Deep Learning Based Predictive Crowd Stampede Analysis and Rerouting

Manavi J[1*], Maithri A Humbarwadi[2], Prarthana A[3], Prashasthi Nand Reddy[4], Nethravathy V[5] and Bhanushree K J[6]

[1,2,3,4,5,6]Department of Computer Science and Engineering, Bangalore Institute of Technology, Bengaluru, India.

*manavii.jagadish@gmail.com, maithri.humbarwadi@gmail.com, prarthanaashwath24@gmail.com, prashasthi.n.reddy@gmail.com, netra.anil@gmail.com, Kjbhanushree@bit-bangalore.edu.in

Abstract. Sudden crowd crushes at festivals, rallies, or sports events turn safe gatherings deadly in seconds when panic ripples through packed spaces. Guards watching fixed cameras can't keep up with the chaos, missing early shoves or squeezes until it's too late. Our setup layers fast people-spotters YOLOv8 + Deformable DETR with motion trackers ByteTrack, feeding flow maps into ConvLSTM/LSTM for speed bursts and a GCN for group pressure reads—fusing both to flag hot zones and reroute via A* paths [1-4]. Tested on MOT20's jammed street clips, Layer 2 nails 98% motion alerts while Layer 3 hits 85% on crowd vibes, proving it catches trouble early for real fixes. Tackles blocks, group math, and live tweaks head-on.

Keywords: Crowd Stampede Prediction, YOLOv8, DETR, ConvLSTM, GCN, Farneback Optical Flow, Behavioral Modeling, Rerouting

1 Introduction

Large public gatherings such as pilgrimages, concerts, league matches, and political rallies can shift from festive to life-threatening within a very short time when panic spreads through a tightly packed crowd. In such situations, people often have limited visibility and almost no room to maneuver, so a small shove or a sudden rush can cascade into a dangerous crush, as illustrated by the incidents in Fig.1. Conventional crowd supervision still depends mainly on guards watching fixed CCTV feeds or patrolling on the ground, which is difficult to scale and often leads to tiredness, missing blind spots, and delayed reactions. With recent advances in computer vision and deep learning, it has become possible to monitor dense crowd scenes automatically, capture subtle motion trends, and detect early zones of congestion or abnormal behavior in near real time.

Traditional approaches to crowd monitoring, including manual observation and fixed-position cameras, are often unreliable because of human error, visibility constraints,

S. Bhalerao et al. (eds.), *Proceedings of the 2nd International Conference on Recent Advancement and Modernization in Sustainable Intelligent Technologies & Applications (RAMSITA-2026)*, Advances in Intelligent Systems Research 207,
https://doi.org/10.2991/978-94-6239-678-4_28

and the inability to analyze real-time crowd dynamics. With advancements in Computer Vision and Artificial Intelligence, deep learning techniques have emerged as powerful tools for automated crowd analysis, capable of detecting subtle motion patterns, behavioral anomalies, and potential congestion zones.Earlier work on crowd safety has explored multiple complementary ideas. Abnormal motion has been detected by analyzing abrupt changes in individual trajectories and motion paths observed in surveillance footage [1].

Social-force-based models describe how people influence one another's movement, showing that sudden changes in interaction forces can signal panic-like behavior [2].

Dynamic texture representations combine appearance and motion so that deviations from typical spatio-temporal patterns are highlighted as anomalies [3].

Other methods divide videos into small 3-D patches and compare learned global and local features, allowing fast localization of unusual events when patch similarities differ from normal statistics [4].

Fig 1(a). Shanghai Chen Yi Square stampede

Fig 1(b). Recent RCB Stamped in Bangalore.

Fig 1(c). Crowd crushes: tragedy at Itaewon

Fig 1. Shows images recognizing stampedes across the globe.

To build on these ideas, this work proposes a hybrid deep learning framework that focuses specifically on early stampede risk and route guidance. The system combines YOLOv8 and Deformable DETR for robust pedestrian detection under occlusion, extracts spatio-temporal motion features using Farneback optical flow, and models sequential behavior through a ConvLSTM–LSTM network, while a Graph Convolutional Network captures interaction patterns within the crowd. By combining these temporal and spatial risk indicators, the framework aims to identify unstable

regions with high confidence and to generate safer alternative routes that support proactive crowd management in real-world deployments.

