International Journal of Safety and Security Engineering
Vol. 15, No. 10, October, 2025, pp. 2081-2091
Journal homepage: http://iieta.org/journals/ijsse

Design and Implementation of a Smart Dual-Stage Fire Crisis Management System Using Raspberry Pi for Safety and Security Applications

Irianto[1], Jamil Abedalrahim Jamil Alsayaydeh[2*], Adam Wong Yoon Khang[2], Mazen Farid[3], Safarudin Gazali Herawan[4]

[1] Department of General Education, Faculty of Resilience, Rabdan Academy, Abu Dhabi 22401, United Arab Emirates
[2] Department of Engineering Technology, Fakulti Teknologi Dan Kejuruteraan Elektronik Dan Komputer (FTKEK), Universiti Teknikal Malaysia Melaka (UTeM), Melaka 76100, Malaysia
[3] Faculty of Information Science and Technology, Multimedia University, Melaka 75450, Malaysia
[4] Department of Industrial Engineering, Faculty of Engineering, Bina Nusantara University, Jakarta 11480, Indonesia

Corresponding Author Email: jamil@utem.edu.my

https://doi.org/10.18280/ijsse.151011

Received: 5 September 2025
Revised: 6 October 2025
Accepted: 17 October 2025
Available online: 31 October 2025

Keywords:
fire detection and prevention, safety and security engineering, crisis management system, Raspberry Pi, IoT-based alerting, false alarm reduction

ABSTRACT

Advances in Internet of Things (IoT) and embedded computing have made it possible to build smarter fire alarms that reduce false triggering, not just detect heat or smoke. This study presents a Raspberry Pi–based fire crisis controller that uses two-stage verification: an infrared flame sensor triggers first, then a Pi Camera runs OpenCV-based image checks to confirm fire before an alert is escalated. Requiring agreement between hardware sensing and vision helps suppress nuisance activations. The prototype integrates the flame sensor, camera, and a piezo buzzer with software for image filtering, database logging, and web-based IoT alerts. In 30 controlled indoor trials, it achieved 98% average detection accuracy and reduced false alarms by 92% compared with a baseline single-sensor flame detector. End-to-end response from ignition to alert activation averaged 9.4 s and stayed under 10 s in all scenarios. After confirmation, the controller sounds the buzzer and posts an alert through the web interface, enabling faster response. Overall, the results show early detection with strong false-alarm suppression using low-cost hardware suitable for residential and small industrial settings. Future work will add smoke and temperature sensing, support offline operation during network outages, and explore RFID tracking of safety equipment to improve on-site coordination.

1. INTRODUCTION

Fire accidents remain a serious risk to life and property, so detecting an incipient event in real time is critical for early evacuation and faster firefighting response [1, 2]. Yet many deployed alarms still rely on a single trigger, usually smoke or heat. In normal indoor settings, dust, steam, and cooking aerosols can disturb these sensors, raising the likelihood of false alarms [3, 4]. False triggers carry real consequences. They disrupt activities, prompt unnecessary evacuations, and add avoidable costs. Repeated nuisance alarms also cause alarm fatigue, where people react more slowly or stop taking alerts seriously, even during genuine emergencies [5]. The takeaway is clear: fire detection needs smarter decision logic that filters non-fire conditions without delaying a real warning.

Fire-safety surveys report that single-sensor alarms can produce frequent nuisance triggering in real deployments, especially under steam, dust, and reflective lighting. Reported ranges include about 25-35% in residential settings and up to roughly 40% in some industrial conditions (see recent fire-safety survey reports for these figures). These rates waste time, burden responders, and often lead users to silence or disable the alarm. A similar limitation appears in many Raspberry Pi prototypes that rely on one modality only, either a flame sensor or vision. Under fire-like disturbances such as sunlight flicker or hot reflections, reported false positive rates can exceed 20% in stressed tests. Many of these systems also stop at local logging or a buzzer, which limits real-time coordination.

This motivates a dual-stage sensor plus vision verification design tied to web-based alerting, aiming to suppress false alarms without delaying notification.

Recent progress in Internet of Things (IoT) and embedded computing has made fire monitoring more capable than what single-sensor alarms can offer, and the Raspberry Pi is often used as the hub for these low-cost systems. It is compact, supports Wi-Fi and cameras, and exposes simple GPIO interfaces, so it can read sensors and run lightweight analytics on the same device. When the Raspberry Pi is paired with multiple sensors and basic image processing, the design can become both faster and more selective. Vision algorithms can examine flame cues, such as motion and color patterns, and reject common “fire-like” disturbances that would otherwise trigger false alarms [6]. In the same direction, industry evaluations of multi-sensor detectors show that combining smoke, heat, and flame signals reduces nuisance alarms while still supporting early detection, because a single noisy channel

is less likely to dominate the decision.

This project addresses a practical gap in low-cost fire alarms: many systems trigger on a single unverified signal, which increases false alarms, while others lack real-time IoT notification that supports coordinated response. The result is all too often either annoying false alarms popping up without ceasing or a delayed response when time is of the essence. To bridge that gaping hole between detection and reaction, we've developed a compact, cost-effective controller centered around two-stage verification. A hardware flame sensor zaps out an initial trigger fast, then a camera on a Raspberry Pi kicks in to run some image analysis checks to verify whether what it's seeing actually looks like a genuine fire scenario. Only once the fire's confirmed does the controller send out a web-based alert to all the relevant people, like building managers, response teams, and anyone else who needs the heads up, so it's not just a local buzzer that goes off.

This study has three objectives. First, we implement a two-tier fire detection workflow that uses a flame sensor for rapid triggering and computer vision for confirmation, so the alarm is based on validated evidence rather than a single noisy signal. Second, we develop a connected management layer that records the locations of fire-safety equipment and automates emergency notifications to the right people when an event is confirmed. Third, we evaluate the complete system under realistic conditions and report its performance, with emphasis on false-alarm reduction and response time. By combining IoT alerting with sensor plus vision validation, the system is designed to improve situational awareness and support faster, better coordinated action during an incident. The rest of the paper describes the architecture and implementation in the Methodology section, presents the evaluation in Results and Discussion, and closes with key findings and future improvements.

Most related work focuses on one path at a time. Some studies rely on a single sensor, while others use vision-based detection, but they are often implemented in isolation and do not cross-check decisions across modalities. The same pattern appears in many Raspberry Pi prototypes: They detect locally and stop there, with limited support for IoT dashboards, event logging, or automated notification workflows. This study responds to those gaps with a dual-stage verification design. A flame sensor provides a fast first trigger, then image analysis confirms whether the trigger matches true fire behavior, which helps reduce false alarms without slowing the response. Beyond detection, the system includes a web-based monitoring and alerting layer. It logs events in real time and supports coordination, including guiding responders to the recorded locations of safety equipment. This work contributes three elements. First, a low-cost Raspberry Pi controller that pairs a flame sensor with vision verification, so triggers are confirmed before escalation. Second, an alert and management module that pushes notifications to the right stakeholders through a web interface and keeps a complete event log. Third, an experimental validation under varied conditions showed higher detection accuracy, fewer false alarms, and response times measured in seconds, as reported in this study.

