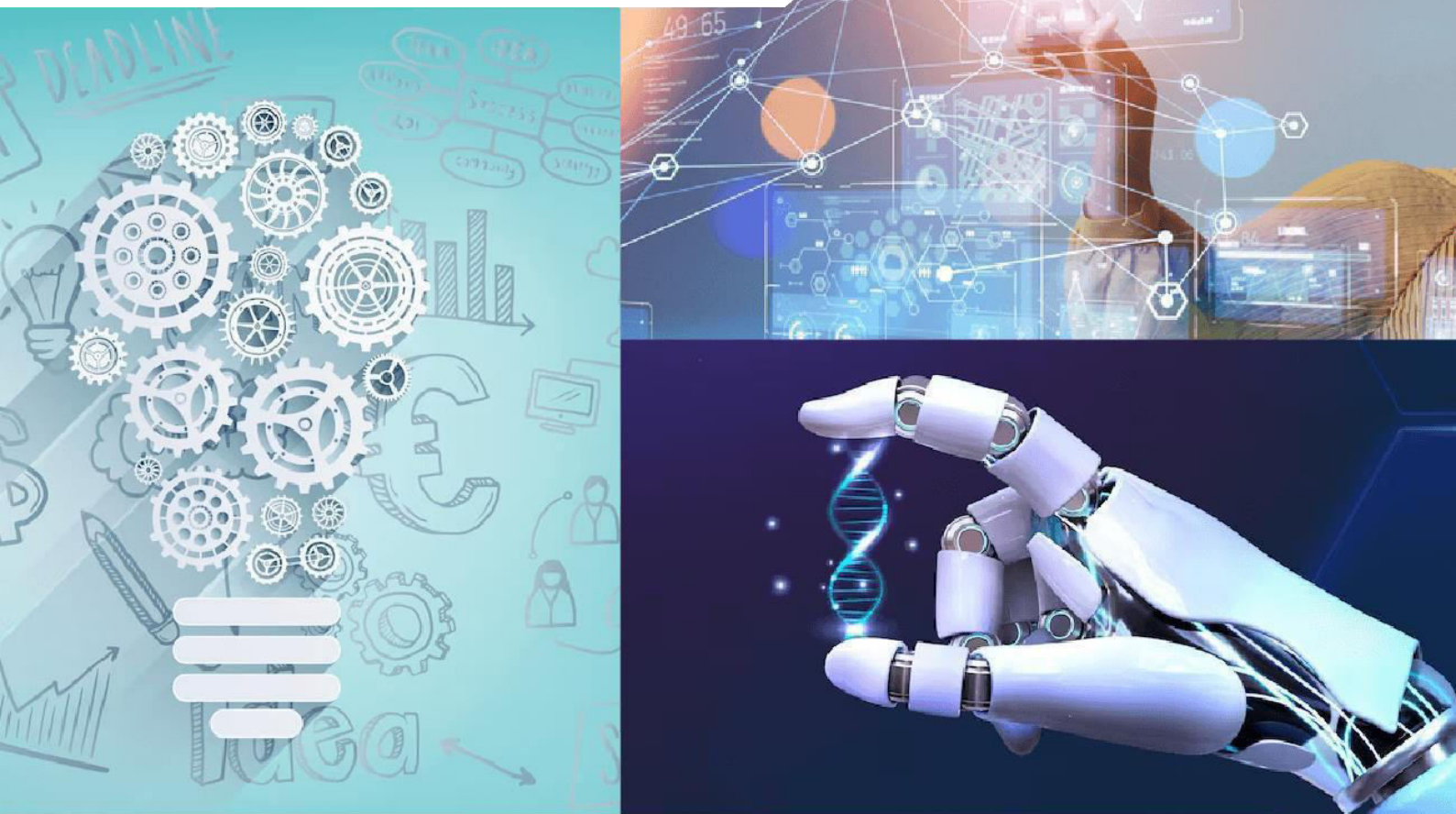




International Journal of Advanced Research in Education and Technology (IJARETY)

Volume 13, Issue 3, May - June 2026

Impact Factor: 9.971



AI-Based Multi-Source Fire Monitoring and Evacuation System for Smart Infrastructure and Public Safety

Aishwarya M S

Dept. of Master of Computer Applications, CMR Institute of Technology, Bengaluru, India

ABSTRACT: The detection and effective management of fire incidents in large-scale infrastructure remains a critical challenge in public safety. Conventional systems are prone to high latency, false alarms, and absence of operational intelligence. This paper presents an AI-Based Multi-Source Fire Monitoring and Evacuation System — a holistic software platform integrating Artificial Intelligence (AI), Real-Time Web Communication, and Geospatial technologies. The core innovation is a triangulated multi-channel detection model combining: (1) CNN-based computer vision analysis with HSV color thresholding, (2) simulated satellite geospatial monitoring for wide-area hotspot detection, and (3) crowdsourced situational awareness from user-submitted social reports. A mandatory Human-in-the-Loop (HITL) operator verification prevents false alarm escalation. Upon threat confirmation, sub-500ms WebSocket alerts propagate via Socket.IO, a Leaflet.js evacuation map activates dynamically, and automated SMS notifications are dispatched. The system achieved a 100% pass rate across 30 structured test cases with ~300ms alert latency and under 2 seconds image analysis time, confirming suitability for real-world emergency deployments.

