li, H. Huang, A. Zhang, P. Liu, and C. Liu, “Approaches on crowd counting and density estimation: A review,” *Pattern Analysis and Applications*, vol. 24, pp. 853–874, 2021.