2 Literature Review

Research on stampede prevention has produced a wide range of techniques for reading dense crowd motion and spotting early signs of danger. Recent work especially leans on deep learning and richer motion features to move from simple counting towards behavior-aware risk detection.

One line of work combines motion information with ensemble learning. In, optical-flow–based entropy features are passed to a stacked classifier built from random forests and SVMs, leading to more sensitive recognition of abnormal motion patterns across several benchmark datasets [5].

Detection of small, partially hidden pedestrians has improved with transformer architectures. The method augments a deformable attention backbone with multi-scale feature fusion geared toward tiny targets, allowing the network to pick out occluded individuals in cluttered environments such as airport terminals more reliably [6].

Crowd counting and behavior tagging are brought together in, where CSRNet is responsible for generating high-quality density maps while UNet performs semantic segmentation to mark regions with different behavior types, such as aggressive versus calm groups. This joint design reduces counting error and supports behavior-aware localization [7]. The ACSAM framework revisits abnormal behavior recognition using CNNs that learn both appearance cues and crowd-level behavior descriptors within a unified network. Evaluations on public datasets show that this approach raises both accuracy and recall compared with models that rely only on visual appearance or only on motion statistics [8]. Temporal dynamics have also been modeled with recurrent networks. The approach feeds spatiotemporal texture descriptors into LSTM-based models under a probabilistic formulation, enabling the system to capture how crowd behavior evolves over time and to improve anomaly detection performance on standard surveillance datasets [9]. Geometric graph neural networks have been explored for interaction-aware prediction. In, pedestrians are represented as nodes with edges defined by realistic interaction neighborhoods based on field of view, direction of travel, and spatial kernels. This structure helps the model forecast human trajectories more accurately than fully connected graphs, though the cost of manual feature engineering and graph processing makes scaling to very large crowds challenging [10].

To focus directly on motion irregularities, CrowdVAS-Net merges velocity, acceleration, and saliency features obtained from deep CNNs and feeds them into a classifier for abnormal crowd-motion detection. On a large video collection, the framework reaches an overall accuracy of about 77.8%, demonstrating its usefulness for early indication of disaster-prone behavior [11].

The MCNN model improves density estimation by using three parallel convolutional branches with different receptive fields. This layout compensates for strong perspective changes in surveillance views, where head size varies sharply with depth, and thus yields more stable head-count estimates across the image [12].

Beyond purely visual methods, applies Social Network Analysis to historical stampede cases to map how different risk factors interact. The study highlights nodes such as excessive density, unmanaged bottlenecks, and weak control measures as central elements in the network of causes, offering guidance for preventive planning even though it does not operate directly on video data [13].

For live monitoring, it employs deep object detectors like YOLOv3 together with tracking schemes such as DeepSORT to build real-time people-detection and counting systems. Optimized GPU implementations allow these pipelines to maintain high accuracy in typical public-venue footage [14].

Wavelet-based spatiotemporal texture (STT) representations provide another route to anomaly detection. The method extracts texture descriptors from video sequences and uses statistical decision rules to flag departures from normal patterns, achieving real-time performance thanks to its relatively light computations [15].

Finally, combines moving-object detection, density estimation, and behavior recognition into an early-warning framework for crowd stampedes. While the system offers a richer, multi-dimensional picture of crowd state using optical flow and pattern-recognition modules, it still faces a practical trade-off between speed and detection accuracy, which can delay alerts in very dynamic scenes [16].

3 ScatteRout- Early Stampede Detection System

The proposed ScatteRout pipeline is built as a sequence of deep learning modules that first detect people, then follow their motion, and finally reason about risk at both temporal and spatial levels. YOLOv8 and Deformable DETR operate in parallel so that clearly visible pedestrians and heavily occluded ones are both picked up, while ByteTrack links detections across frames to form stable identity-preserving tracks. Optical-flow features derived from these tracks are processed by a ConvLSTM/LSTM block, which learns motion patterns that deviate from normal flow and therefore act as early indicators of crowd instability. instability.