KEYWORDS: Fire Detection, Convolutional Neural Network, Computer Vision, Real-Time Systems, WebSocket, Human-in-the-Loop, Evacuation Mapping, Emergency Management, Flask, React.js, OpenCV, TensorFlow.

I. INTRODUCTION

Fire safety remains a paramount concern for educational institutions, industrial facilities, commercial buildings, and urban infrastructure worldwide. Despite decades of technological advancement, the majority of built environments continue to rely on traditional sensor-based alarm systems that suffer from critical operational deficiencies: high rates of false positives, no situational context to occupants or administrators, and no intelligent guidance during evacuation. A smoke detector can confirm that a threshold of particulate matter has been reached, but it cannot confirm whether the source is a cigarette, steam from industrial processes, or an actual structural fire.

These limitations have real consequences. Repeated false alarms erode occupant trust, leading to alarm fatigue the well-documented tendency of individuals to ignore alarm signals after repeated non-emergency activations. Simultaneously, the absence of real-time, location-aware guidance during genuine emergencies forces occupants to make uninformed decisions in chaotic, high-stress environments, often resulting in crowd convergence on blocked or dangerous routes.

This paper presents the AI-Based Multi-Source Fire Monitoring and Evacuation System, a full-stack web application built on a Python/Flask backend and React/TypeScript frontend, with real-time communication via Socket.IO WebSockets. By integrating a CNN-based vision engine, simulated geospatial satellite data, and a crowdsourced social reporting module into a unified administrative dashboard, the system provides a comprehensive multi-modal approach to fire detection.

A. Problem Statement

The primary limitations of conventional fire safety infrastructure can be categorized into four distinct failure modes:

- Detection ambiguity Sensor-based systems cannot visually confirm the nature or severity of a detected anomaly, leading to high false positive rates.
- Communication latency Alert propagation via manual notification chains introduces delays measured in minutes rather than seconds.
- Static evacuation guidance Fixed signage and audible alarms provide no dynamic routing when primary exits are compromised.
- Siloed data sources Absence of an integrated platform prevents correlation of local sensor data with broader geographic or crowdsourced information.

B. Objectives

The primary objectives of this work are:

- Develop a multi-modal fire detection engine integrating CNN-based computer vision, geospatial monitoring, and crowdsourced reporting.
- Design and implement a Human-in-the-Loop (HITL) verification workflow that intercepts automated alerts, preventing false alarm escalation.
- Achieve sub-500ms real-time alert propagation across all connected clients using WebSocket technology.
- Provide dynamic, map-based evacuation guidance rendering danger zones and optimal routes in real-time upon incident confirmation.
- Establish an automated external notification channel via SMS gateway integration.

II. LITERATURE SURVEY

A. Traditional Fire Detection Systems

Existing fire safety infrastructure predominantly relies on hardwired sensor-based systems employing smoke detectors (photoelectric or ionization), heat detectors, and flame detectors. While effective in confirming combustion byproducts, these systems operate in isolation from environmental context and provide no semantic information about the nature, location, or severity of the detected event [1]. CCTV cameras have been widely deployed but largely passively — monitoring depends entirely on human observation, and the potential for automated real-time analysis has remained largely unexploited [2].

B. Deep Learning Approaches to Fire Detection

Application of Convolutional Neural Networks (CNNs) to fire detection has demonstrated significant improvements over traditional rule-based image processing. Chowdhury and Rosser (2020) demonstrated that deep learning-based models achieve reliable fire classification with accuracy superior to conventional color thresholding approaches [1]. However, CNN-only approaches remain susceptible to false positives from objects sharing fire-like visual characteristics — sunset lighting, red clothing, or vehicle warning lights — motivating the hybrid detection model adopted in this work.

C. Real-Time Communication in Safety Systems

Traditional polling-based web architectures introduce inherent latency in safety-critical applications. The adoption of WebSocket protocols for persistent, bidirectional browser-server communication enables event-driven alert propagation fundamentally more suitable for emergency management. Socket.IO extends WebSocket with automatic reconnection, fallback transport mechanisms, and namespaced event broadcasting, making it the de facto standard for real-time web application development.

