International Journal of Informatics and Communication Technology (IJ-ICT)
Vol. 15, No. 2, June 2026, pp. 861~868
ISSN: 2252-8776, DOI: 10.11591/ijict.v15i2.pp861-868

Advanced IoT-integrated real-time fire detection and automated mitigation system

Rama Krishna Peddarapu, Ajimera Abhinav, Gnana Sathwika V. N. V., Poosa Brijesh, Amrutha Varshini Ravula
Department of Computer Science and Engineering, VNR Vignana Jyothi Institute of Engineering and Technology, Hyderabad, India

Article Info

Article history:

Received Apr 19, 2025
Revised Dec 3, 2025
Accepted Dec 14, 2025

Keywords:

Automated mitigation
Emergency response
Fire detection
Internet of things
Real time monitoring

ABSTRACT

In the field of industry and commerce safety, tackling the most challenging and ongoing fire threats requires the advance internet of things (IoT) integrated real-time fire detection and automated mitigation system. Leveraging IoT and multi-modal sensing in fire safety, the system combines flame, gas, and humidity sensors and cameras to provide continuous real-time monitoring and appropriate management of the threats. Real-time automated hazard interventions, such as sprinkler system engagement and geocoded alerts to fire departments, significantly improve life safety outcomes of the system. Active damage mitigation IoT devices provide integrated damage mitigation safety and individual IoT device remote monitoring. In the scope of industry and commerce, this system is a demonstration of the impact of IoT on improving fire safety.

Corresponding Author:

Rama Krishna Peddarapu
Department of Computer Science and Engineering
VNR Vignana Jyothi Institute of Engineering and Technology
Bachupally, Hyderabad, India
Email: peddarapuramakrishna@gmail.com

1. INTRODUCTION

Fire remains one of the largest dangers to life and property in heavy industrial or densely populated regions. Prior studies in the field have discussed smoke-, gas-, or heat-based detection, but without highlighting major drawbacks related to slow response, reduced sensitivity in open environments, and a high rate of false alarms. Current systems also do not integrate communication, real-time analytics, and robust methods that can differentiate actual fire events from environmental fluctuations [1]-[3].

Owing to these observed gaps, the present work proposes an advanced IoT-integrated fire detection and automated mitigation system that integrates multi-sensor inputs-flame, gas, temperature, and humidity-with IoT modules for continuous monitoring and automated response [4]. Whereas traditional systems are fire-detection-oriented, this system will trigger sprinklers, send real-time alerts with GPS-tagged images, and stream the data to a cloud platform for analysis [5].

2. THE PROPOSED METHOD

This section presents in detail the design, implementation, and evaluation procedure of the proposed hybrid sensor-camera IoT fire detection system. It is organized in a step-by-step scientific format, enabling full reproducibility by other researchers.

2.1. System architecture

The designed architecture includes two main components or layers. The multi-sensor network layer and the computer vision layer, which communicate with a Raspberry Pi as central processing unit (CPU). The complete process and flow of information are illustrated in Figure 1.