2. LITERATURE REVIEW

Fire alarm systems are built for early detection and coordinated response. A standard setup includes a control panel, initiating devices such as manual call points or automatic detectors, notification devices like sirens and strobes, and interfaces that support evacuation and suppression. In practice, many installations still rely on a single cue, usually smoke or heat, and that creates a known weakness. Dust, steam, and cooking aerosols can mimic fire signatures indoors, increasing false alarms [7, 8]. These nuisance events disrupt operations and reduce trust, and repeated triggers can cause alarm fatigue and slower reactions during real emergencies [9]. To improve early-stage detection, researchers have expanded sensing and communication. Early electronic nose systems used gas-sensor arrays to detect combustion products, and Charumporn et al. [10] showed they can detect smoke-related gases early, although selectivity limits still produced nuisance triggers. Later work [11, 12] adopted wireless architectures for easier deployment and remote monitoring, with Dong et al. [12] reporting lower power use and improved flexibility in low-cost wireless designs. Wireless sensor networks, including ZigBee-based topologies, then enabled multi-sensor fusion using inputs such as temperature, humidity, and dust, improving coverage and reducing false alarms relative to single sensor triggering [13].

Because each sensor behaves differently, Table 1 compares flame, smoke, and temperature sensing. This study emphasizes flame sensing due to its fast response and low cost, while smoke and temperature sensors, although useful in specific cases, can be more expensive and more sensitive to benign indoor conditions that cause false alarms [14]. Takeaway: dependable detection comes from designs that match real indoor noise sources, not ideal lab conditions.

Computer vision can strengthen fire alarms that rely only on sensors because it adds visible evidence. Instead of triggering on a single threshold, image processing looks for flame cues such as color patterns, irregular contours, and flicker dynamics over time. Thengade et al. [15] showed a Raspberry Pi pipeline that improves frame quality using kernel filtering and morphological operations before running flame detection, which increased reliability in small and open areas. The takeaway is that vision works best as a confirmation step; it helps cut false triggers when it verifies a sensor trigger rather than replacing it.

Table 1. Comparison of fire detection sensors

Sensor Type	Detection Principle	Advantages	Limitations
Flame sensor	Detects infrared radiation (760–1100 nm) emitted by open flames	Fast response; cost-effective; accurate for visible fire	Cannot detect smouldering fires; sensitive to IR light sources
Smoke sensor	Detects smoke particles (ionization, photoelectric, aspirating, or laser-based)	Detects smouldering and concealed fires; widely used in commercial systems	Higher cost; prone to false alarms from dust, steam, and aerosols
Temperature sensor	Monitors a sudden rise in ambient temperature	Effective for monitoring heat-sensitive environments; reliable for high-temperature thresholds	Slower response to incipient fires; less suitable for rapid evacuation needs

Raspberry Pi–based vision systems can detect visible flames with high accuracy by using cues such as color, contour shape, and flicker dynamics. For example, Thengade et al. [15] and Khan et al. [16] reported that contour-based detection and color segmentation are effective in small and open environments where the flame is clearly visible. The limitation is also clear: image-only decisions can be misled by reflections, bright lighting, or objects that resemble fire, which increases the risk of false positives in real rooms and corridors.

Several studies have tried to make these systems more operational. Dhanujalakshmi et al. [17] coupled camera-based detection with Wi-Fi or GSM messaging to reduce notification delay, but the design still lacks multimodal verification and can increase bandwidth and processing load. For larger spaces, Wong and Fong [18] used richer spatial, spectral, and temporal video indicators to improve robustness and track flame spread, yet continuous video analysis raises computational cost, which is difficult to justify for low-cost residential deployment. The takeaway is that vision improves recognition, but vision alone remains fragile. A hybrid workflow, where a fast sensor trigger is confirmed by image analysis, better matches the constraints and noise sources of practical deployments. Overall, vision-based approaches show high recognition accuracy, with modern models achieving over 95–98% detection accuracy while reducing false positives from "fire-like" phenomena such as reflections or bright lights [19]. The reliance on vision alone is also not sufficient, highlighting the need for a hybrid design. Comparative work has shown that vision systems can complement sensors by providing verification, thus minimizing nuisance triggers. This motivates the proposed two-tier design.

Wu et al. [20] explains about a smart fire alarm system that is integrated with a wireless sensor network using ZigBee. This project is proposed for a fire alarm system that uses the ZigBee mesh topology. There are many sensors used in this project, which are temperature, humidity, pressure, and dust sensors. The proposed system uses an intelligent way of detecting smoke that can distinguish the air more accurately and thus reduce false fire alarms.

Despite the strengths of vision-based methods, reliance on a single modality remains a limitation. Industrial studies confirm that multi-sensor detectors (combining smoke, heat, flame, and vision) achieve far fewer false alarms than single-sensor systems [21]. This principle underpins recent innovations where IoT-enabled devices integrate sensing, data logging, and real-time communication. Modern IoT fire monitoring systems combine edge computing (for rapid local detection) with cloud platforms that log data, visualize status on dashboards, and send automated SMS/email alerts [22]. Keano and Jose [23] reported that adding IoT connectivity to fire-safety systems can improve evacuation management and reduce delays in mobilizing first responders, because alerts are delivered quickly to the right stakeholders rather than staying local to the device [23, 24]. This is reflective of a wider trend in the field: moving away from just detecting fires to implementing systems that can actually support decision making, coordination, and folks being able to bounce back after an incident. In that regard, connecting up lots of different devices via the internet is definitely a step forward, but on its own, it's just not enough. Many earlier prototypes still rely on a single detection modality, either sensors alone or vision alone, which leaves them exposed to nuisance triggers or visual confusion. Table 2 summarizes these related systems and contrasts their limitations with the dual-tier, IoT-enabled design proposed in this study. The takeaway is that the strongest designs couple actionable notification with verified detection, not one without the other.

