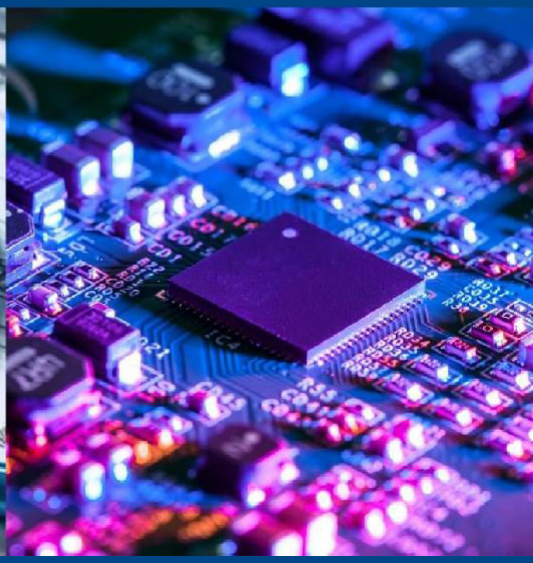
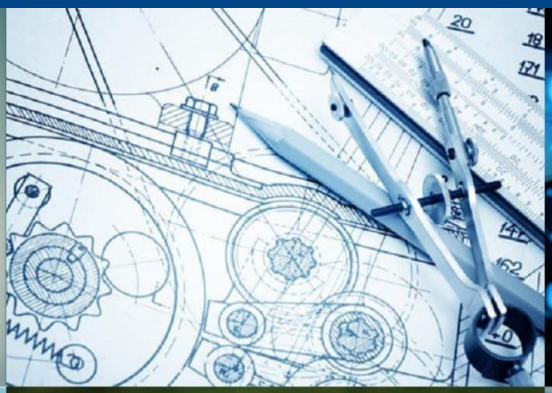
ISSN: 2582-7219

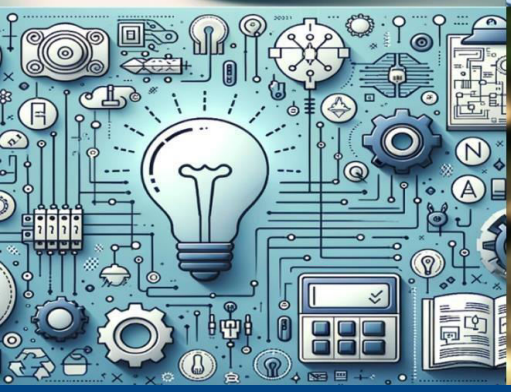




International Journal of Multidisciplinary Research in Science, Engineering and Technology

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.206

Volume 8, Issue 8, August 2025

© 2025 IJMRSET | Volume 8, Issue 8, August 2025| DOI:10.15680/IJMRSET.2025.0808051

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

CRONUS-VIEW: A REAL-TIME STAMPEDE RISK DETECTION AND SAFETY TRIGGER SYSTEM FOR URBAN SURVEILLANCE

Dr.M S Shashidhara, Atul Kumar

Professor & HOD, Department of MCA, AMC Engineering College, Bengaluru, India

Student, Department of MCA, AMC Engineering College, Bengaluru, India

ABSTRACT: The system is called the Stampede Detection System – YOLOv8 which is a smart surveillance system that detects overcrowding in real-time to prevent mass- casualty crowd-related incidents by sending emergency alerts. It uses the YOLOv8 model, an object detection model based on deep learning technologies, to detect humans and track the number of people in the video format in real time without needing a human to view the live video or camera feed. After the video frame has been obtained and pre-processed using OpenCV software, YOLOv8 uses spatial features to identify individuals in each frame. A mobile or IP camera serves as the video source, continuously capturing frames which are then analyzed in real time. When the number of detected individuals exceeds a predefined safety threshold, the system sends immediate alerts via email and SMS using integrated services like Gmail SMTP and Twilio. A Flask-based backend manages detect logic and video streaming, while a React- based frontend displays live video content that also has bounding boxes, crowd counts and control options. the automated, contactless system enhances public safety during large gatherings by permitting early detection of thrusting situations,. However, specific challenges still exist such as varying or low light, partial occlusion of a person or object, and dynamic crowd behavior, which inhibit accuracy and surround detection performance the system features represents an evolutionary improvement in smart surveillance. the creative combination of deep learning with enhanced, real time alerting, and user- friendly interactability creates innovative, scalable, responsive, contact or very close contact methods of actively managing sensitive situations in highly populated spaces.