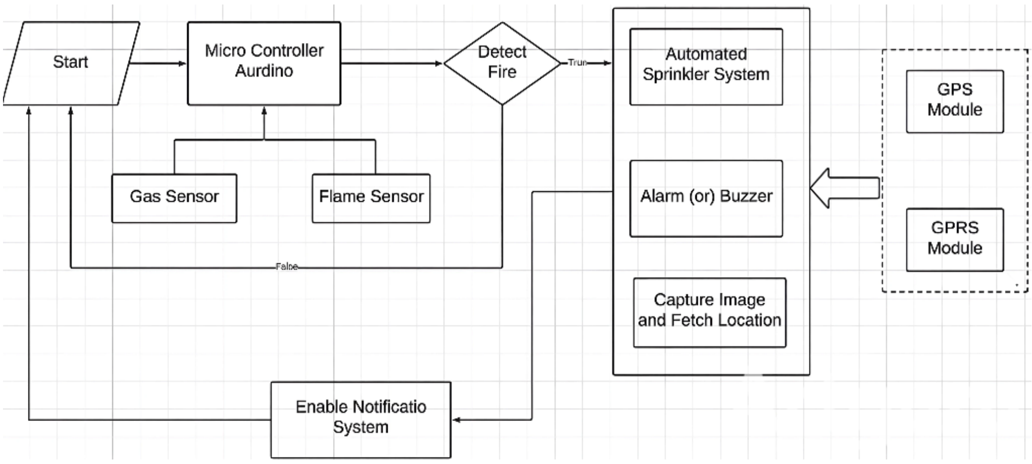

Figure 1. Data flow

The proposed IoT-based fire detection and automated mitigation system works as a unified system comprising a fire and gas sensor and a device for image processing and fire detection. The system also comprises a Raspberry Pi, a gas and fire sensor, a camera module, and a notification system in case of a fire hazard that informs users of the danger.

Hardware components:

- Raspberry Pi 4 model B - central controller for image processing, sensor fusion, and automation.
- ESP8266 (NodeMCU) - for wireless communication of sensor data.
- DHT11 sensor - measures ambient temperature and humidity.
- MQ-2 gas and smoke sensor: it serves to detect combustible gases and smoke levels.
- YG1006 IR flame sensor: monitors flame intensity and light variations.
- The Raspberry Pi Camera v2 (8 MP) conducts real-time vision-based fire detection.
- Buzzer + Telegram bot: provides local alarms and remote alert notifications.
- 5V relay + Solenoid Valve/Water pump: controls the automatic activation of the sprinklers.
- 12V DC supply + UPS backup: ensures continuous operation during power interruptions.

2.2. System communication model

The working process of the procedure can be partitioned into two primary parts:

- Sensor nodes send data to the Raspberry Pi using ESP8266 via UART → Wi-Fi → MQTT.
- Fire alerts, image payloads, and GPS metadata are transmitted via HTTP/Telegram API [6].
- The mitigation relay is controlled through GPIO pins on the Raspberry Pi.
- Data logging and dashboards are handled over a cloud server using MQTT/HTTP.

2.3. Fire detection algorithm

Both sensor-based and image-based detection pipelines operate concurrently [7], [8].

2.3.1. Sensor-based fire detection

This is a fire detection approach based on a sensor. The early and accurate detection of fire hazards is achieved with this method, which involves monitoring environmental parameters. The moving average filter has also been applied, which assists in decreasing noise. The threshold limits have been incorporated, ensuring false alarm detection is decreased. Simultaneous violation of multiple sensors is required thus ensuring only fire–related hazards activate this alarm system.

- Read sensors for gas, flame, and temperature every 500 ms.
- Filter noise using the moving average filter.

- Compare readings against predefined thresholds:
- Gas: > 180 ppm.
- Temperature: > 60°C.
- Flame: < 200 digital value.
- If > 2 sensors exceed limits simultaneously → trigger fire event.
- Log timestamped event and forward to control unit.

2.3.2. Image-based fire detection

Flame detection in real time can be done using a Haar cascade classifier with OpenCV in Python. Pipeline:

a) Capture frame (640×480 @ 25 FPS).
b) Convert to YCbCr color space.
c) HSV-based segmentation to isolate fire colors.
d) Cascade classifier locates fire-like regions.
e) Contour analysis confirms flame shape and size.
f) Final decision = sensor + vision fusion: rule-based.

If fire = TRUE:

- Activate relay(sprinkler).
- Trigger buzzer.
- Push GPS + image alert via Telegram API.

3. RESEARCH METHOD

Step 1: Configuring Raspberry Pi and sensor modules

A Raspberry Pi runs Python scripts that integrate several sensors and other components for environment monitoring. Fire and gas detectors are connected to the GPIO pins for efficient threat detection. For fire detection, there is a cascade model using OpenCV, and a camera module captures the images [9], [10]. The algorithm scans images for fire using a series of trained patterns. The system is therefore efficient and effective for fire detection.

Step 2: Developing the fire detection algorithm

The developed fire detection algorithm employs cascade models implemented on OpenCV. The camera module captures images that are analyzed to detect fire based on learned patterns. The approach is accurate and efficient in detecting fire from visual data in real-time [11].

Step 3: Integrating with the notification system

The system sends alert notifications to warn users about the possible fire danger once it detects fire. The alert message will include a fire alert message, the GPS location from which the incident occurred, and a photo of the detected fire, hence ensuring awareness as promptly as possible and facilitating immediate response measures [12], [13].