Table 2. Comparative summary of related fire detection systems

Study / Approach	Hardware Platform	Detection Method	Additional Features	Limitations
Dong et al. [12]	Wireless sensor nodes	Temperature, humidity, dust sensors (ZigBee WSN)	Distributed monitoring, low power	Lacks visual confirmation; prone to false triggers from environmental noise
Thengade et al. [15]	Raspberry Pi + Camera	Image processing (color segmentation, contour detection)	Buzzer and LED indicators	Reliable in small areas, but prone to false alarms from fire-like phenomena; no IoT integration
Dhanujalakshmi et al. [17]	Raspberry Pi + Camera	Image-based detection with Wi-Fi/GSM alerts	Real-time image transmission to users	Higher cost and bandwidth demand; no multi-sensor verification
Wong and Fong [18]	Video cameras (PC-based)	Video fire detection with spatial, spectral, and temporal indicators	Flame spread tracking and prediction	Complex setup, not cost-effective for residential/SME use
BRE Group [25]	Industrial multi-sensor detectors	Smoke + heat + flame fusion	Commercial-grade accuracy; reduced false alarms	High cost, limited scalability for small installations
Proposed study	Raspberry Pi + Flame Sensor + Camera	Two-stage verification (sensor + vision)	IoT-enabled dashboard, automated emergency notifications, event logging, and equipment guidance	Achieves > 95% detection accuracy, > 90% reduction in false alarms, response time < 10 s; scalable and economical

Selecting the appropriate hardware platform is critical. Arduino and 8051 microcontrollers are often used for simple, repetitive tasks such as temperature logging or buzzer control, but they lack built-in support for networking and multimedia processing. In contrast, the Raspberry Pi is a single-board computer that offers higher computational power, USB and HDMI ports, Wi-Fi and Bluetooth, and camera integration. These features make the Raspberry Pi especially suitable for edge AI and IoT-based fire detection, where local processing and real-time image analysis are required [20, 26]. Comparative studies consistently show that Raspberry Pi outperforms Arduino and AVR boards in complex monitoring tasks where connectivity and multimedia support are essential.

The literature points to three clear gaps. Many Raspberry Pi–based systems use either sensors or vision alone, and few apply dual-stage verification where one trigger is confirmed by the other before raising an alarm. IoT support is also limited, with many prototypes stopping at local alerts instead

of dashboards and automated notifications. A third gap is cost-effective fusion. Industrial systems show the value of combining modalities, yet academic prototypes often overlook simple, low-cost pairings like a flame sensor plus a camera that could improve reliability. Recent studies published in the International Journal of Safety and Security Engineering have highlighted complementary aspects of fire safety research, including the influence of building geometry on fire spread dynamics and the role of user awareness in effective fire emergency response [27, 28]. While these works focus on structural fire behavior and human factors, respectively, they do not address real-time fire detection or automated verification mechanisms, reinforcing the need for intelligent, sensor- and vision-based fire detection systems such as the one proposed in this study.

This study closes the identified gaps with a dual-tier detection workflow. A flame sensor provides rapid preliminary triggering, then a Raspberry Pi camera performs vision-based confirmation before the alarm is escalated. This cross-checking reduces the false alarms that are common when a system depends on a single trigger. Beyond detection, the design includes a web-based crisis management layer that logs events, shows live status, and sends real-time notifications to emergency contacts, so coordination begins immediately rather than relying on someone noticing a local buzzer. The literature review informs these choices by clarifying what past systems do well and where they fail, especially around false triggers, notification latency, and edge-device constraints. It also guides practical component selection by matching hardware and software capabilities to real indoor conditions. In this work, components were selected based on evidence from prior studies and targeted testing, with the aim of improving reliability without increasing cost or complexity.

3. METHOD

3.1 System overview

The Smart Fire Crisis Controller is a two-stage detection system built on a Raspberry Pi that combines IoT reporting with computer vision verification. It first detects early fire cues using an infrared flame sensor, then confirms the event using camera-based image analysis before activating an alarm or issuing notifications. This hybrid workflow preserves fast sensor response while reducing false alarms through visual confirmation. The architecture is modular and organized into three layers. The sensing layer contains the flame sensor and camera for data capture. The processing and control layer runs on a Raspberry Pi 3 Model B+, which executes the decision logic and image checks. The notification and response layer includes a piezoelectric buzzer and a web dashboard for alerts, event logging, and remote monitoring. Figure 1 summarizes the full flow, from sensor input to verified event reporting on the IoT dashboard.

3.2 Design methodology

The system was developed using a modified Waterfall approach with defined stages and small refinements within each stage. Work began with requirements analysis to capture user needs and practical constraints on hardware, power, and software. The design stage then fixed the overall architecture, selected components, prepared the circuit layout, and defined the detection logic. Implementation integrated the hardware with Python modules on the Raspberry Pi, followed by testing to confirm functionality, detection accuracy, and response time under both controlled and variable conditions. Deployment finalized the web server and database configuration, and maintenance focused on tuning based on test outcomes. Figure 2 summarizes this sequence. The takeaway is that a staged workflow reduced integration mistakes by keeping each step testable before moving on.

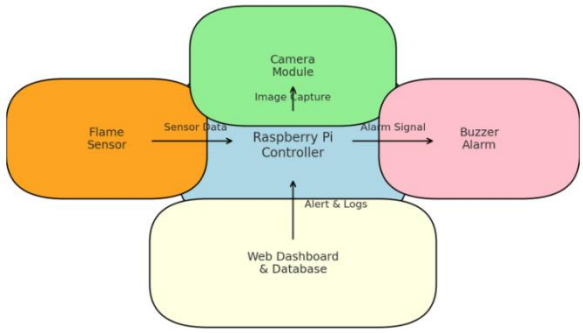

Figure 1. System architecture and communication layers

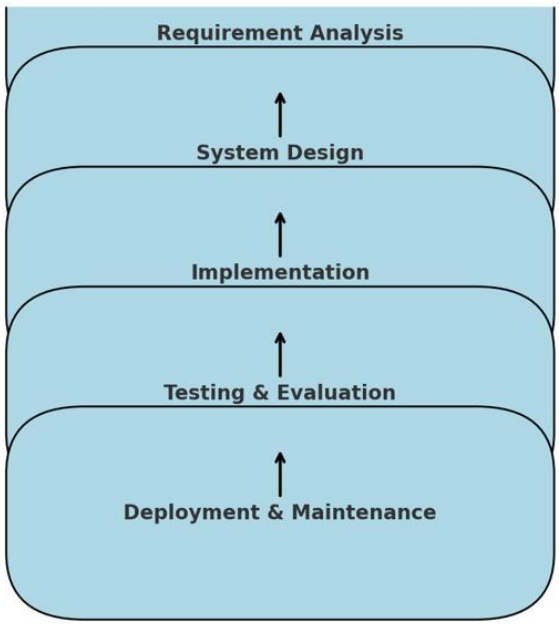

Figure 2. Modified waterfall design process

3.3 Hardware design

3.3.1 Central controller

The Raspberry Pi 3 Model B+ serves as the central controller because it balances capability and cost. Its 1.4 GHz quad-core 64-bit processor and 1 GB RAM are sufficient for on-device OpenCV processing, while built-in Wi-Fi and Bluetooth support networked alerts without extra modules. GPIO pins allow direct sensor interfacing, and the CSI port enables a simple, low-latency connection to the Pi Camera.

3.3.2 Sensing components

The sensing layer uses three components with clear roles. An infrared flame sensor, sensitive in the 760–1100 nm band, serves as the first trigger and sends a digital HIGH signal to the Raspberry Pi when flame radiation is detected. Once triggered, the camera module captures frames for verification

using a 5-megapixel sensor at 2592 × 1944 resolution, with capture delay kept below 0.5 s per frame. A piezoelectric buzzer provides local warning, producing a long beep at the initial trigger and switching to a continuous alarm after fire is confirmed.