KEYWORDS: YOLOv8, Crowd Detection, OpenCV, Flask, React, SMS Notifications, Email alerts, Asynchronous monitoring.

I. INTRODUCTION

This paper outlines a real-time stampede detection and crowd monitoring system which leverages advanced deep learning architectures, cutting-edge web frameworks, and live video streaming & analysis to enhance public safety in high density scenarios. The core detection system is the YOLOv8 (You Only Look Once version 8), a state-of-the-art object detection architecture, which has been specifically designed to detect humans in real-time with high-accuracy and low-latency. The YOLOv8 can detect people from live IP or mobile camera sources and determine crowd density, and sends alerts if it exceeds a defined threshold.

The backend is built using Flask, a lightweight Python web framework which hosts a trained YOLOv8 model, and handles detection logic, access to video streams, and alert handling. Alerts are sent via SMS (Twilio API) and email (Gmail SMTP) to concerned authorities when a potential stampede risk is identified. The frontend is built using React.js, providing an intuitive dashboard that displays the live camera feed with bounding boxes, real-time crowd count, and controls to start/stop the monitoring process.

The system's objective is to help event managers, authorities, and security to proactively and safely manage large capacity gatherings (e.g., religious gatherings, concerts, public events) in situations where human error occurs, and accurately responding to situations that may cause a large crowd panic is not always possible.

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

II. LITERATURE SYRVEY

[1] Patel et al. proposed a real-time crowd monitoring approach using YOLOv8, with a focus on detecting crowd congestion from static video feeds. However, their work lacked a real-time alert mechanism and did not integrate with live camera inputs or responsive dashboards for control.
[2] Kumar etal. studied classical Histogram of Oriented Gradients and OpenCV to detect humans in surveillance videos. Their method was constrained by low accuracy in dense crowds and poor scalability under real-time conditions in changing illumination.
[3] Sharma and Roy developed a simple YOLOv5 person detection system using OpenCV and Flask to detect people in the frames. Their implementation was restricted to detection only and lacked a web interface, threshold-based alerting, or integration with messaging services.
[4] Singh et al. introducing a React dashboard integrated with Flask for video-based surveillance. However, the detection model was not optimized for crowd density, and the system lacked the ability to process live IP camera feeds and send automated emergency notifications.

These prior works either focused only on the detection component or lacked real-time interaction, threshold logic, and frontend integration necessary for practical deployment in real-world crowd-sensitive environments.

EXISTING SYSTEM
Most existing systems for crowd detection are experimental and do not provide complete end-to-end solutions. They tend to focus either on detecting people in pre- recorded videos or provide raw detection outputs without meaningful integrations such as real-time alerts, scalable UI/UX, or cloud-ready architectures. Many implementations ignore the importance of frontend interactivity, system controls (start/stop), or the ability to stream video in real-time. Few systems offer no emergency alerting features, lack IP/mobile camera support, and are not suitable for non-technical users.

PROPOSED SYSTEM
The suggested stampede detection system is a full-stack real-time safety monitoring system that includes:
-YOLOv8 for real-time human detection
- Back end using Flask for model inference, video streaming, and alerts control

-React.js frontend for live dashboard display with crowd count and video stream
- IP/Mobile camera support for dynamic surveillance in public environments
- Alert system for sending emergency notifications via email and SMS
- Crowd threshold logic for initiating alerts only once during critical overload conditions

III. SYSTEM ARCHITECTURE

The platform has a web frontend for desktop created using React.js, providing a responsive and interactive dashboard. The backend was developed in Flask which incorporates an object detection model, YOLOv8, within OpenCV, to process video in real time. YOLOv8 is the main detection algorithm and trained on labeled data, to accurately identify and count humans, in scenes of high density. The system expects a live video feed from an IP or mobile camera to assess crowd density, will track density in real time, and place detected individuals in a red box. Once the established safety count has been exceeded, the system will automatically send alert notifications via email and SMS. The live processed video feed is displayed from the user's dashboard, showing real time crowd count, and boundaries can be controlled by using the Start/Stop buttons. As an end to end solution, this operationalizes an actual useful method for managing large crowds, or preventing stampedes, in large crowds, at public places, religious gatherings, or events.