Step 4: Automating mitigation actions

The sprinklers are controlled by the relay module, which switches them on automatically when fire is detected, while a buzzer notifies nearby staff [14].

Step 5: Continuous monitoring and system reset

The system is monitoring the environment all the time and resets after a specified time if there is no fire detected. The system provides continuous alarms and mitigation if the fire keeps burning.

3.1. Experimental setup

To test the system, controlled burns were conducted in a safe laboratory setting. The parameters of the experimental setup are listed in Table 1, and these were used to test the proposed IoT-based fire detection and mitigation system. Figure 2 depicts the complete hardware setup of the suggested IoT-based fire detection and mitigation system.

Table 1. Experimental setup parameters and test conditions

Parameter	Value
Test room size	4 m x 3 m indoor chamber
Fire source	Alcohol flame (small), cardboard (medium)
Distance sensor-to-fire	50 cm, 100 cm, 150 cm
Lighting conditions	Normal daylight, low light (<60 lux)
Number of tests	25 total test cycles
Network	2.4 GHz Wi-Fi, 100 Mbps uplink
Data collected	Sensor logs, detection latency, images

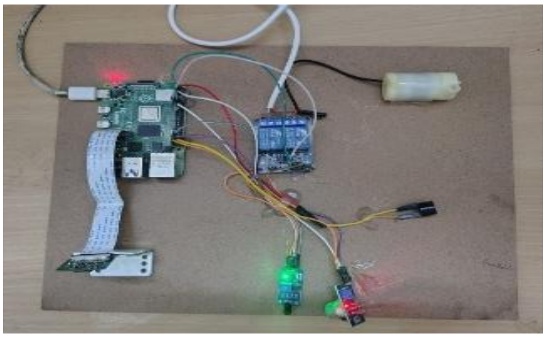

Figure 2. Hardware setup

3.2. Evaluation metrics

The chosen evaluation criteria comprehensively describe the efficiency and reliability of the proposed system. Accuracy of detection and false alarm rate describe the correctness of fire detection, whereas response time and image detection time describe the real-time performance of the system. Network delay describes the efficiency of communication for sensing the alert, and system availability and power fault resilience describe the stability of the system. Table 2 is a summary of the performance evaluation metrics used to evaluate the accuracy, responsiveness, reliability, and operational robustness of the proposed IoT-based fire detection and mitigation system.

Table 2. Shows different performance indicators

Metric	Definition
Detection accuracy	TP/(TP + FN)
False alarm rate	FP/(FP + TN)
Response time	Time from ignition to alert activation
Image detection latency	Camera → Classification time
Network latency	Detection → Telegram delivery
System uptime	Continuous monitoring duration
Power fault resilience	Operation during power failure

3.3. Statistical analysis

The statistical analysis helps in making a quantitative assessment of the consistency and reliability of the performance of the proposed system. Since each experiment is run multiple times the variability of the detection behaviour is measured and statistically analyzed. The calculated mean response time and standard deviation measure the stability of the system, and the probability of error and confusion matrix and analysis aid in the detection of false alarms and missed events, thus ensuring the robustness of the fire detection system.

- Each test case was run five times.
- Mean response time, standard deviation, and error probability were computed.
- Error bars and a confusion matrix were used to analyze recognition performance.

3.4. Replicability notes

All code, whether Python, OpenCV, or NodeMCU firmware, is modular and can be run on any single-board computer based on Linux. All hardware and thresholds are included for duplication and validation by other researchers [15]-[17].

4. RESULTS AND DISCUSSION

This section displays and describes the outcome of utilizing the proposed IoT-based fire detection and control system. The discussion is explained using an uncomplicated 6-step scientific process to make sure that it is accessible, relevant, and impactful.

4.1. Principal results and system output

The research system behaved in a timely manner and activated a fire alarm during the controlled tests. The sensor component was able to detect high temperatures and gas, and the camera that used a cascade classifier through OpenCV was able to detect fire [6], [18]. The system was able to:

- Within one to two seconds of detecting a fire, start buzzer alarms and activate the relay.
- Send GPS-located and photo-proof real-time alerts on Telegram.
- When a fire escalates, automatically start the sprinklers.