On top of this, a Graph Convolutional Network is used to study how individuals are arranged with respect to one another and how their local interactions evolve over time. By combining the temporal risk from the motion model with the spatial risk inferred from the graph, the system marks windows and regions that are potentially unsafe; these are treated as obstacles for a dynamic A* routing module that recomputes safer paths whenever the risk map changes, keeping the guidance in Fig. 2–5 aligned with real-time crowd conditions.

3.1 MOT20 dataset- Dataset Description

All experiments are carried out on the MOT20 benchmark, which is specifically designed for tracking in very dense urban scenes. The dataset consists of eight long video sequences captured in busy streets, plazas, and pedestrian corridors, containing

more than four thousand labeled pedestrian trajectories that span situations from light traffic to severe congestion. Each sequence provides challenging variations in viewpoint, illumination, occlusion, and motion style, making MOT20 a suitable test bed for behavior prediction and stampede-risk assessment in high-density environments [17, 18].

3.2 Model Architecture

The overall architecture aims to move from raw frames to actionable safe routes by tightly coupling detection, tracking, motion analysis, interaction modeling, and path planning. YOLOv8, Deformable DETR, Farneback optical flow with ConvLSTM/LSTM, and a GCN are integrated into a three-layer framework whose data flow is summarized in Fig. 4. Together, these components provide reliable person localization, rich spatio-temporal descriptors, and graph-based behavior inference, enabling the system to recognize congestion patterns and trigger rerouting when necessary.

3.2.1 Layer 1 — Detection and Fusion

YOLOv8 (convolution-based local detector)YOLOv8 serves as a fast, one-stage convolutional detector that processes each frame and outputs pedestrian bounding boxes along with class scores and objectness values, offering precise detections for individuals who are relatively well separated from the crowd

Deformable DETR (transformer-based global detector)Deformable DETR extends transformer detection with deformable multi-head attention so that the model can focus on sparse, informative key points rather than scanning dense feature grids. By aggregating multi-scale context and relational cues, it often succeeds in finding small or heavily overlapped pedestrians that purely convolutional methods miss.

Weighted Boxes Fusion and ByteTrack association Predictions from YOLOv8 and Deformable DETR are merged using Weighted Boxes Fusion (WBF) to produce a single, consolidated set of bounding boxes per frame, reflecting the consensus of both detectors.

These fused detections are then linked through time with ByteTrack in a tracking-by-detection fashion, resulting in trajectories of the form

$$Track = \{ID, Frame, (x1, y1, x2, y2), Velocity, Confidence\} \quad (1)$$

which serve as structured input to the subsequent layers and correspond to the flow shown in Fig. 2

3.2.2 Layer 2 — Temporal Risk Modeling

Dense optical flow and feature extraction. To capture how motion evolves between frames, dense optical flow is computed using the Farneback algorithm on the frame sequence, as visualized in Fig. 3. For each tracked pedestrian box from Layer 1, the local flow field is summarized by statistics such as mean magnitude, spread, and directional variability, yielding compact descriptors of short-term motion.

Sliding-window aggregation and heuristic labels. Instead of treating frames independently, the tracks are segmented into overlapping windows of 30 frames with a stride of 10 so that short motion segments can be analyzed.A simple rule-based scheme assigns each window a binary label $R \in \{0,1\}$, where R=1 marks windows whose motion characteristics hint at possible instability; these pseudo-labels guide the temporal risk model during training.

ConvLSTM for spatio-temporal risk. The ConvLSTM network receives the stacked flow magnitude and orientation maps for each window and maintains a recurrent memory of how motion patterns change over time. After passing through convolutional and recurrent layers, a sigmoid output unit produces a risk probability

$$P_{risk} \in [0,1] \tag{2}$$

for every temporal window, indicating how likely that segment is to contain abnormal crowd motion.

3.2.3 Layer 3 — Spatial Behavioral Modeling

Layer 3 focuses on collective behavior by transforming tracked positions into an interaction graph. Using the centroid coordinates from Layer 1, a Ball-Tree–based neighbor search connects each pedestrian to nearby individuals within a preset distance or among the closest neighbors, forming a graph whose structure is depicted in Fig. 4.