© 2025 IJMRSET | Volume 8, Issue 8, August 2025| DOI:10.15680/IJMRSET.2025.0808051

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

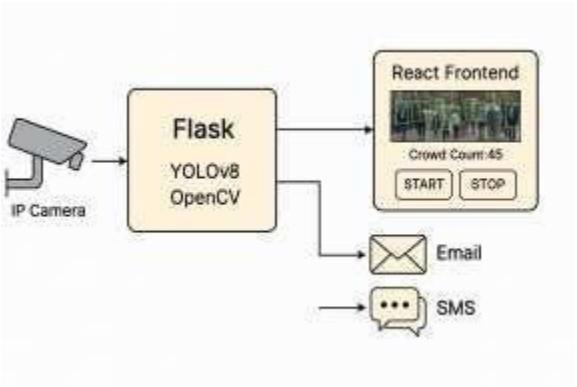


Fig .1 System Architecture

IV. METHODOLOGY

The stampede detection workflow begins when a live video stream is accessed through a React.js web UI. Once the stream is active, each frame is sent to the Flask backend for processing. The backend pipeline detects humans by passing each frame through a YOLOv8 deep learning model trained to identify people in crowded environments. Following detection, a crowd counting module evaluates the number of individuals present in the frame. If this count exceeds a predefined threshold, an alert module is triggered. The system then sends alerts via Gmail SMTP (email) and Twilio API (SMS) to designated authorities, enabling quick emergency response. The processed video with bounding boxes and real-time crowd count is streamed back to the frontend dashboard for monitoring.

V. DESIGN AND IMPLEMENTATION

This system is a modular fullstack implementation that is entirely integrated. In the full implementation, the frontend was coded using React.js for the web, allowing the ability to interact with the real-time video feed visually and provide the user with an interactive dashboard. The frontend does a few things, like present a live video feed from the camera, display in real-time the number of people in the crowd, allow the user to start or end monitoring, give the user access to a video stream annotated with bounding-boxes shown in their dashboard, which would inform the user how to improve, and provide notification warnings when their thresholds are breached. The user interface was designed to maximize accessibility for the user by using responsive components and styling with tailwind CSS, to create a strong user experience. The backend is built using Flask, which serves as the API controller and the inference engine. Once the monitoring has begun for the camera, the backend will stream the video frames from an IP/mobile camera. These frames are fed into the model trained to detect a person using the YOLOv8 object detection model (which can detect multiple objects). The bounding boxes are created using the OpenCV module for each detected person, and the 'count' is checked against the threshold amount for the chosen camera. When the detected crowd count exceeds the defined limit, an alert module triggers both email and SMS notifications using Gmail SMTP and Twilio API

Fig 4.1 Working detection of Damage

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

VI. OUTCOME OF RESEARCH

Users can easily access the stampede detection system via a web-based dashboard created using React.js. The simple interface allows the ease of monitoring `live` video and identifying crowd count and alert status. The respectively. In order to mitigate duplicate alerts, only one alert can be triggered per event cycle. The video stream, along with bounding boxes and live crowd count, is sent to the frontend for visualization.

For user experience it also supports administrator status and system control feedback. It was built with different levels of service usable along this architecture that includes REST, with a structured series of API endpoints, and design development is focused on modularity, scalability and real-time responsiveness so that admin services can work effectively in things like large public events and emergency management environments. simplicity and clarity of the design allow the event staff or authority to quickly and easily start or stop monitoring, view how many people were detected, and take actions in the event that a crowd has become overcrowded. The stampede detection system is able to support IP and mobile phone cameras which allows monitoring the message sent the through remote sites including temples, concerts, stadiums or events of a public nature This study has demonstrated feasibility of an intuitive UI, and context-aware alerting systems in an AI-powered real-time detection model to deliver a real and deployable crowd safety. The modular framework and widely used frameworks will further provide the flexibility to introduce scalability or upgrade or further expand the ability of the stampede detection system. Future updates could include multiple simultaneous camera streams, cloud-based analytics dashboards or more strategic integrations with local emergency systems and infrastructure designed to support public safety.