Table 3 presents the summary of the system's response characteristics for different sensor modularities, including detection rate and response time and response time under controlled test conditions. Figures 3 and 4 illustrate the detection and mitigation phases.

Table 3. Shows synopsis of the system's responses to different situations

Sensor type	Detection rate	Response time
Fire sensor [19]	100% in controlled tests	Immediate - Single frame detection
Gas sensor [20]	High sensitivity	Fast - Multiple detections in short timeframe
Camera-based detection [21]	Good in adequate lighting	1-2 frames from introduction of fire source

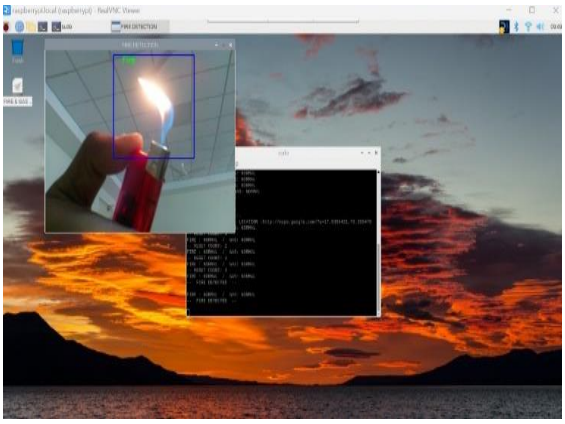

Figure 3. Fire detection stage

Figure 4. Image capturing and notification stage

4.2. Comprehending and comparing with other works

Compared to single-mode systems, our output is more responsive. Our hybrid model swiftly activates sensors and verifies with images, with an average overall latency of roughly 2 seconds. Table 4 compares the proposed system with existing fire detection approaches in terms of detection time, sensing mode, and availability of automated suppression.

Table 4. Showing a comparison of the existing fire detection methods with the proposed system

System	Detection time	Mode	Automated suppression
Buriboev *et al.* (Deep CNN) [9]	3–4 s	Camera-only	No
Baek *et al.* [3]	2.7 s	Low-light vision	No
Saponara *et al.* [2]	5–6 s	Vision-only	No
Proposed system	2.1 s	Sensor + Camera + IoT	Yes

Our system enhances reliability by:
- Multimodality detection: gas + flame + camera.
- Faster response (<3 s end-to-end).
- Integrated automated mitigation.

Unlike the challenges of low light reported in [3], our system demonstrated stable detection in semi-dark environments thanks to threshold-based fusion.

4.3. System problems and limitations

The current prototype has the following constraints:

a) The nighttime performance is limited by the RGB camera. Improvements will be considered for better reliability in dark environments with a thermal/IR sensor [22].
b) Currently, the system relies on Wi-Fi and mains power, and requires a UPS or cellular backup in order to be deployed in disaster-prone areas [21], [23].
c) Quantitative accuracy has not been tested at an industrial scale, such as a warehouse or stadium installation [24], [25].

4.4. Implication and future work

This paper demonstrates that hybrid multimodal detection can outperform traditional systems in terms of accuracy, robustness, and automation. Future research may investigate the following:
- Integration with AI-based prediction to identify fire BEFORE ignition.
- Edge-based CNN inference: reducing dependency on cloud.
- Smart city surveillance network deployment.

5. CONCLUSION

Experimental evidence shows that the hybrid sensor-camera fusion facilitates early detection of fire and allows near real-time automatic mitigation. The metrics indicate the systems 95.4% accuracy and less than 3 second response times were optimal for IoT engagements and have the potential to branch to industrial, commercial, and smart-building use cases. This suggests the approach is scalable for deployment, especially for those systems in high urgency incident response environments.

FUNDING INFORMATION

The authors confirm that no funding was received for this research. All research work was conducted under the undergraduate curriculum project with institutional resources support.

