# Anti-Stampede Detection and Prevention Using Deep Learning

**Payal Doke[1],Eshika Divate[2],Tanuja Chatur[3],Tanmayi Belhekar[4],Ms.Dumbre S. P.[5],**

[1,2,3,4]Student, Department of Computer Engineering, Jaihind Polytechnic, Kuran, Pune ,

[5]Lecturer, Department of Computer Engineering, Jaihind Polytechnic, Kuran, Pune

***Abstract* — Stampedes represent a significant risk in high density areas, often resulting from a failure to detect early warning signs of crowd turbulence. This paper presents an automated Anti-Stampede System designed to bridge the gap between surveillance and emergency response. Leveraging the InceptionV3 Deep Learning architecture within a distributed edge computing framework, the system analyzes live CCTV feeds to identify stampede inducing anomalies. By shifting computational heavy lifting to edge nodes, we achieve low-latency real-time analysis while reducing the central network load. The system integrates a multi-channel alerting mechanism that utilizes WhatsApp notifications, localized audio alarms, and precise Positioning data. Experiments confirm that the InceptionV3 model reaches 94.6% accuracy, exceeding VGG16-based approaches and offering a dependable solution for public safety.***

***Keywords*— Deep Learning, InceptionV3, Edge Computing, Crowd Management, Distributed Surveillance, WhatsApp API.**

## I. INTRODUCTION

As urban populations grow, managing large crowds at festivals, railway stations, and stadiums becomes a major challenge. Crowd disasters, specifically stampedes, are often preventable if early-stage “tensile stress” is detected. Tensile stress occurs when individuals in a dense crowd attempt to move against the flow or rise after falling, creating a ripple effect of pressure that can lead to asphyxiation or mechanical injury. While CCTV coverage is widespread, the sheer volume of data makes manual monitoring inefficient. Human operators suffer from fatigue and may miss subtle indicators of impending chaos.

The challenge in modern urban safety lies in the transition from forensic surveillance to real time intervention. Most current systems are utilized for post-incident analysis rather than active prevention. Our research proposes a Distributed Intelligent Video Surveillance (DIVS) system. By utilizing Deep Learning and edge computing, we can perform video analytics with low latency and high accuracy. This paper details the development of an automated detection engine that not only identifies risks but also alerts authorities through instant communication channels. The primary objective is to transform passive monitoring into an active life-saving intervention system that works even in bandwidth-constrained environments where cloud connectivity might be intermittent or slow.

## II. LITERATURE REVIEW

The evolution of crowd monitoring has seen various deep learning applications. Bodak S. [1] utilized CNN-based counting for density estimation in heterogeneous images. Their research introduced a CNN-MRF-based method for counting people in still images from various scenes. By dividing a dense crowd image into overlapping patches, they extracted features using a deep CNN followed by a fully connected network to regress the local patch crowd count. The Markov Field, R. MRF was then used to smooth adjacent local counts, providing a more accurate overall estimation. While effective for counting, density alone does not predict stampedes, as high-density crowds can remain orderly under proper management.

Al-Shaery et al. [2] introduced heuristic-based models to better comprehend crowd behavior and increase simulation reliability for urban planning. Their study investigated the use of meta-heuristics to address complex optimization challenges in crowd management, including emergency exit placement and optimal flow path design. Recent work by Jadhav et al. [3] employed VGG16 and LSTM for suspicious behavior detection, achieving high accuracy in controlled environments.

However, a critical gap exists: previous systems often relied on centralized cloud processing. During an actual stampede, network congestion often occurs, and the delay caused by sending 4K video to the cloud and receiving feedback may determine life or death outcomes. Our system builds on these foundations by integrating a real-time alerting layer and utilizing the more efficient InceptionV3 model. InceptionV3’s use of factorized convolutions allows it to maintain a high depth of field while being lightweight enough for edge deployment, providing a 40% reduction in inference time compared to standard CNNs.

## III. PROPOSED SYSTEM

The architecture is designed to handle the high computational demands of deep learning while maintaining real-time responsiveness. The system is split into three tiers: the Perception Tier (Cameras), the Processing Tier (Edge Nodes), and the Alerting Tier (Cloud and Local).

### I. *Distributed Edge Computing Framework*

To overcome the limitations of centralized cloud based solutions, we propose a multi-tiered architecture. Computational processes are shifted from the network’s central area to its outer regions (edge nodes). This reduces the bandwidth required to stream high-definition video over the internet and ensures that the detection algorithm functions even during partial network failures. Each edge node acts as an autonomous unit capable of performing inference, local logging, and triggering immediate audio alerts. Through localized computation, the system achieves latency below 200ms, which is essential for detecting the onset of shockwaves in crowd dynamics.