Node-level features that summarize local density, interaction counts, speed statistics, and direction variance are then propagated through a Graph Convolutional Network, which learns how neighborhood patterns correlate with risk. For every sliding window, the GCN outputs a behavioral risk label, represented as

$$Behavioral\ Risk\ \{TrackID, Window, RiskLabel\} \tag{3}$$

distinguishing windows judged as “Risk” from those regarded as “No Risk.”

3.2.4 End – to – pipeline

Finally, outputs from the temporal model in Layer 2 and the spatial model in Layer 3 are brought together in the Divergence Computation Module highlighted in Fig. 5. This module aggregates the two risk estimates at both frame and window scales, resolves disagreements, and assigns each region a unified binary label of Risk or No Risk. Regions tagged as risky are encoded as high-cost or blocked nodes in the spatial search graph, allowing an A* planner to generate alternative routes that circumvent unstable areas; these safe paths can then be projected on a geographic map and updated continuously as the risk distribution changes.

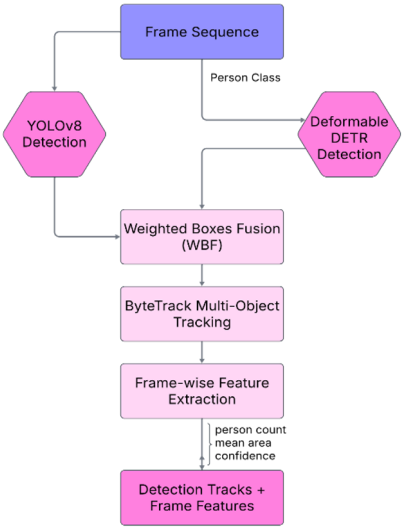

Fig.2. Layer 1 Detection

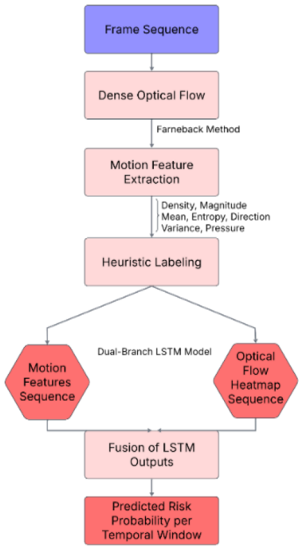

Fig.3. Layer 2 Motion prediction

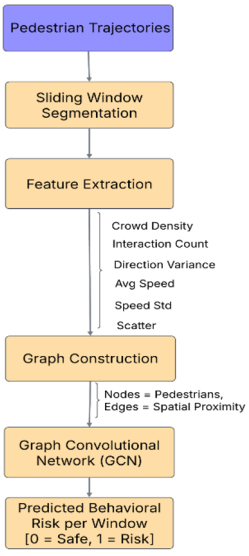

Fig.4. Layer 3 Behaviour Inference.

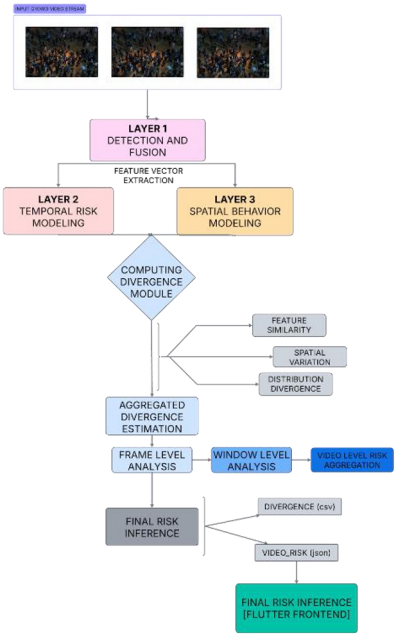

Fig.5. Multi-Layer Pipeline.

4 Results

All models were trained and evaluated in a Google Colab setup with an Intel x86_64 host processor and an NVIDIA Tesla T4 GPU, using Python 3.12.0 for implementation. Data loading, pre-processing, plotting, and metric calculation were carried out with standard libraries such as pandas and matplotlib, while the hyper-parameters for each layer are summarized in Table 1 and the performance indicators in Table 2.