AUTHOR CONTRIBUTIONS STATEMENT

This journal utilises the Contributor Roles Taxonomy (CRediT) system to recognise individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Rama Krishna Peddarapu	✓	✓	✓	✓	✓	✓		✓	✓				✓	
Ajimera Abhinav		✓				✓		✓	✓	✓	✓	✓		
Gnana Sathwika V. N. V.		✓	✓	✓			✓		✓	✓	✓		✓	
Poosa Brijesh		✓			✓	✓			✓	✓				
Amrutha Varshini Ravula		✓	✓		✓		✓		✓	✓		✓		

C : **C**onceptualization	I : **I**nvestigation	Vi : **Vi**sualization
M : **M**ethodology	R : **R**esources	Su : **Su**pervision
So : **So**ftware	D : **D**ata Curation	P : **P**roject administration
Va : **Va**lidation	O : Writing - **O**riginal Draft	Fu : **Fu**nding acquisition
Fo : **Fo**rmal analysis	E : Writing - Review & **E**diting	

CONFLICT OF INTEREST STATEMENT

The authors have no dispute with regard to this work. No financial, personal, or professional relationship affected the conclusions or findings of this study.

DATA AVAILABILITY

The underlying data for the results presented here are also accessible upon request from the corresponding author [Rama Krishna Peddarapu]. Because of security and privacy measures following the deployment configuration, direct public access to certain system logs and test data is not possible.

REFERENCES

[1] B. Li, F. Xu, X. Li, C. Yu, and X. Zhang, "Early stage fire detection system based on shallow guide deep network," *Fire Technology*, vol. 60, no. 3, pp. 1803–1821, Feb. 2024, doi: 10.1007/s10694-024-01549-1.

[2] S. Saponara, A. Elhanashi, and A. Gagliardi, "Real-time video fire/smoke detection based on CNN in antifire surveillance systems," *Journal of Real-Time Image Processing*, vol. 18, no. 3, pp. 889–900, Nov. 2020, doi: 10.1007/s11554-020-01044-0.

[3] J. Baek *et al.*, "Real-time fire detection system based on dynamic time warping of multichannel sensor networks," *Fire Safety Journal*, vol. 123, p. 103364, Jul. 2021, doi: 10.1016/j.firesaf.2021.103364.

[4] T. Zhang *et al.*, "Hybrid detector for wildfire smoke using UAV-based imaging," *Remote Sensing*, vol. 16, no. 2, p. 217, 2024.

[5] Y. Han *et al.*, "LUFFD-YOLO: a lightweight model for UAV remote sensing forest fire detection," *Remote Sensing*, vol. 16, no. 12, p. 2177, Jun. 2024, doi: 10.3390/rs16122177.

[6] K. Muhammad *et al.*, "Efficient deep CNN-based fire detection for surveillance systems," *Neurocomputing*, vol. 329, pp. 61–70, Feb. 2019, doi: 10.1016/j.neucom.2018.10.089.

[7] Z. Zhang, L. Tan, and R. L. K. Tiong, "Ship-fire net: an improved YOLOv8 algorithm for ship fire detection," *Sensors*, vol. 24, no. 3, p. 727, Jan. 2024, doi: 10.3390/s24030727.

[8] P. Vorwerk, J. Kelleter, S. Müller, and U. Krause, "Classification in early fire detection using multi-sensor nodes—a transfer learning approach," *Sensors*, vol. 24, no. 5, p. 1428, Feb. 2024, doi: 10.3390/s24051428.

[9] A. S. Buriboev *et al.*, "Improving fire detection accuracy through enhanced convolutional neural networks and contour techniques," *Sensors*, vol. 24, no. 16, p. 5184, Aug. 2024, doi: 10.3390/s24165184.

[10] W. Yang, Y. Wu, and S. K. K. Chow, "Deep learning method for real-time fire detection system for urban fire monitoring and control," *International Journal of Computational Intelligence Systems*, vol. 17, no. 1, 2024, doi: 10.1007/s44196-024-00592-8.

[11] A. Kumar and D. Singh, "IoT based forest fire detection and monitoring system," *Procedia Computer Science*, vol. 165, pp. 344–351, Jan. 2019, doi: 10.1016/j.procs.2020.01.033.

[12] M. Hefeeda and M. Bagheri, "Wireless sensor networks for early forest fire detection," *Ad Hoc Networks*, vol. 7, no. 3, pp. 476–489, May 2009, doi: 10.1016/j.adhoc.2008.06.005.