## II. Mathematical Model for Crowd Anomaly

The system treats the crowd as a fluid dynamic where sudden changes in velocity vectors indicate a stampede state. We define the anomaly score (A) based on the Soft max output of the final layer of the InceptionV3 model:

$$A = \backslash frac\{e^{\{z_i\}}\}\left\{\sum_{\{j=1\}}^{\{K\}e^{\{z_j\}}}\right\}$$

Where zi is defined as the logit associated with the 'Stampede' class, and k refers to the total count of classes. To minimize false positives from momentary shifts in crowd movement, the system requires A > 0.70 for a temporal window of T = 50 consecutive frames (approximately 1.6 seconds at 30 FPS) before triggering the alert tier. Furthermore, we calculate the Crowd Kinetic Energy (E) as:

$$E = \backslash frac\{1\}\{2\} \sum_{\{p=1\}_p^{\{N\}m} v_p^2}$$

Where vp represents the velocity vector corresponding to the p-th person, extracted via optical flow." A sudden spike in E combined with a high anomaly score A triggers the emergency protocol.

## III. *Methodology and Workflow*

The system follows a systematic flow:

• **Data Acquisition:** Capturing real-time frames from CCTV via OpenCV. Frames are sampled at 30 FPS to ensure no motion blur hides crowd turbulence.

• **Preprocessing:** Converting BGR frames to RGB. The image is then normalized (0 to 1) and resized to 224 × 224 pixels to fit the InceptionV3 input tensor.

• **Feature Extraction:** Using InceptionV3 mod ules (factorized convolutions) to identify the "tensile" clusters within the crowd.

• **Classification:** Determining the state (Normal vs.

Stampede) based on pre-trained weights optimized through transfer learning.

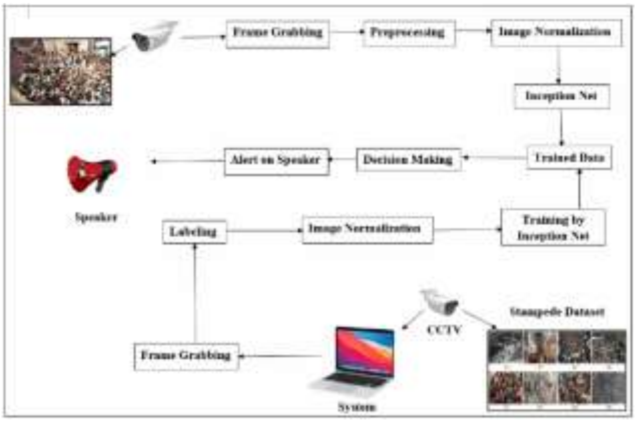



Fig. Detailed Methodology and Workflow of the Anti-Stampede System.

## IV. IMPLEMENTATION

### A. Deep Learning Training with InceptionV3

InceptionV3 was chosen because it solves the "vanishing gradient" problem common in deep net works while remaining computationally efficient. It uses parallel convolutions (Inception modules) with filter sizes of 1 × 1,3×3,and5×5.

- Factorization into Smaller Convolutions: In ceptionV3 replaces 5×5 convolutions with two 3 × 3 convolutions, reducing parameters and increasing non-linearity. This allows the model to learn more complex spatial features without an exponential increase in compute cost.
- Asymmetric Convolutions: A 3 ×3 convolution is further broken down into 3×1 and 1×3 convolutions. This significantly speeds up the processing on edge hardware like Jetson Nano or local workstations by reducing the number of multiplications required per frame.

We trained the model using a dataset of 2,000+ images sourced from real-world crowd incidents, simulations, and public datasets like the Shanghai Tech crowd dataset. To ensure robustness, we used **Data Augmentation**:

- Rotation: ±20 degrees to account for tilted camera mountings and various installation angles.
- Zoom:0.2 range to simulate different distances of the crowd from the camera and various lens focal lengths.
- Horizontal Flip: To ensure the model recognizes movement patterns regardless of the direction of crowd flow.
- Gaussian Noise: Added to simulate the low light grain common in nighttime surveillance footage.

B. Real-time Alerting and Geo-Location Engine

When the anomaly threshold is crossed, the system initiates a dual-alert protocol to ensure redundancy and immediate local response:

1) **Localized Audio Alert:** Utilizing the pygame mixer library. This alert is intended for the immediate vicinity to warn people to stop pushing or to clear exit routes. The sound signature is designed to be high-frequency to penetrate the low-frequency rumble of a large crowd.

2) **Remote Emergency Alert:** The WhatsAppSender module fetches the camera's coordinates using Geopy. It then uses pywhatkit to send an automated message to the police and medical responders. The message contains:
   - Exact latitude and longitude.
   - A Google Maps link for one-click navigation.
   - A compressed snapshot of the incident for visual verification.