Table 1. Parameters and Values

Parameters	**Layer 1** (YOLOv8/DETR/WBF)	**Layer 2** (Dual LSTM)	**Layer 3** (GCN)
Activation (Hidden Layer)	SiLU (YOLOv8), ReLU (DETR)	ReLU	ReLU
Activation (Output Layer)	Sigmoid / Softmax	Sigmoid	Softmax
Loss Function	CIoU+BCE (YOLO), Focal (DETR)	Binary Cross-Entropy	Cross entropy
Batch Size	16	8	N/A (all nodes)
Epochs	20	12	20
Optimization function	SGD/AdamW	Adam	Adam

Table 1 shows that the three layers use different activation and loss combinations tailored to their roles: YOLOv8/DETR rely on SiLU/ReLU with CIoU+BCE and Focal losses for dense detection, the dual LSTM block applies ReLU with Binary Cross-Entropy to classify temporal instability, and the GCN adopts ReLU–Softmax with cross-entropy for graph-level risk prediction. Batch size and epoch choices also reflect computational demands—Layer 1 operates with larger batches and more epochs, Layer 2 is trained with smaller batches over fewer epochs to stabilize sequence learning, while Layer 3 processes all nodes jointly per step.

Table 2. Comparative summary of different models.

Layer/Model	**Accuracy**	**Precision**	**Recall**	**F1**
Layer 1 – Detection Fusion	0.807	0.788	0.746	0.137
Layer 2 – Dual LSTM	94.2%	0.91	0.89	0.90
Layer 3 – GCN (risk-averse)	85%	0.84	0.85	0.81

From Table 2, the fused detection layer (Layer 1) achieves an accuracy of 0.807 with precision 0.788 and recall 0.746, confirming that combining YOLOv8 and Deformable DETR via WBF produces a competitive detector for dense scenes even though its F1-score is low when treated as a stand-alone classifier. Fig. 6 visualizes this behavior by comparing frame-wise person counts from YOLOv8, Deformable DETR, and the fusion module against ground-truth annotations, where the fused curve tracks the reference trend more closely than either individual model.

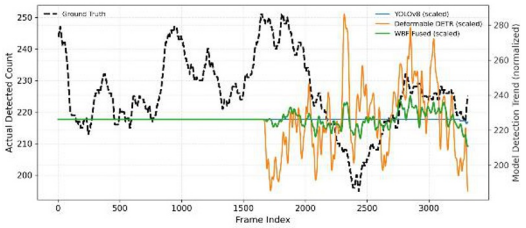

Fig. 6. Frame-wise comparison of ground-truth and model-based person detection trends using YOLOv8, Deformable DETR, and WBF fusion.

The temporal instability module in Layer 2 is evaluated using five-fold cross-validation on motion windows derived from Layer 1 tracks, with details listed in Table 3. Across folds, the ConvLSTM–LSTM fusion maintains accuracy around 0.98, precision between 0.98 and 1.00, and F1-scores of 0.97–0.98, yielding a mean F1 of 0.98 ± 0.01 and AUC of 0.99 ± 0.01, as further illustrated by the confusion matrix, ROC curve, and precision–recall curve in Fig. 7.

Table 3. Performance of Layer 2 ConvLSTM–LSTM fusion model across five folds.

Metric	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean ± SD
Accuracy	0.97	0.98	0.99	0.99	0.98	0.98 ± 0.01
Precision	0.98	0.99	1.00	1.00	0.98	0.99 ± 0.01
Recall	0.96	0.98	0.97	0.97	098	0.98 ± 0.01
F1-score	0.97	0.98	0.98	0.98	0.98	0.98 ± 0.01
AUC	0.99	0.99	0.99	1.00	0.99	0.99 ± 0.01

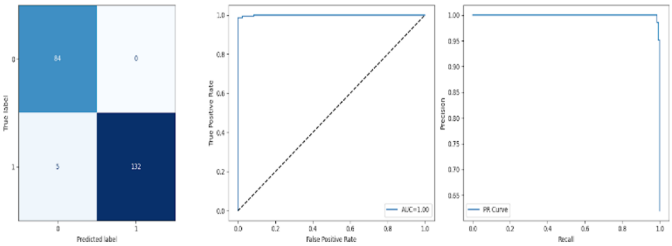

Fig.7. Layer 2 ConvLSTM–LSTM performance. (Left) Confusion matrix. (Center) ROC curve (AUC ≈ 1.00). (Right) Precision–Recall curve.

