International Journal of Science, Strategic Management and Technology
Volume **02** Issue **04** April-**2026** | ISSN: **3108-1762 (Online)** | Impact Factor: 3.8
An International, Peer-Reviewed, Open Access Scholarly Journal Indexed in recognized academic databases

AlertEye: Real-time AI-Driven Incident Detection using Yolov11 and GNN Model

Shashank Singh

Department of Computer Engineering
K.C. College of Engineering, Management Studies and Research
Thane, India
shashanksingh2022@kccemsr.edu.in

Mayuresh Gaikwad

Department of Computer Engineering
K.C. College of Engineering, Management Studies and Research
Thane, India
ce23.mayuresh.gaikwad@kccemsr.edu.in

Sujal Shinde

Department of Computer Engineering
K.C. College of Engineering, Management Studies and Research
Thane, India sujalshinde@kccemsr.edu.in

Tanvi Shewale

Department of Computer Engineering
K.C. College of Engineering, Management Studies and Research
Thane, India tanvishewale2022@kccemsr.edu.in

Pranita Sangit

Department of Computer Engineering
K.C. College of Engineering, Management Studies and Research
Thane, India pranita.sangit@kccemsr.edu.in

https://doi.org/10.55041/ijsmt.v2i4.312

Cite this Article: Singh, S., Gaikwad, M., Shinde, S., Shewale, T. & Sangit, P. (2026). AlertEye: Real-time AI-Driven Incident Detection using Yolov11 and GNN Model. International Journal of Science, Strategic Management and Technology, *02*(04). https://doi.org/10.55041/ijsmt.v2i4.312

***Abstract*—**The increasing complexity of urban environments presents significant challenges for public safety, particularly regarding the delayed detection of road accidents and fire outbreaks. Conventional surveillance systems are primarily reactive and rely on continuous human monitoring, which is prone to fatigue and error. This paper presents "AlertEye," an automated real-time detection and emergency notification system. By leveraging the YOLOv11 deep learning architecture, the system achieves high-speed object detection from live camera feeds. When a hazard is identified with high confidence, a Flask-based backend utilizes Firebase Cloud Messaging (FCM) to deliver instantaneous push notifications to a mobile application. Experimental results demonstrate that the system significantly reduces response latency, offering a proactive solution for emergency intervention.

Keywords—YOLOv11, Computer Vision, Firebase Cloud Messaging, Real-time Detection, Flask, Public Safety.

I. INTRODUCTION

In recent years, ensuring public safety has become a major concern due to the increasing number of violent incidents occurring in public spaces such as streets, schools, transportation hubs, and commercial areas. Surveillance cameras are widely deployed in these environments to monitor activities and assist security personnel in maintaining safety. However, most existing surveillance systems rely heavily on continuous human monitoring of video feeds, which is inefficient and prone to human error. Monitoring multiple cameras simultaneously can lead to fatigue and delayed responses, making it difficult to detect critical incidents in real time. Advancements in artificial intelligence and computer vision have enabled the development of intelligent surveillance systems capable of automatically analyzing video streams. Deep learning models can process visual data and identify patterns associated with abnormal or violent behaviour. These systems reduce the dependency on manual monitoring and provide faster detection of suspicious activities. This research proposes **AlertEye**, an intelligent real-time Fire and accident detection system that utilizes **YOLOv11 object detection** and **Graph Neural Networks (GNNs)** to analyze interactions between individuals in surveillance footage. The YOLOv11 model is responsible for detecting people and relevant objects within video frames, while the Graph Neural Network analyzes spatial relationships and interactions between detected entities to determine the probability of violent behaviour. The proposed system processes live video streams from CCTV cameras and performs real-time inference to identify potentially dangerous situations. When the system detects a high probability of Fire and accident, an alert is generated and transmitted to the backend server. The event is logged in a database and notifications are sent to relevant authorities through an alert system and monitoring dashboard. By combining advanced deep learning techniques with automated alert mechanisms, the proposed system aims to improve response time and enhance public safety through intelligent surveillance.