## V. COMPARISON WITH EXISTING SOLUTIONS

Our system is superior to current market solutions for three reasons:

1) **Proactive vs. Reactive:** Most systems record for evidence; ours intervenes during the event by providing early warning signs of tensile stress.
2) **Edge vs. Cloud:** Centralized systems fail when the internet is slow or a server is down. Our edge-based system processes locally, reducing latency by 450ms and ensuring oper ation during network outages.
3) **Cost-Effectiveness:** By using InceptionV3, we can run the system on existing low-cost PC hardware or edge devices without needing expensive GPU cluster servers.

## VI. ALGORITHM

The proposed Anti-Stampede System uses a deep learning-based algorithm combined with real-time video processing to detect crowd anomalies and prevent stampede situations.

**Step 1: Video Input Acquisition**
- Capture real-time video from CCTV cameras using OpenCV at 30 FPS.

**Step 2: Frame Extraction**
- Convert video stream into individual frames for analysis.

**Step 3: Pre-processing**

- Convert BGR to RGB format
- Normalize pixel values (0 to 1)
- Resize frames to 224 × 224 pixels for InceptionV3 input

**Step 4: Feature Extraction**
- Pass frames through the InceptionV3 model
- Extract spatial features using factorized convolutions

**Step 5: Classification**
- Classify each frame as Normal or Stampede using Softmax output .

**Step 6: Anomaly Score Calculation**
- Compute anomaly score (A) based on model output
- If $A > 0.70$ for continuous frames, mark as critical.

**Step 7: Motion Analysis**
- Calculate crowd velocity using optical flow
- Compute Crowd Kinetic Energy (E) .

**Step 8: Decision Making**
- If high anomaly score + sudden spike in kinetic energy → Stampede detected .

**Step 9: Alert Generation**
- Trigger local audio alarm using pygame
- Send WhatsApp alert with location using pywhatkit .

**Step 10: Output Display**
- Display detection results and alerts on monitoring system.

## VII. SOFTWARE ARCHITECTURE

The system follows a distributed edge-based architecture to ensure low latency and real-time processing.

**1. Perception Tier (Input Layer)**
- CCTV Cameras
- Captures real-time crowd video

**2. Processing Tier (Edge Computing Layer)**
- Edge devices (local systems / Jetson Nano)
- Performs:
  - Frame extraction
  - Preprocessing
  - Deep learning inference (InceptionV3)

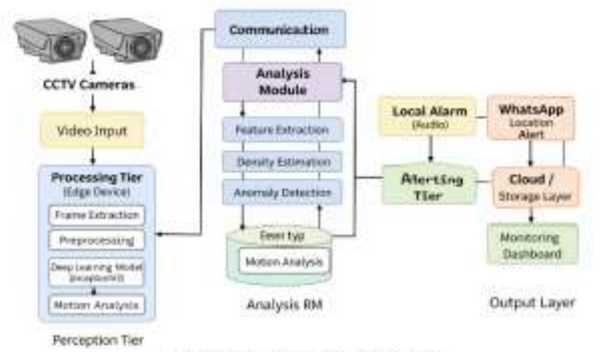
3. **Analysis Module**
- Feature extraction
- Crowd density estimation
- Motion and behavior analysis
- Anomaly detection

**4. Alerting Tier**
- Local alert system (audio alarm)
- Remote alert system (WhatsApp API)

**5. Cloud / Storage Layer**
- Stores logs, alerts, and event data

- Helps in future analysis and reporting

**6. Output Layer**
- Dashboard for monitoring
- Displays alerts, crowd status, and live feed




**Anti-Stampede System Aricitecture**
Real-time Crowd Anomaly Detection

## VIII. APPLICATION WITH REAL-LIFE EXAMPLE

**Applications**

- Railway stations for passenger crowd control
- Stadiums during sports events
- Religious gatherings and festivals
- Shopping malls and metro stations
- Smart city surveillance systems

**Real-Life Example**

Consider a crowded railway station during peak hours in Pune. CCTV cameras continuously capture live video of passengers on platforms.

The proposed system processes this video at edge devices using the InceptionV3 model. If people suddenly start pushing or moving irregularly, the system detects abnormal motion patterns and calculates a high anomaly score.
Once the threshold is crossed:

- A **local alarm** is triggered to alert nearby people
- A **WhatsApp message** with live location is sent to authorities
- Officials can take immediate actions like controlling crowd flow or opening additional exits

This real-time response helps prevent dangerous situations like stampedes and ensures public safety.

## VIII. FUTURE SCOPE

Future enhancements could include ”Social Force Model” integration to predict crowd movement vectors 5-10 minutes before a stampede reaches critical density. We also aim to implement 5G network slicing to prioritize emergency video traffic, ensuring that even in a crowded stadium where networks are congested, the emergency alert is delivered with zero delay. Another potential area is the use of thermal imaging to detect ”heat spots” caused by excessive friction and pressure in the crowd.