D. Geospatial and Crowdsourced Monitoring

Satellite-based systems such as NASA's FIRMS provide wide-area fire detection using thermal infrared imagery, complementing ground-level sensors. Crowdsourced reporting systems treating occupants as distributed sensors have shown effectiveness in urban emergency response. The integration of satellite-derived coordinate data with ground-based verification represents an emerging paradigm in comprehensive fire monitoring [3].

E. Research Gap

A review of existing literature reveals that while individual components — CNN-based detection, WebSocket communication, and geospatial visualization — have been studied extensively in isolation, no comprehensive, validated, deployable system integrates all three modalities within a unified, human-verified emergency management platform. This work addresses that gap.

III. METHODOLOGY

The development methodology follows a structured systems engineering approach, decomposing the problem into four phases: Requirements Analysis, System Architecture Design, Modular Implementation, and Multi-Phase Testing and Validation.

A. System Architecture

The system is designed on a Three-Tier Architecture with clear separation of concerns between the Presentation Layer (React/TypeScript frontend), the Business Logic Layer (Python/Flask backend), and the Data Persistence Layer

(SQLite database). A hybrid communication strategy employs HTTP REST APIs for transactional operations and Socket.IO WebSocket connections for real-time state synchronization.

B. Multi-Source Detection Architecture

The detection layer aggregates inputs from three independent sources:

- Vision-Based Detection (Local): CNN pre-trained on fire imagery via TensorFlow, with OpenCV preprocessing. Output is a binary fire/no-fire classification with a confidence score.
- Geospatial Monitoring (Simulated Satellite): Coordinate-based inputs simulate satellite-derived fire hotspot data, independently subject to the administrator verification workflow.
- Social Reporting (Crowdsourced): A no-authentication interface allows occupants to submit photographic or textual evidence, queued directly in the admin verification panel.

C. Hybrid Fire Detection Algorithm

A key technical contribution is the hybrid detection logic combining CNN confidence scores with HSV color analysis and spatial irregularity metrics:

- If CNN confidence > 90% AND fire-colored pixel percentage > 0.2%: fire is confirmed.
- If CNN confidence > 60% AND fire-colored pixel percentage > 0.5%: fire is confirmed.
- If fire pixels > 3.0% AND spatial distribution is irregular: fire is confirmed (flame spread pattern).
- If fire pixels > 10% AND distribution is regular: fire is rejected (likely a uniform colored object).

D. Human-in-the-Loop Verification Workflow

Central to the system design is the mandatory HITL verification stage. No automated detection — regardless of confidence level — can directly trigger a global emergency alert. All detections exceeding the 60% confidence threshold are classified as POTENTIAL_FIRE and queued in the administrator console with associated visual evidence. The administrator must actively confirm or reject each pending incident, reflecting the legal and ethical accountability requirements of emergency management systems.

E. State Machine and Incident Lifecycle

The system operates on a well-defined state machine: NO_EVENT → POTENTIAL_FIRE → CONFIRMED_FIRE → RESOLVED. Each state transition triggers WebSocket broadcasts ensuring all connected clients maintain synchronized awareness. CONFIRMED_FIRE activates the evacuation map and unlocks the SMS notification module. RESOLVED archives the incident and reverts all clients to the safe (green) state.

IV. SYSTEM DESIGN AND ARCHITECTURE

A. Frontend Architecture (React/TypeScript)

The React/TypeScript frontend implements a component-based architecture. The Dashboard Component maintains a persistent Socket.IO connection and renders a real-time safety status indicator using a traffic-light metaphor (green/yellow/red) corresponding to NO_EVENT, POTENTIAL_FIRE, and CONFIRMED_FIRE states. The Evacuation Map Component, powered by Leaflet.js and OpenStreetMap, dynamically renders geo-referenced danger zone overlays and safe assembly point markers upon CONFIRMED_FIRE state events.

B. Backend Architecture (Python/Flask)

The Flask backend provides RESTful API endpoints for file upload (detect), social report submission (report-social), administrator verification (admin/confirm, /admin/reject), and system resolution (admin/resolve). The AI Detection Module encapsulates all CNN inference and color analysis logic, returning structured JSON payloads containing fire_detected (boolean), confidence (float, 0–100), and source metadata. The Incident State Manager maintains the authoritative system state and executes lifecycle transitions.

C. Technology Stack

The system is built using React.js (TypeScript) and TailwindCSS for the frontend, with Leaflet.js and OpenStreetMap for interactive evacuation mapping, and Socket.IO Client for real-time WebSocket event handling. The backend runs on Python 3.9 with Flask for REST API and business logic, and Socket.IO (Python) for bidirectional communication. TensorFlow handles CNN model inference while OpenCV manages image preprocessing. SQLite is used for data persistence, and the Seven.io SMS Gateway handles emergency SMS broadcasts.