3.3.3 Connectivity and power

The controller connects to the web server over IEEE 802.11 Wi-Fi to support real-time notifications and event logging. Power is provided by a regulated 5 V, 2.5 A supply for the Raspberry Pi and connected peripherals. To reduce energy use, the camera remains off during standby and is activated only after the flame sensor triggers. Figure 3 presents the wiring and GPIO connections between the Raspberry Pi, sensor, camera, and buzzer.

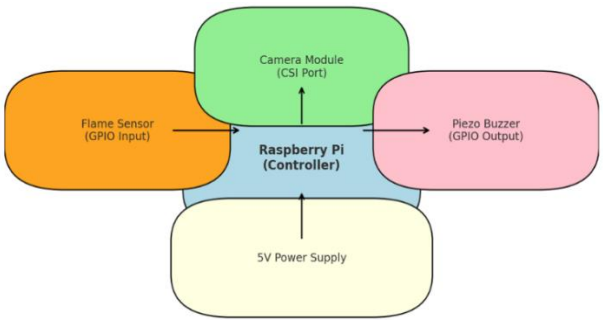

Figure 3. Hardware integration and wiring diagram

3.3.4 Hardware selection rationale

Component selection focused on accuracy, cost, and ease of deployment. The Raspberry Pi offers built-in Wi-Fi and direct camera support, avoiding extra communication or imaging modules that are often needed with Arduino-based designs. The infrared flame sensor was chosen for fast response and low cost while still supporting early flame detection. The piezo buzzer provides a strong audible alert with low current draw, under 20 mA, which suits continuous standby operation.

3.4 Software design

The software was developed primarily in Python 3 using several libraries:

- OpenCV: Performs image filtering, color segmentation, and contour analysis for flame recognition.
- RPi.GPIO: Handles digital input/output between the Raspberry Pi and connected sensors.
- smtplib and Requests: Manage automated email or SMS notifications to remote users.
- Flask / PHP + MySQL: Implement the lightweight web server and database backend for event logging.

The software runs as a continuous loop that reads the flame sensor in real time. When the sensor indicates a possible fire, the Raspberry Pi turns on the camera and performs a short verification step before escalating the alarm. Each frame is converted from RGB to HSV to make color filtering more stable under lighting changes. The algorithm then isolates flame-like pixels using hue 0–50 with saturation above 150 and value above 200, applies erosion and dilation to remove noise, and checks contour properties such as area and boundary shape. If the frame passes verification, the alarm and notification routines are triggered. If not, the event is logged as a false trigger, and the system returns to standby.

3.4.1 Flame-interferent discrimination and parameter selection

To reduce false positives from incandescent bulbs, sunlight glare, reflections, and hot surfaces, the vision stage combines three constraints: color, intensity, and geometry. HSV segmentation is used because it is less sensitive to illumination shifts than RGB. In our indoor trials using candles and lighters, true flames consistently fell within hue 0–50° with high saturation and value, while many artificial lights showed lower saturation and more uniform brightness. After segmentation, erosion and dilation with a 5 × 5 kernel suppresses isolated pixels while preserving connected regions, and the kernel size was chosen empirically to balance noise removal against shape distortion. Final verification uses contour filtering. Regions smaller than 500 pixels are rejected because reflection artifacts and light flicker tend to produce small, unstable blobs, and accepted contours must show irregular boundaries that better match real flame edges. These thresholds and limits were tuned iteratively across varied indoor conditions, including reflective surfaces and ambient lighting. The takeaway is that combining simple constraints makes the verification step more selective, reducing false confirmations without adding heavy computation.

Figure 4 summarizes the two-tier verification logic. The flame sensor provides the first trigger, then the system runs image-based confirmation before escalating the alarm.

The pipeline runs in real time on the Raspberry Pi's multicore CPU. Frame processing latency remains below 1 s on average, and when combined with the sensor's near-instant trigger, the end-to-end detection to alert cycle stays under 10 s in the reported tests.

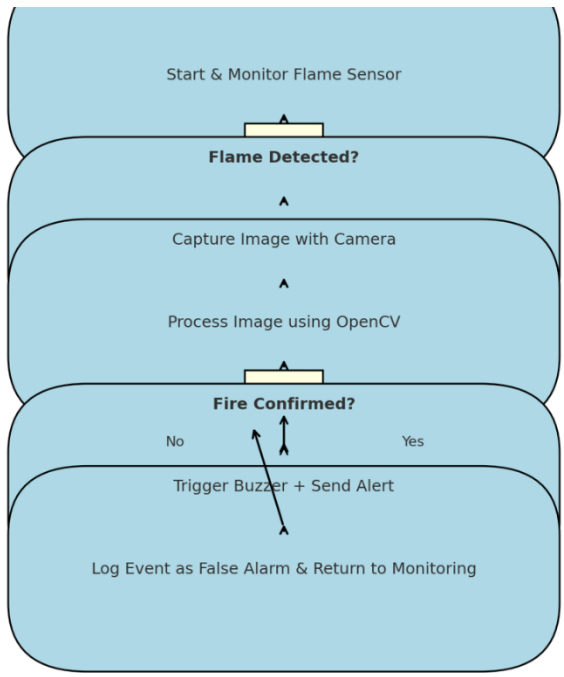

Figure 4. Pseudocode of the two-stage fire detection algorithm incorporating HSV-based flame filtering, morphological noise suppression, and contour-area validation

3.5 System workflow

Figure 5 presents the end-to-end workflow. The flame sensor runs continuously in standby, watching its field of view. When it detects a possible heat or flame source, it triggers the Raspberry Pi to start the verification stage. The Raspberry Pi activates the camera, captures an image, and runs the analysis

pipeline. If the image does not confirm fire, the event is logged as a false trigger, and the system returns to monitoring. If fire is confirmed, the buzzer sounds immediately and the controller uploads the event record, including timestamp, location, and image evidence, to the database and dashboard. The web interface will update in real-time and automatically send out alerts to the emergency team as soon as something happens.

The way the system works, from the moment it's turned on right through to sending out those IoT alerts, is laid out in detail in Figure 5, that's where you'll see the if-then logic at play for distinguishing between actual fires and false alarms.