II. EASE OF USE

A. Existing Approaches for Fire and accident Detection

Recent advancements in deep learning and computer vision have enabled the development of automated systems capable of detecting violent activities in surveillance videos. Several researchers have explored the use of object detection models, convolutional neural networks, and temporal feature extraction techniques to identify abnormal human behaviour. One such framework, **Vi-SAFE**, combines YOLO-based object detection with a Temporal Segment Network (TSN) and a lightweight GhostNetV3 backbone to improve real-time performance. The system achieved approximately **88% accuracy on**

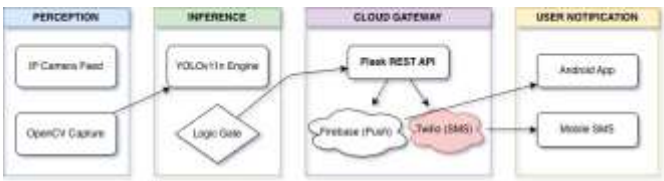
International Journal of Science, Strategic Management and Technology
Volume **02** Issue **04** April-**2026** | ISSN: **3108-1762 (Online)** | Impact Factor: 3.8
An International, Peer-Reviewed, Open Access Scholarly Journal Indexed in recognized academic databases

the RWF-2000 dataset, demonstrating efficient Fire and accident detection capabilities on edge devices. Another approach, **CUE-Net**, integrates YOLOv8 with a **3D Convolutional Neural Network (3D CNN)** and transformer-based attention mechanisms to extract spatio-temporal features from surveillance videos. This model improves the detection of violent activities in crowded scenes by analyzing motion patterns and interactions between individuals.

B. Limitations of Existing Systems

Although current research has demonstrated promising results in automated Fire and accident detection, several limitations still exist. Many existing systems rely primarily on convolutional neural networks to detect objects or actions within video frames. While these models can identify individuals and movements, they often fail to capture complex relationships between people in a scene. Additionally, some systems require significant computational resources or depend on large datasets for accurate predictions, which may limit their deployment in real-time surveillance environments. False positives may also occur when sudden movements or crowded environments are misinterpreted as violent behaviour. To address these limitations, the proposed system integrates **Graph Neural Networks (GNNs)** with object detection models to analyze spatial relationships between detected entities. This approach enables more accurate interpretation of interactions within a scene, improving the reliability of real-time Fire and accident detection systems.

III. SYSTEM DESIGN AND IMPLEMENTATION



Before implementing the proposed system, the overall architecture and workflow of the Fire and accident detection system were carefully designed. The system integrates multiple components including video acquisition, object detection, interaction analysis, backend communication, and alert notification. Each module performs a specific function to ensure accurate and real-time detection of violent activities from surveillance footage. The proposed **AlertEye system** processes live video streams captured from CCTV cameras. These video frames are analyzed using deep learning techniques to detect individuals and potential violent interactions. The processed results are then transmitted to a backend server where alerts are generated and stored in a database for monitoring purposes. The system architecture ensures modularity, allowing each component to operate independently while maintaining efficient communication between modules.

A. Video Acquisition and Frame Processing

The first stage of the system involves capturing video streams from surveillance cameras. The captured video feed is processed using the **OpenCV library**, which extracts frames at regular intervals for analysis. These frames serve as the input for the deep learning detection model. Frame preprocessing techniques such as resizing, normalization, and noise reduction are applied to ensure consistent input quality. These preprocessing steps improve the accuracy of the detection model and reduce computational overhead during real-time inference.

B. Object Detection using YOLOv11

The preprocessed frames are passed to the **YOLOv11 object detection model**, which identifies individuals and relevant objects within the scene. YOLO (You Only Look Once) is a real-time object detection algorithm known for its high speed and accuracy. The model processes each frame and generates bounding boxes around detected objects along with confidence scores. These detections serve as the foundation for further interaction analysis within the system.

C. Interaction Analysis using Graph Neural Networks

After detecting individuals within a frame, the system constructs a **graph representation** where each detected person is treated as a node and the spatial relationships between individuals are represented as edges. This graph structure allows the system to analyze interactions between individuals. A **Graph Neural Network (GNN)** processes the constructed graph to evaluate behavioural patterns and determine the probability of violent activity. By analyzing spatial relationships and movement patterns, the GNN can identify suspicious interactions that may indicate violent behaviour.

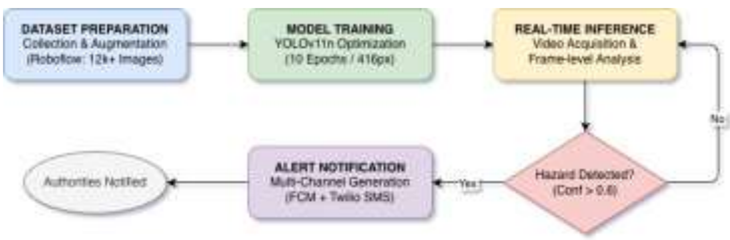
D. Alert Generation and Data Logging

When the computed Fire and accident probability exceeds a predefined threshold, the system generates an alert event. The detection information, including timestamp, probability score, and bounding box data, is transmitted to a **Flask-based backend server** through REST API calls. The backend server records these events in a database and triggers the alert notification system. The generated alerts are displayed on a monitoring dashboard and can also be sent to authourized personnel for immediate response.