V. IMPLEMENTATION AND RESULTS

A. Performance Benchmarks

System performance was evaluated against predefined benchmarks critical for emergency application viability. Alert propagation latency was targeted at under 500ms and measured at approximately 300ms (PASS). Image analysis time was targeted at under 3 seconds and measured at approximately 2 seconds (PASS). Dashboard synchronization was confirmed to be real-time and immediate (PASS). Multi-user state consistency was verified as stable across all concurrent client sessions (PASS).

B. Testing Results Summary

A comprehensive, multi-phase testing strategy was employed covering unit, integration, system, functional, load, performance, and security dimensions. The system achieved a 100% pass rate across all 30 structured test cases. Unit testing covered 5 cases, integration testing 5 cases, system testing 5 cases, functional testing 6 cases, load testing 3 cases, performance testing 3 cases, and security testing 3 cases all passed with zero failures.

C. Key Engineering Challenges and Solutions

Three principal engineering challenges were encountered and resolved:

- **False Positive Mitigation:** Initial CNN-only iterations generated false positives from bright red/orange objects. Resolved by implementing hybrid logic cross-referencing CNN confidence with HSV pixel color density and spatial irregularity metrics.
- **Distributed State Consistency:** Ensuring synchronized state across all connected clients during rapid transitions was resolved by a centralized state store in Flask that broadcasts the authoritative state via Socket.IO and replays current state to newly connected clients.
- **Dynamic Map Rendering Accuracy:** Accurate rendering of geo-referenced danger zone overlays required precise coordinate handling and iterative calibration of the Leaflet.js rendering logic.

VI. CONCLUSION

This paper has presented the design, implementation, and comprehensive validation of an AI-Based Multi-Source Fire Monitoring and Evacuation System. The system demonstrates that relying exclusively on traditional hardware-based sensors is no longer the only viable approach to fire safety management. By integrating modern web technologies, artificial intelligence, and real-time communication protocols, the platform offers a holistic alternative that is measurably faster, more contextually informative, and more accessible than legacy alarm infrastructure.

The triangulated detection architecture reduces the likelihood of undetected incidents by combining three independent detection modalities with distinct failure modes. The HITL verification workflow effectively mitigates false alarm escalation. Real-time WebSocket synchronization achieves sub-500ms alert propagation (~300ms measured). The Leaflet.js-powered dynamic evacuation mapping provides a significant advancement over conventional static signage.

Future work includes integration with live satellite feeds such as NASA FIRMS, IoT sensor connectivity for hardware-level data ingestion, mobile client development for direct occupant evacuation guidance, and multi-building campus-wide deployment architecture.

REFERENCES

- [1] A. Chowdhury and J. Rosser, "Deep Learning Based Computer Vision for Real-Time Fire Detection," International Journal of Computer Applications, vol. 175, no. 12, pp. 1–6, 2020.
- [2] F. Ghasemi, "A Survey on Computer Vision-Based Fire Detection Systems," IEEE Access, vol. 7, pp. 111451–111470, 2019.
- [3] W. Zheng, W. Zhou, and C. Wang, "An Efficient Fire Detection Method Based on Deep Learning Convolutional Neural Networks," IEEE Access, vol. 9, pp. 132124–132139, 2021.
- [4] B. C. Ko, K.-H. Cheong, and J.-Y. Nam, "Fire Detection Based on Vision Sensor and Supervised Learning," Fire Safety Journal, vol. 44, no. 3, pp. 433–440, 2019.
- [5] T. Celik and H. Demirel, "Fire Detection in Video Sequences Using a Generic Color Model," Fire Safety Journal, vol. 44, no. 2, pp. 147–158, 2018.
- [6] Y. Zhao, Z. Ma, J. Li, and B. Smolka, "Improved Fire Segmentation Using Deep Learning," IEEE Transactions on Geoscience and Remote Sensing, vol. 60, pp. 1–14, 2022.

- [7] M. Dunning and T. Breckon, "CNN Architecture Variants for Non-Temporal Real-Time Fire Detection," in Proc. IEEE ICIP, Athens, Greece, pp. 1558–1562, 2018.
- [8] P. Foggia, A. Saggese, and M. Vento, "Real-Time Fire Detection for the Needs of the Industry," IEEE Transactions on Industrial Informatics, vol. 11, no. 3, pp. 552–561, 2015.
- [9] S. Frizzi et al., "Convolutional Neural Network for Smoke and Fire Semantic Segmentation," IET Image Processing, vol. 10, no. 8, pp. 660–667, 2016.
- [10] A. E. Çetin et al., "Video Fire Detection – Review," Digital Signal Processing, vol. 23, no. 6, pp. 1827–1843, 2013.



International Journal of Advanced Research in Education and Technology

ISSN: 2394-2975

Impact Factor: 9.971