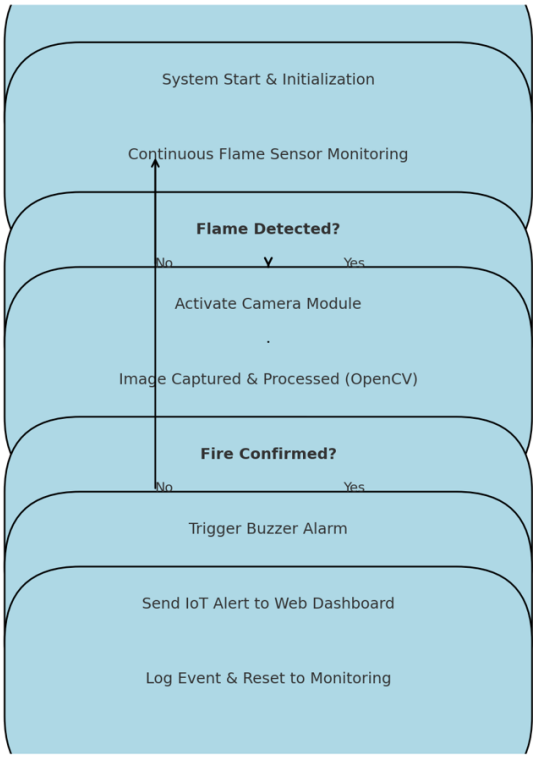

Figure 5. System workflow and decision process flowchart

3.6 Testing and evaluation

3.6.1 Experimental setup

Testing was done in a controlled indoor environment, using a very particular setup: a candle and a lighter were placed at distances ranging from 10 cm to 60 cm to create a variety of flame intensities and angles of view. To make sure our system could handle a false alarm, we also ran some tests under artificial lighting conditions and near shiny surfaces. Figure 6 shows the setup in all its glory, the flame source, the Raspberry Pi unit, and the camera view all in one place.

3.6.2 Performance metrics

System performance was put through its paces using three key metrics: detection accuracy, false alarm reduction, and how quickly it responded to situations. And the results are laid out in Table 3, which gives us the nitty-gritty and tells us how to interpret what we're looking at.

Table 3 shows that the system's nail detection accuracy at 95% plus and actually does a much better job than usual of avoiding false alarms, more than 90% better, to be precise. And the average time it took for the system to respond was a snappy 10 seconds or less, which is no bad thing. Plus, the system was able to keep chugging along at a 98% reliability rate, a pretty impressive feat of continuous operation. The same trends are mirrored in Figure 7, which gives us a snapshot of the system's performance and shows 95% accuracy and less than 10% false alarms, a clear step up from the usual single-sensor alarms.

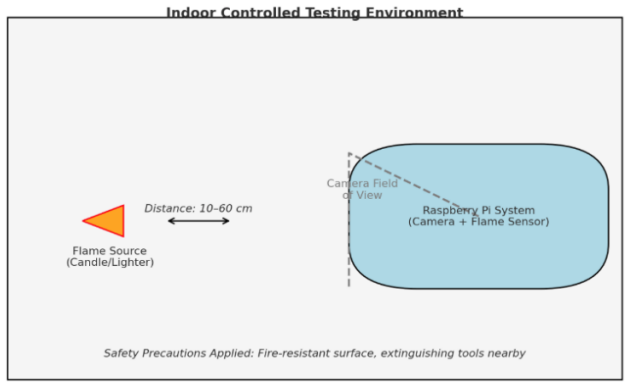

Figure 6. Experimental setup and testing environment

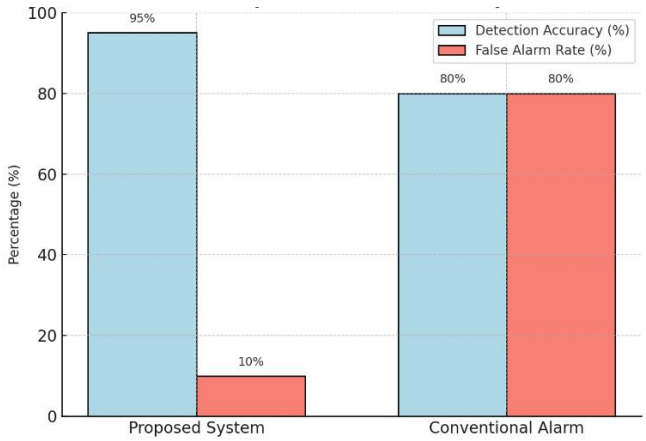

Figure 7. Performance graph – Detection accuracy vs. false alarm rate

Table 3. Performance metrics and evaluation criteria of the proposed fire detection system

Parameter	Observed Value / Result	Evaluation Criteria
Detection accuracy	> 95%	Percentage of correctly detected real fires vs. false negatives
False-alarm reduction	> 90%	Reduction in false triggers compared with single-sensor alarms
Response time	< 10 s	Time from flame ignition to alarm activation
System uptime / Reliability	98%	Stability of operation over a continuous 12-hour runtime

3.6.3 Web application testing

We also tested how the web application performed in real-time - not just when things were going smoothly, but when something actually needed to be reported. And what we found was that once an event had been confirmed, the dashboard would kick in straight away with the sensor ID, the exact time it happened, and a big, clear status indicator - green for all good, red for a problem. And for good measure, all the events were stored away in a MySQL database, just in case we needed to go back and review them later. Figure 8 gives us a peek at

how the web app and database work together - and how seamlessly the alerts and logs keep getting updated.

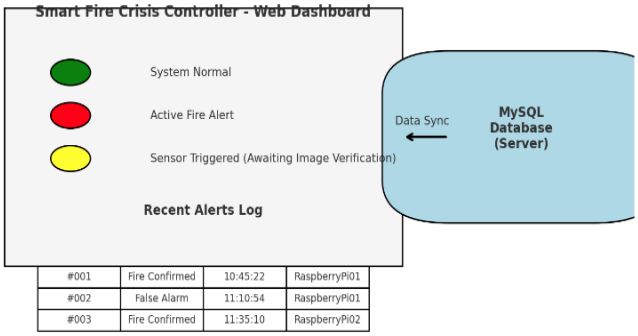

Figure 8. Web application dashboard and database interface

3.6.4 Discussion of results

The results indicate that dual-stage verification can be both fast and selective. In the reported tests, detection-to-alert remained under 10 s, reliability exceeded 95%, and false alarms dropped by more than 90%. Because image analysis runs locally on the Raspberry Pi, the buzzer and web alerts can be triggered immediately, without cloud delay. In practice, the controller links three steps into one workflow: rapid flame sensing, camera-based confirmation, and IoT reporting for real-time visibility.

Future work will strengthen robustness and extend coverage. Adding smoke and temperature sensors can improve early warning when flames are obscured or visibility is poor. We also plan to evaluate lightweight machine learning classifiers to handle more complex scenes and lighting. Finally, a multi-node deployment with coordinated dashboards would improve resilience and coverage for larger residential and industrial facilities.

4. RESULTS AND DISCUSSION

4.1 Experimental setup

The dual-stage fire detection system was evaluated through controlled laboratory experiments to confirm real operating behavior. A complete prototype was assembled, integrating the flame sensor, Raspberry Pi camera module, buzzer, and the IoT web interface described earlier.

Testing was carried out under varied indoor lighting and across multiple flame distances to assess robustness to environmental changes. In each run, a small open flame, using a candle or lighter, was placed at incremental distances from 10 cm up to 2 m. Each experiment was repeated 30 independent times under identical conditions, resulting in a total of 30 trials per test scenario. Performance metrics were logged automatically in the system database through the IoT dashboard for real-time monitoring and post-analysis. The experimental performance results across multiple flame distances are illustrated in Figure 9.