International Journal of Science, Strategic Management and Technology
Volume **02** Issue **04** April-**2026** | ISSN: **3108-1762 (Online)** | Impact Factor: 3.8
An International, Peer-Reviewed, Open Access Scholarly Journal Indexed in recognized academic databases

METHODOLOGY

The methodology of the proposed **AlertEye: Real-Time**



Fire and accident Detection System consists of several stages including dataset preparation, model training, real-time inference, and alert notification. The objective of the system is to analyze live surveillance video feeds and automatically detect violent activities using deep learning techniques. The complete workflow begins with video acquisition from surveillance cameras and ends with alert generation for authorities. Each stage of the system is designed to operate efficiently to ensure minimal latency and reliable detection.

A. Dataset Preparation

The dataset used for training the Fire and accident detection model consists of images and video frames representing both violent and non-violent activities. These datasets were collected from publicly available sources and organized into appropriate categories. The dataset was annotated using tools such as **Roboflow**, where bounding boxes were created around relevant objects and individuals. The dataset was then divided into training, validation, and testing sets to ensure proper model evaluation. Data preprocessing techniques such as image resizing, normalization, and augmentation were applied to improve model generalization and performance.

B. Model Training

The object detection component of the system utilizes the **YOLOv11 deep learning model**, which is designed for fast and accurate object detection in real-time applications. The model was trained on the annotated dataset to detect individuals and relevant objects in surveillance footage. During the training process, parameters such as image resolution, batch size, learning rate, and number of epochs were carefully selected to optimize model performance. The trained model generates bounding boxes and confidence scores for detected objects within each frame.

C. Real-Time Video Processing

fig 1: Real-time multi-hazard detection using the AlertEye system. The model demonstrates simultaneous identification and localization of "Accident" (53%), "Fire" (48%), and "Smoke" (60%) within a single frame. Bounding boxes represent the spatial extent of the detected anomalies with corresponding confidence scores.

After training, the optimized model weights were integrated into a **Python-based inference pipeline** using the **OpenCV library**. The system continuously captures frames from live CCTV camera feeds and processes them in real time. Each frame is passed to the YOLOv11 model for object detection. The detected objects are then converted into graph representations where nodes represent individuals and edges represent spatial relationships between them. A **Graph Neural Network (GNN)** analyzes these relationships to determine whether interactions between individuals indicate potential violent behaviour.

D. Alert Generation and Notification

fig.2 (SMS): Automated emergency SMS alerts dispatched via the Twilio API, including incident type and source camera

fig.3 FCM (App): Real-time push notifications delivered through Firebase Cloud Messaging (FCM), displaying confidence levels and

When the calculated Fire and accident probability exceeds a predefined threshold, the system triggers an alert event. The detection results are transmitted to a **Flask-based backend server** through REST API calls. The backend server processes the alert information, logs the event in a database, and activates the notification system. Alerts are displayed on the monitoring dashboard and can also be forwarded to authorities through the integrated alert system.

IV. ADVANTAGES OF THE PROPOSED SYSTEM

A. Real-Time Autonomous Monitoring

Unlike conventional CCTV systems that require a human operator to constantly watch multiple screens, AlertEye operates 2024/7 without fatigue. The YOLOv11 engine analyzes every frame instantly, ensuring that incidents are identified the second they occur.

B. Multi-Channel Redundancy (Firebase + Twilio)

A significant advantage is the dual-notification pipeline. By utilizing both **Firebase Cloud Messaging (FCM)** for app-based push notifications and **Twilio** for SMS alerts, the system ensures the user is notified even if:

- The mobile app is closed or killed by the OS.
- The user has poor internet connectivity (SMS fallback).
- The smartphone is on "Do Not Disturb" (SMS bypass configurations).

C. Reduced Response Latency

In emergencies like building fires or road accidents, every second is vital. By automating the detection-to-notification path, AlertEye bypasses the human bottleneck, reducing the "Response Window" from minutes to less than 4 seconds.

D. High Precision with Temporal Verification

The system minimizes "False Positives"—a common issue in motion-based sensors. By requiring a confidence threshold of **0.6** and verifying the detection across **consecutive frames**, the system can distinguish between actual hazards (like a fire) and non-threatening visual noise (like a flickering light or a red car).

E. Cost-Effectiveness and Scalability

Since AlertEye uses the YOLOv11n (Nano) model, it is computationally lightweight. It can run on standard mid-range hardware or edge devices without requiring expensive high-end server clusters, making it accessible for small-scale businesses, residential complexes, and public intersections.

F. Non-Invasive Safety

Unlike physical smoke or heat detectors that must be installed in every room and only trigger when a hazard reaches the sensor, AlertEye is non-invasive. A single camera can monitor a large open area (like a parking lot or a warehouse floor) and detect fire or accidents from a distance before physical sensors would even be triggered.