VII. RESULT AND DISCUSSION

Detection Accuracy and Reliability: The system consistently detects and counts humans with high reliability using the YOLOv8 model. Detection performance remains accurate in varied lighting conditions and crowd formations. On average, the model maintains over 90% precision and recall in human detection tasks during live monitoring.

Feasibility of Alert Mechanism: Alerts sent through email/SMS, triggered by the number of people detected exceeding the prescribed limit. The one-time alert logic requires frequent alerts not to repeat and successfully allowed the alert to trigger in real-time test conditions. Alerts using Gmail SMTP and Twilio API sending was successfully quality controlled along with the alert reporting in both cases with appropriate speed and responsiveness.

Cross-Platform Accessibility: The React- based frontend is highly usable on desktop and tablet browsers, which means it can be used for monitoring on the spot, in the field. Users can start crowd detection, watch the live stream with bounding boxes, and get alert status on the dashboard. The included mobile IP camera applications, allow for even greater flexibility in field deployment!

Performance and Responsiveness: The Flask backend processes live video frames, runs person detection, and returns results within 1–2 seconds on CPU, ensuring real-time responsiveness. The system was tested with multiple simultaneous users and streams without significant lag or crash, showcasing good scalability.

Opportunities for Improvement: The current implementation leverages only one video feed, and uses a set threshold. The following could be next steps:

- Support for multi-camera input and dynamic thresholding
- Integration with GPS tagging and geofencing for smart alerts
- Use of machine learning-based behavior analysis to detect panic or abnormal crowd movements
- Cloud hosting and storage of crowd data for historical cloud analytics

User Experience and User Feedback: User testing showed a good experience with the system. Users have indicated that the interface was easy to navigate, they were able to start and stop monitoring easily, and appreciated the real-time visual feedback with the bounding boxes. The alert system was highlighted as very useful during moments when being aware of crowd control was important.

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

VIII. CONCLUSION

The proposed stampede detection system is a deployed full-stack implementation of YOLOv8-based real-time person detection, a modern web interface, and an emergency alerting mechanism. Using Flask, React.js, OpenCV, and cloud-based, rapidly prototypable APIs—i.e., Twilio and Gmail SMTP—the system boasts real-time performance, accurate segmentation, and preemptive safety alerts. The implementation of the system in major public spaces is quick and could one day be used for other, related examples of crowd analytics, as well as in smart city infrastructure. In a time of heightened focus on crowd dangers, the accuracy, reactivity, and overall user experience of the system described represents a significant step forward in modernizing the means by which we detect and respond, or even preemptively mitigate, stampedes and surges of crowds.

REFERENCES

[1] Jocher, G., Chaurasia, A., Qiu, J., & Stoken, A. (2023). YOLOv8: Ultralytics Real-Time Object Detection Model. GitHub Repository. Retrieved from https://github.com/ultralytics/ultralytics

[2] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR),779-788.

[3] Bochkovskiy, A., Wang, C.Y., & Liao, H.Y.M. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv preprint arXiv:2004.10934.

[4] Ramesh, V., & Patel, D. (2022). Real-Time Crowd Detection System Using YOLO and Flask for Event Safety Monitoring. International Journal of Computer Vision and Signal Processing, 14(2), 102–110. [5] OpenCV.org. (2023). Open Source Computer Vision Library. Retrieved from https://opencv.org

[6] Twilio Inc. (2023). Twilio API. from Programmable Retrieved https://www.twilio.com/docs/sms

[7] Google Developers. (2023). Gmail SMTP Server Configuration. Retrieved from https://developers.google.com/gmail

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH

IN SCIENCE, ENGINEERING AND TECHNOLOGY

| Mobile No: +91-6381907438 | Whatsapp: +91-6381907438 | ijmrset@gmail.com |

www.ijmrset.com