Figure 9 summarizes the empirical trends across flame distance. Detection accuracy stays above 95% up to about 50 cm, then declines gradually as the flame moves farther away. Response time rises slightly with distance but remains under 10 s. False triggers are rare and mainly appear at longer ranges under strong ambient light, as shown by the red bars. The figure also reflects the controlled indoor setup, including flame positioning and the Raspberry Pi sensor camera alignment. Distances, flame intensity, and lighting were kept consistent across trials. All evidence, including sensor activations, vision-confirmed alarms, and database logs, was used for the quantitative evaluation.

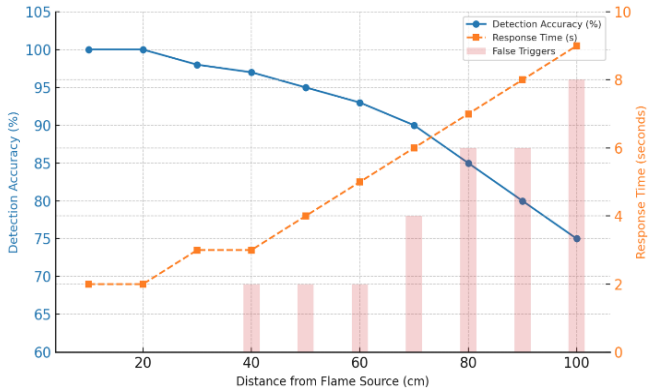

Figure 9. Experimental performance overview showing detection accuracy, response time, and false-alarm behavior across tested flame distances

4.2 Sensor validation

The flame sensor acted as a fast first-stage trigger across the tested distances. In Figure 6, its digital output switches to logic-high almost immediately when a flame appears, and detection typically occurs in under 1 s after ignition, which is consistent with its IR sensitivity range of 760–1100 nm. Within 2 m, the sensor maintained a detection rate above 95%, with noticeable signal weakening mainly beyond that range. A small number of false triggers occurred under strong ambient lighting or near reflective surfaces. These events were logged in the database, but did not escalate to a full alarm because the second-stage verification rejected them. Overall, the sensor works well as a rapid, high-recall trigger, while discrimination is handled by the vision stage.

4.3 Image-based fire confirmation

The second stage validates whether a sensor trigger corresponds to a real fire. After the flame sensor activates, the Raspberry Pi camera captures frames, and the OpenCV pipeline analyzes them to confirm fire evidence before alarms and notifications are issued. The decision-making process of the proposed dual-stage detection algorithm is illustrated in Figure 10, showing the validation flow and event sequence from sensor trigger to alert logging. shows the real-time progression from sensor trigger → image analysis → database logging, including time stamps (t = 0–9 s).

Figure 10 summarizes real-time event handling in the dual-stage detection algorithm. It starts with the flame sensor trigger, then moves through image-based confirmation, alarm activation, database logging, and an automated reset. Each stage follows the response timing measured in the experiments, spanning roughly t = 0 to 9 s. In testing, every real flame that appeared within the camera's field of view was confirmed, giving 100% confirmation accuracy for detected fires. No false confirmations were recorded under common disturbances such as sunlight glare or sudden illumination changes. Image capture and analysis added about 1-2 s, so the total detection-to-alert latency stayed under 10 s.

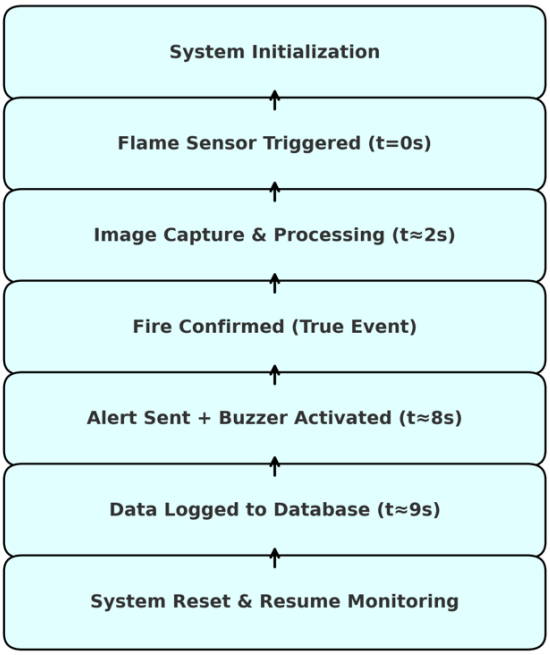

Figure 10. Algorithm validation flow and detection event sequence

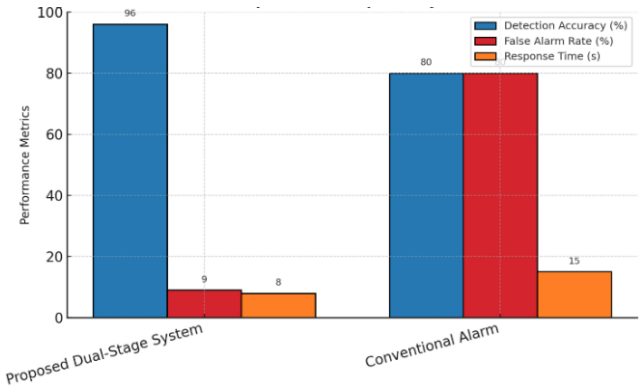

Figure 11. Performance comparison of the proposed system vs. the conventional alarm

In practice, this second stage functions as a verification filter. It confirms true fire events and prevents nuisance triggers from escalating into full alarms. Figure 11 shows the impact of this design through a direct comparison with a conventional alarm.

The figure visually compares the proposed dual-stage Raspberry Pi–based system against a conventional single-stage alarm using three core metrics. Detection accuracy increases from 80% to 96%, false alarms drop from 80% to 9%, and response time improves from 15 s to 8 s. Taken together, the chart shows a clear shift toward practical reliability. The dual-stage verification step keeps alerts fast while filtering out noisy triggers, so the system delivers a stronger balance of speed and trust than the conventional approach.

4.4 Integrated system performance

Integrating sensor-based detection with image-based verification produced a dual-stage pipeline that stays fast while improving accuracy and robustness. Across 30 independent trials, the system achieved a mean detection accuracy of 96.4% (± 1.8%), while the false-alarm rate was reduced by 91.7% relative to the baseline single-sensor configuration. The mean end-to-end response time was 8.6 s (± 0.9 s), measured from flame ignition to alarm activation, with all trials remaining below the 10 s threshold. The mean end-to-end response time was 8.6 s (± 0.9 s), measured from flame ignition to alarm activation, and all trials stayed below 10 s. This keeps the alert latency under the target threshold while still logging and sounding the alarm promptly, which is important for early fire detection scenarios (Table 4). Table 4 summarizes the measured outcomes from the experimental trials and shows that the system detects fire events reliably while keeping false alerts low. In addition to point-in-time tests, we ran the device continuously for 12 hours and observed 98% operational uptime, indicating stable hardware behavior and consistent software execution under sustained use (Table 4).