For spatial behavioral reasoning, the GCN in Layer 3 is trained to classify sliding windows as “Risk” or “No Risk,” with training and validation curves plotted in Fig. 8. Accuracy stabilizes near 88% on the training set and about 85% on validation data, indicating good convergence under the chosen risk-averse configuration.

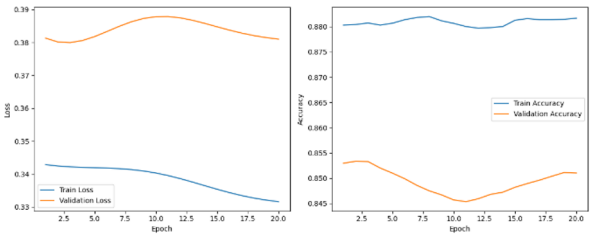

Fig. 8. Layer 3: Validation loss and accuracy curves (left and right).

Class-wise statistics in Table 4 and the confusion matrix in Fig. 9 reveal the trade-off behind this configuration. The model achieves very high recall (0.99) and F1-score (0.92) for the Risk class, at the cost of lower recall (0.14) for the No-Risk class and some over-prediction of risky windows; the overall accuracy remains 0.85, and macro-averaged F1 is 0.58, which is acceptable given the safety-critical goal of missing as few dangerous situations as possible.

Table 4. Layer 3 (GCN) class-wise precision, recall, and F1-score.

	Precision	**Recall**	**F1-score**	**Support**
No Risk	0.73	0.14	0.24	5267
Risk	0.86	0.99	0.92	27335
Accuracy			0.85	32602
Macro avg	0.80	0.57	0.58	32602
Weighted avg	0.84	0.85	0.81	32602

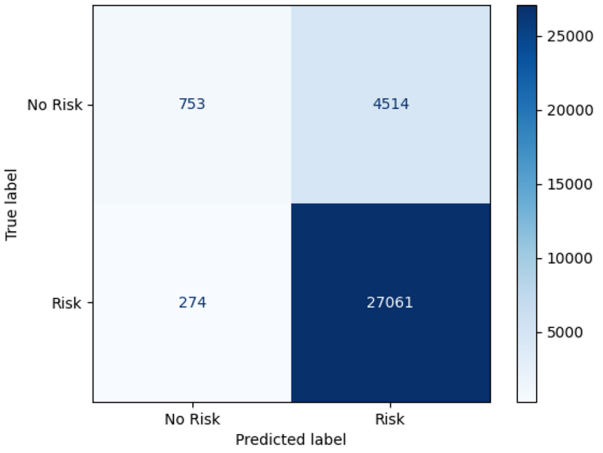

Fig.9. Confusion matrix (Layer 3)

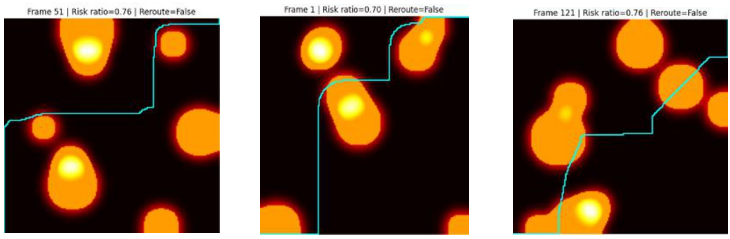

Fig. 10. Rerouting outputs across frames: The path continuously adjusts in response to changing risk patterns.

The rerouting module exploits these risk maps to adapt navigation in real time. As shown in Fig. 10, when new high-risk regions emerge, the A* planner dynamically shifts the recommended path around them, generating smooth alternative routes that steer agents away from unstable zones while keeping the trajectory updated over successive frames.

5 Conclusion

The study introduces a multi-layer deep learning framework that links person detection, motion analysis, interaction modeling, and path planning into a single pipeline for early crowd-stampede risk prediction and rerouting. On the MOT20 benchmark, the temporal module reaches an average accuracy of 98%, and the graph-based behavioral layer attains an F1-score of 0.81 under a deliberately risk-averse setting, demonstrating that the combination of layers can both localize instability and support preventive navigation decisions.