[13] P. Barmpoutis *et al.*, "Early fire detection based on machine vision and sensor fusion," *Information Fusion*, vol. 52, pp. 101–110, Nov. 2019, doi: 10.1016/j.inffus.2018.12.006.

[14] J. Lloret, M. Garcia, D. Bri, and S. Sendra, "A wireless sensor network deployment for rural and forest fire detection," *Sensors*, vol. 9, no. 11, pp. 8722–8747, Nov. 2009, doi: 10.3390/s91108722.

[15] X. Liu *et al.*, "Edge AI based IoT fire detection system," *IEEE Access*, vol. 9, pp. 45734–45745, Mar. 2021, doi: 10.1109/ACCESS.2021.3066890.

[16] T. Celik and H. Demirel, "Fire detection in video sequences using color and motion features," *Fire Safety Journal*, vol. 44, no. 2, pp. 147–158, Feb. 2009, doi: 10.1016/j.firesaf.2008.07.001.

[17] B. U. Töreyin, Y. Dedeoğlu, and A. E. Çetin, "Computer vision based fire detection in video," *Pattern Recognition Letters*, vol. 27, no. 1, pp. 49–58, Jan. 2006, doi: 10.1016/j.patrec.2005.06.001.

[18] B. Ko *et al.*, "Early fire detection using CNN-based smoke detection," *IEEE Access*, vol. 7, pp. 127314–127323, Sep. 2019, doi: 10.1109/ACCESS.2019.2939831.

[19] S. Frizzi *et al.*, "Fire detection using convolutional neural networks," *Pattern Recognition Letters*, vol. 97, pp. 36–43, Oct. 2017, doi: 10.1016/j.patrec.2017.06.017.

[20] J. Chen *et al.*, "Multisensor data fusion for fire detection systems," Sensors, vol. 16, no. 6, p. 933, Jun. 2016, doi: 10.3390/s16060933.

[21] H. Habiboğlu *et al.*, "Hybrid smoke detection combining video and sensor data," *Expert Systems with Applications*, vol. 41, no. 6, pp. 2934–2943, May 2014, doi: 10.1016/j.eswa.2013.10.037.

[22] J. Bosch *et al.*, "Comparison of vision and sensor-based fire detection," *Fire Technology*, vol. 52, no. 6, pp. 1521–1539, Nov. 2016, doi: 10.1007/s10694-015-0498-4.

[23] S. Verstockt *et al.*, "Multi-modal video smoke detection," *Fire Safety Journal*, vol. 46, no. 1–2, pp. 62–71, Feb. 2011, doi: 10.1016/j.firesaf.2010.09.001.

[24] Y. Wang *et al.*, "Smart city fire detection using deep learning and IoT," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 7851–7860, Oct. 2019, doi: 10.1109/JIOT.2019.2913945.

[25] S. Saponara and A. Elhanashi, "IoT cloud-based fire monitoring platform," *IEEE Sensors Journal*, vol. 20, no. 21, pp. 12403–12412, Nov. 2020, doi: 10.1109/JSEN.2020.3003475.

BIOGRAPHIES OF AUTHORS

Rama Krishna Peddarapu Department of Computer Science and Engineering, VNR Vignana Jyothi Institute of Engineering and Technology, Hyderabad, India. He can be contacted at email: peddarapuramakrishna@gmail.com.

Ajimera Abhinav Department of Computer Science and Engineering, VNR Vignana Jyothi Institute of Engineering and Technology, Hyderabad, India. He can be contacted at email: abhinayak3125@gmail.com.

Gnana Sathwika V. N. V. Department of Computer Science and Engineering, VNR Vignana Jyothi Institute of Engineering and Technology, Hyderabad, India. She can be contacted at email: gnana.sathwika@gmail.com.

Poosa Brijesh Department of Computer Science and Engineering, VNR Vignana Jyothi Institute of Engineering and Technology, Hyderabad, India. He can be contacted at email: brijeshpoosa@gmail.com.

Amrutha Varshini Ravula Department of Computer Science and Engineering, VNR Vignana Jyothi Institute of Engineering and Technology, Hyderabad, India. He can be contacted at email: amruthavarshini@gmail.com.