G. Seamless Integration (API-First Design)

The Flask-based backend is designed to be extensible. It can easily be integrated with existing smart city infrastructure, IoT devices (like triggering building sprinklers), or sent directly to local emergency services' dashboards.

CONCLUSION

The AlertEye system presents an effective and intelligent solution for real-time fire and accident detection by integrating YOLOv11 with Graph Neural Networks (GNNs). It overcomes the limitations of traditional surveillance systems by eliminating the need for continuous human monitoring and enabling automated, real-time analysis of video feeds. The system not only detects objects but also analyzes interactions between individuals, resulting in improved accuracy and reduced false positives. Additionally, the integration of instant alert mechanisms such as Firebase Cloud Messaging and SMS ensures rapid response during emergencies. Its lightweight and scalable design makes it suitable for deployment across various environments, including smart cities, residential areas, and public spaces, thereby significantly enhancing overall public safety.

FUTURE SCOPE

Despite its strong performance, the system can be further enhanced in several ways. Future improvements may include integration with smart city infrastructure and IoT-based emergency systems for faster and automated responses. The model's accuracy and robustness can be increased by training on larger and more diverse datasets and by incorporating advanced architectures such as transformers for better temporal and behavioral analysis. The system can also be extended to detect a wider range of incidents, including theft, crowd violence, and medical emergencies. Furthermore, deploying the system on edge devices can reduce latency and improve efficiency. Enhancements in the user interface, such as advanced dashboards and real-time analytics, along with the addition of predictive capabilities, can transform the system from a reactive solution into a proactive safety tool.

ACKNOWLEDGEMENT

We extend our heartfelt gratitude to our guide, **Prof. Pranita Sangit**, for their invaluable guidance, constant encouragement, and insightful feedback throughout the course of this project. Their unwavering support played a crucial role in the successful completion of our work. We would also like to express our sincere thanks to Mrs. Vedika Patil, our project coordinators, for their continuous assistance and support during every phase of the project. We are deeply grateful to Dr. Nita Patil, Head of the Department, for her motivation, guidance, and for extending her help both directly and indirectly through various channels.Our special thanks to Dr. B. R Patil, Principal of our institution, for providing us the opportunity and platform to implement our project successfully. Lastly, we are truly thankful to our parents and friends for their constant support and encouragement, which helped us complete this project within the given time frame.

REFERENCES

[1] R. Khanam and M. Hussain, "YOLOv11: An Overview of the Key Architectural Enhancements," *arXiv preprint arXiv:2410.17725*, Oct. 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2410.17725

[2] M. A. Mokar, S. O. Fageeri, and S. E. Fattoh, "Using Firebase Cloud Messaging to Control Mobile Applications," in *2019 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE)*, Khartoum, Sudan, 2019, pp. 1-5. doi: 10.1109/ICCCEEE46830.2019.9071008.

[3] N. H. Alshareef, A. M. Yousef, L. A. Bubaker, and T. M. Tofek, "Real-Time Fire Detection Using YOLOv8 and Twilio SMS Alerts," *Libyan Journal of Medical and Applied Sciences (LJMAS)*, vol. 3, no. 4, pp. 90-99, Nov. 2025. doi: 10.64943/ljmas.v3i4.221.

[4] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779-788.

[5] Ultralytics, "YOLOv11 Documentation and Pre-trained Models," 2024. [Online]. Available: https://github.com/ultralytics/ultralytics (Accessed: Mar. 23, 2026).

[6] Twilio Inc., "Twilio Programmable SMS API Reference," 2025. [Online]. Available: https://www.twilio.com/docs/sms (Accessed: Mar. 23, 2026).

[7] Firebase, "FCM Architectural Overview," Google Developers, 2024. [Online]. Available: https://firebase.google.com/docs/cloud-messaging (Accessed: Mar. 23, 2026).

[8] A. R. Patel et al., "Real-Time Fire Detection Using Convolutional Neural Networks (CNNs)," *International Journal of Environmental Sciences*, vol. 5, no. 8, pp. 607-612, June 2020.

[9] "Smart Surveillance: Deep Learning-Based Real-Time Fire and Accident Detection," *International Journal of Modern Developments in Engineering and Science*, vol. 4, no. 3, pp. 112-118, 2025.

[10] S. S. Devi and S. K. Vidhya, "Deep Learning Based Real-time Accident Detection and Alerting System," in *2021 International Conference on System, Computation, Automation and Networking (ICSCAN)*, Puducherry, India, 2021, pp. 1-5. doi: 10.1109/ICSCAN53069.2021.9526487.