Current results confirm that the system can reliably flag emerging risk zones and suggest safer routes, but they also point to future work on deploying the models on edge-GPU hardware for real-time operation, extending coverage through multi-view camera fusion, and exploring reinforcement-learning-based strategies to refine rerouting behavior under changing crowd conditions.

References

1. Bhuiyan, M.R., Abdullah, J., Hashim, N. et al. Hajj pilgrimage abnormal crowd movement monitoring using optical flow and FCNN. J. Big Data 10, 86 (2023). https://doi.org/10.1186/s40537-023-00779-4
2. Mehran, R., Oyama, A., Shah, M.: Abnormal Crowd Behavior Detection using Social Force Model. In: Proc. CVPR (2009).
3. Mahadevan, V., Li, W., Bhalodia, V., Vasconcelos, N.: Anomaly Detection in Crowded Scenes. In: Proc. CVPR (2010).https://doi.org/10.1109/CVPR.2010.5539872
4. Sabokrou, M., Fathy, M., Hoseini, M., Klette, R.: Real-Time Anomaly Detection and Localisation in Crowded Scenes. In: Proc. CVPR (2015).

5. Cob-Parro, A.C., Losada-Gutiérrez, C., Marrón-Romera, M., Gardel-Vicente, A., Bravo-Muñoz, I., Sarker, M.I.: A Proposal on Stampede Detection in Real Environments. In: IPIN WiP Proceedings (2021).
6. Sun, M., Wang, Y., Zhao, Z.: Stampede Alert Clustering Algorithmic System Based on Tiny-Scale Strengthened DETR. Intelligent Information Processing Laboratory, Beijing Jiaotong University (2024).
7. Ganga, B., Lata B.T., Rajshekar, Venugopal K.R.: Deep Learning Algorithm using CSRNet and UNet for Enhanced Behavioral Crowd Counting in Video. Int. J. Comput. Digit. Syst. 17(1) (2025).
8. Wu, Y., Qiu, L., Wang, J., Feng, S.: The use of Convolutional Neural Networks for Abnormal Behavior Recognition in Crowd Scenes. Inf. Process. Manag. 62 (2025).
9. Dey, A., Mohammad, F., Ahmed, S., Sharif, R., Saif, A.F.M.S.: Anomaly Detection in Crowded Scene by Pedestrians Behaviour Extraction using Long Short Term Method. IJEME 9(1), 51–63 (2019).
10. Honarvar, S., Díaz-Mercado, Y.: Geometric Graph Neural Network Modeling of Human Interactions in Crowded Environments. IFAC (2024).
11. Gupta, T., Nunavath, V., Roy, S.: CrowdVAS-Net: A Deep-CNN Based Framework to Detect Abnormal Crowd-Motion Behavior in Videos for Predicting Crowd Disaster (2019).
12. Suvarna Kumari, T.: Early Detection and Prevention of Anomalies using MCNN. Int. J. Eng. Innov. Adv. Technol. 2(1) (2020).
13. Guo, X., Niu, L., Guan, H.: The Mechanism of Crowd Stampede Based on Case Statistics through SNA Method. Tech. Vjesnik 28(2), 548–555 (2021).
14. Mokayed, H., Quan, T.Z., Alkhaled, L., Sivakumar, V.: Real-Time Human Detection and Counting System Using Deep Learning Computer Vision Techniques. Artif. Intell. Appl. 14 (2023).
15. Wang, J., Xu, Z.: Crowd Anomaly Detection for Automated Video Surveillance (2015).
16. Liu, S., Zhu, Z., Cheng, Q., Zhang, H.: Analysis and Design of Public Places Crowd Stampede Early-Warning Simulating System. In: IEEE Intl. Conf. Industrial Informatics (2016).
17. Dendorfer, P., Rezatofighi, H., Milan, A., Shi, J., Cremers, D., Reid, I., Roth, S., Schindler, K., Leal-Taixé, L.: MOT20: A Benchmark for Multi-Object Tracking in Crowded Scenes. arXiv:2003.09003 (2020).
18. MOT20 Dataset. https://motchallenge.net/data/MOT20/