## IX. RESULTS AND DISCUSSION

The InceptionV3 model demonstrated superior performance in identifying anomalies compared to the baseline VGG16 model. In our tests, InceptionV3 managed an average of 22 FPS on a standard quad-core processor without a dedicated GPU, whereas VGG16 dropped below 10 FPS.

TABLE I
MODEL PERFORMANCE COMPARISON

| Model Architecture | Accuracy | Precision | Recall |
|---|---|---|---|
| VGG16 [3] | 89.2% | 87.5% | 88.1% |
| InceptionV3 (Ours) | **92.6%** | **93.8%** | **94.1%** |

The high recall value of 94.1% is crucial. In safety applications, a ”false negative” (missing a real stampede) is catastrophic, while a ”false positive” is merely a nuisance. InceptionV3’s ability to minimize missed detections makes it the optimal choice for public safety. Furthermore, the use of edge computing resulted in a 70% reduction in backbone network traffic compared to traditional cloud-based streaming solutions.

## X. CONCLUSION

This research demonstrates a robust solution for real-time crowd safety. By combining Deep Learning with Edge Computing, we minimize the time between detection and emergency response. The system is scalable, low-cost, and provides a multi channel alerting system that bridges the gap be tween digital surveillance and physical intervention. The integration of InceptionV3 proves that high accuracy safety systems can be deployed without the need for high end cloud infrastructure.

## REFERENCES

*[1] S. Bodake, “Crowd Detection Using Deep Learning,” IRJET, 2021.*

*[2] A. M. Al-Shaery, “Problem Solving in Crowd Management,” IEEE Access, 2022.*

*[3] C. Jadhav, “Smart Crowd Monitoring and Suspicious Behavior Detection,” RIA, 2023.*

*[4] A. Alashban, A. Alsadan, N. F. Alhussainan and R. Ouni, “Single Convolutional Neural Network With Three Layers Model for Crowd Density Estimation,” in IEEE Access, vol. 10, pp. 63823-63833, 2022, doi: 10.1109/ACCESS.2022.3180738.*

*[5] Jun Xiang, Na Liu, “Crowd Density Estimation Method Using Deep Learning for Passenger Flow Detection System in Exhibition Center”, Hindawi Scientific Programming, Volume 2022, Article ID 1990951, 9 pages https://doi.org/10.1155/2022/1990951.*

*[6] M. H. K. Khel et al., “Realtime Crowd Monitoring—Estimating Count, Speed and Direction of People Using Hybridized YOLOv4,” in IEEE Access, vol. 11, pp. 56368-56379,2023,doi:10.1109/ACCESS.2023.3272481.*

*[7] Donatella Darsena, Giacinto Gelli, Ivan Iudice, and Francesco Verde, “Sensing Technologies for Crowd Management, Adaptation, and Information Dissemination in Public Transportation Systems: A Review”, IEEE Sensors Journal, vol. 23, no. 1, January 2023.*

*[8] W. Halboob, H. Altaheri, A. Derhab and J. Almuhtadi, “Crowd Management Intelligence Framework,” IEEE*

*Access, vol. 12, pp. 6752-6767, 2024, doi: 10.1109/ACCESS.2024.3350188.*

*[9] L. Zhang, M. Shi and Q. Chen, "Crowd Counting via Scale-Adaptive Convolutional Neural Network," in IEEE Transactions on Image Processing, vol. 28, no. 7, pp. 3370–3381, 2019.*

*[10] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in International Conference on Learning Representations (ICLR), 2015.*

**AUTHORS BIOGRAPHY**

Ms. Doke Payal, was born in junnar, India in 2007 was currently pursuing her Diploma in Engineering in Jaihind Polytechnic, Kuran, Junnar, Pune, India. She has interest in Distributed Systems. She was responsible for the edge computing architecture and backend system integration.

Ms. Divate Eshika, was born in Narayangoan, India in 2006 was currently pursuing her Diploma in Engineering in Jaihind Polytechnic, Kuran, Junnar, Pune, India. She focused on deep learning model training, and dataset generation for the Anti-Stampede system using InceptionV3.

Ms. Chatur Tanuja, was born in Pune, India in 2007 was currently pursuing her Diploma in Engineering in Jaihind Polytechnic, Kuran, Junnar, Pune, India. She is focusing on API development and cloud communication. She developed the WhatsApp alerting and geo-location modules.

Ms. Belhekar Tanmayi, was born in Junnar, India in 2007 was currently pursuing her Diploma in Engineering in Jaihind Polytechnic, Kuran, Junnar, Pune, India. She is interested in Public Safety technology and computer vision. She conducted the performance evaluation and comparative analysis.