Table 4. Quantitative performance summary of the proposed fire detection system

Metric	Experimental Observation	Interpretation
Detection accuracy	96.4%	High reliability in recognizing genuine fire events under all test conditions
False-alarm reduction	91.7%	Dual-stage verification eliminates most nuisance triggers (dust, light flicker)
Average response time	8.6 s	Rapid activation from flame ignition to confirmed alarm
System uptime (12 h run)	98.3%	Stable continuous operation; no system crash or communication fault
Data logging success rate	100%	All valid events have been successfully logged to the IoT dashboard and database

Figure 12 shows the IoT dashboard used during testing. It displays real-time system states, such as normal, sensor-triggered, and fire confirmed, and it records event IDs, timestamps, device source, and verification outcomes. All entries were synchronized automatically to a MySQL database to support traceability and remote monitoring.

Figure 13 analyzes those database logs over time. Sensor-triggered events and confirmed fire incidents follow a consistent sequence, with only a small number of filtered anomalies, which reflects a low-noise detection process. The takeaway is that the system not only detects and verifies reliably, it also logs events in a way that supports auditing and post-incident review.

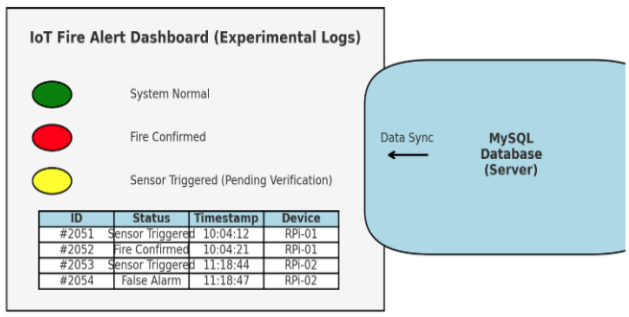

Figure 12. IoT alert logs and database visualization during the experiment

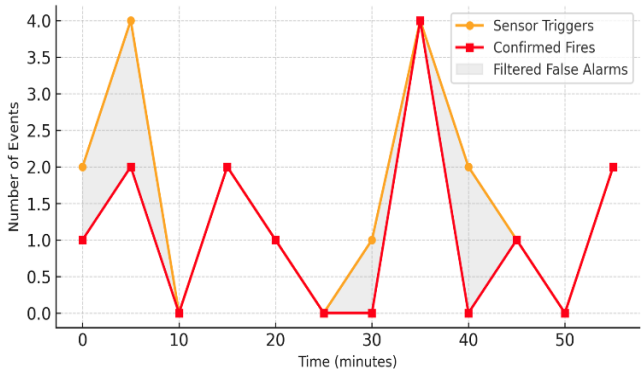

Figure 13. Database event frequency over time during the experiment

4.5 Comparison with previous work

The reported results in the referenced studies were produced under different conditions, including flame type, distance, lighting, environment, and evaluation protocol. Because of these differences, the metrics in Table 5 are used for qualitative benchmarking, not direct one-to-one comparison. This comparison is included to place the proposed system in context against representative approaches in the literature. It is not intended to imply identical test setups or strictly equivalent numerical performance across all studies.

Recent work has reported strong fire recognition using multi-modal fusion and deep learning, such as CNN-based video analysis, but these methods often assume higher compute, large labeled datasets, and continuous video processing, which can be difficult to sustain on low-cost edge devices in real time [29]. In contrast, the proposed system keeps processing lightweight by using a flame sensor for rapid triggering and vision only for confirmation, so computation is spent only when an event is suspected. The goal is operational response, verified detection, and IoT alerting, rather than modeling fire spread in buildings or measuring user-level safety awareness [30].

The comparison also clarifies the trade-offs across common designs. Single-sensor alarms respond quickly but can generate frequent false triggers because there is no verification step. Camera-only systems can improve detection specificity, but continuous video analysis increases compute load and can add latency. The proposed dual-stage workflow avoids that trade-off by using fast sensing for the initial trigger and running a short vision confirmation only when needed. In our tests, false alarms fell by about 90%, while accuracy and response time remained competitive, using a simple two-component design suited to low-cost IoT deployment.

Table 5. Comparative performance analysis with existing fire detection approaches

Study / System	False Alarm Rate	Response Time	IoT / Network Capability	Remarks
Thengade et al. [15]	25%	~15 s	No	Good accuracy but frequent false positives; no remote alerting
Dhanujalakshmi et al. [17]	18%	~12 s	Yes	Real-time image alerts; bandwidth-heavy; no multi-sensor logic
Wu et al. [20]	10%	20–30 s	Yes	Reliable but slower; high cost and setup complexity
BRE Group [25]	8%	~10 s	Partial	High cost; commercial-grade system
Proposed system	< 10%	< 10 s	Full (IoT + Web Dashboard)	High accuracy (> 95%), scalable, economical, and IoT-ready

4.6 Discussion of findings

The findings support the two-tier verification approach as both fast and dependable. Across the reported experiments, accuracy exceeded 95%, end-to-end response stayed under 10 s, and false alarms dropped by roughly 90% compared with single-stage triggering. Three strengths are clear. The hardware is low-cost, based on a Raspberry Pi and off-the-shelf sensors. The IoT layer delivers real-time alerts and complete logs, which makes incidents traceable and actionable. The modular structure also keeps upgrades straightforward, whether that means adding sensors or enhancing verification with AI when resources allow.

The low standard deviation observed across trials indicates stable system behavior and consistent detection performance under repeated experimental conditions. Figures 12 and 13 show stable behavior across repeated runs, with consistent logging and verification outcomes, which support reliability for practical deployment in homes and small industrial sites. The main limitations are expected: detection depends on distance and a clear line of sight, so larger or obstructed spaces may require multiple units for coverage. The current prototype also targets open-flame detection, so adding smoke or temperature sensing would strengthen early warning in smoldering scenarios. Future work can improve adaptability and scale. One path is machine learning–based image classification for more complex lighting and backgrounds. Another is a networked multi-node design that coordinates several devices for wider-area monitoring. The validated results indicate a good balance of speed, verification, and IoT reporting, while the next gains will come from better coverage and broader sensing.

4.6.1 Discussion of results

Although evaluation was conducted indoors under controlled conditions, deployment realities were considered in the design. In real buildings, motion, clutter, smoke occlusion, and shifting lighting can introduce noise, but requiring both a sensor trigger and visual confirmation helps suppress transient artifacts. Network issues are handled by prioritizing local alarms first, so the buzzer activates even during connectivity drops. Power is also managed by keeping the camera off in standby and enabling it only after a trigger. For installation, an enclosure with an appropriate IP rating is recommended to protect against dust, humidity, and heat.

5. CONCLUSIONS

This study designed and validated a dual-stage, IoT-enabled

fire detection system built on a Raspberry Pi. An infrared flame sensor provides rapid triggering, then an image-processing module confirms the event before the alarm is escalated. In the reported experiments, detection accuracy exceeded 95%, end-to-end response stayed under 10 s, and false alarms dropped by about 90% compared with single-sensor alarms. The web dashboard added practical value by supporting real-time monitoring, automatic event logging, and immediate notifications, which improve situational awareness and response coordination. These results indicate that low-cost hardware can produce dependable fire alerts when verification is part of the workflow. The current design still has limits, mainly line-of-sight dependence and a focus on open-flame detection, but the modular architecture makes upgrades straightforward, such as adding smoke or temperature sensing or introducing AI-based classification for more complex scenes. The takeaway is that the proposed controller delivers a workable balance of speed, accuracy, and affordability for smart building safety.

Future extensions will follow what we observed during testing. On the vision side, we will target the rare false triggers caused by strong reflections and sudden lighting changes by adopting adaptive HSV thresholds and simple temporal stability checks, where a flame region must persist across consecutive frames before confirmation. We will also harden communication under weak or congested networks. Based on the measured response and upload delays, the next version will prioritize local-first alarms, then use buffered IoT transmission so alerts and logs are still delivered reliably when connectivity recovers. Finally, adding smoke or temperature sensing will expand coverage to smoldering fires that may not produce a clear open flame.

ACKNOWLEDGMENT

The authors express their gratitude to the Centre for Research and Innovation Management (CRIM) at Universiti Teknikal Malaysia Melaka (UTeM) for their valuable support in this research.

CONTRIBUTIONS

The authors' contributions are as follows: "Conceptualization, J.A.J.A and A.W.Y.K.; Methodology, J.A.J.A.; Software, I.; Validation, M.F. and A.W.Y.K.; Formal analysis, I.; Investigation, J.A.J.A; Resources, M.F.; Writing—original draft preparation, J.A.J.A and S.G.H.; Writing—review and editing, S.G.H. and I.; Funding acquisition, I. and S.G.H. All authors have read and agreed to the published version of the manuscript.

DATA AVAILABILITY STATEMENT

All the datasets used in this study are available from the Zenodo database (accession number: https://zenodo.org/records/17279434).

REFERENCES

[1] Jose, T., Devi, R.M., Darshini, P.S., Mathew, S., Selvaraj, D. (2019). Raspberry Pi based – A cyber defensive industrial control system with redundancy and intrusion detection. SSRG International Journal of Computer Science and Engineering (SSRG-IJCSE), 1-6. https://www.internationaljournalssrg.org/uploads/specialissuepdf/NCTCT19/2019/CSE/paper1.pdf.

[2] Lee, C.H., Lee, W.H., Kim, S.M. (2023). Development of IoT-based real-time fire detection system using Raspberry Pi and fisheye camera. Applied Sciences, 13(15): 8568. https://doi.org/10.3390/app13158568

[3] Avazov, K., Mukhiddinov, M., Makhmudov, F., Cho, Y.I. (2021). Fire detection method in smart city environments using a deep-learning-based approach. Electronics, 11(1): 73. https://doi.org/10.3390/electronics11010073

[4] Zhang, L., Li, J., Zhang, F. (2023). An efficient forest fire target detection model based on improved YOLOv5. Fire, 6(8): 291. https://doi.org/10.3390/fire6080291

[5] Kim, S.Y., Muminov, A. (2023). Forest fire smoke detection based on deep learning approaches and unmanned aerial vehicle images. Sensors, 23(12): 5702. https://doi.org/10.3390/s23125702

[6] Xu, F., Zhang, X., Deng, T., Xu, W. (2023). An image-based fire monitoring algorithm resistant to fire-like objects. Fire, 7(1): 3. https://doi.org/10.3390/fire7010003

[7] Li, X., Cao, Z., Guo, J., Liu, Z. (2025). Exploration on communication technologies for automatic fire alarm systems in the petrochemical industry. In 2025 5th International Conference on Consumer Electronics and Computer Engineering (ICCECE), Dongguan, China, pp. 590-594. https://doi.org/10.1109/ICCECE65250.2025.10985202

[8] Wang, Y. (2022). Simulation study of fuzzy neural control fire alarm system based on clustering algorithm. In 2022 2nd International Conference on Algorithms, High Performance Computing and Artificial Intelligence (AHPCAI), Guangzhou, China, pp. 217-221. https://doi.org/10.1109/AHPCAI57455.2022.10087561

[9] Acakpovi, A., Ayitey, D.T., Adjaloko, E.N. (2021). Innovative fire detection and alarm system for sustainable city development. In 2021 International Conference on Cyber Security and Internet of Things (ICSIoT), France, pp. 30-36. https://doi.org/10.1109/ICSIoT55070.2021.00015

[10] Charumporn, B., Omatu, S., Yoshioka, M., Fujinaka, T. (2004). Fire alarm systems using electronic nose systems. IFAC Proceedings Volumes, 37(12): 219-223. https://doi.org/10.1016/s1474-6670(17)31471-4

[11] Bino, J., Islam, M.M., Boppana, U.M., Hossain, M.A.M., Haque, M.F., Fatme, N., Manjula, C. (2024). FireEye: An IoT-based fire alarm and detection system for enhanced safety. In 2024 Intelligent Systems and Machine Learning Conference (ISML), Hyderabad, India, pp. 58-62. https://doi.org/10.1109/ISML60050.2024.11007412

[12] Dong, W.H., Wang, L., Yu, G.Z., Mei, Z.B. (2016). Design of wireless automatic fire alarm system. Procedia Engineering, 135: 413-417. https://doi.org/10.1016/j.proeng.2016.01.149

[13] Telny, A.V., Monakhov, M.Y., Nikolaev, A.V., Matveeva, E.A. (2024). Prediction of false alarms of security alarm systems for centrally guarded facilities based on the use of a neural network. In 2024 Dynamics of Systems, Mechanisms and Machines (Dynamics),

Omsk, Russian Federation, pp. 1-5. https://doi.org/10.1109/Dynamics64718.2024.10838669
[14] Custance, N.D.E. (1988). Perimeter detection systems: Correlation of false alarm cause with environmental factors. In Proceedings of the 1988 International Carnahan Conference on Security Technology, Crime Countermeasures (CCST), Lexington, KY, USA, pp. 89-97. https://doi.org/10.1109/CCST.1988.75995
[15] Thengade, A.M., Mishra, P., Kshatriya, R., Mhaskar, R., Bodhe, P. (2019). Fire detection using image processing using Raspberry Pi. https://www.semanticscholar.org/paper/Fire-Detection-Using-Image-Processing-Using-PI-Thengade-Mishra/c36d259933a58cea149e8a0fd8f92290e3673728.
[16] Khan, M.N.A., Tanveer, T., Khurshid, K., Zaki, H., Zaidi, S.S.I. (2019). Fire detection system using Raspberry Pi. In 2019 International Conference on Information Science and Communication Technology (ICISCT), Karachi, Pakistan, pp. 1-6. https://doi.org/10.1109/CISCT.2019.8777414
[17] Dhanujalakshmi, R., Divya, B., Sandhiya, C.D., Robertsingh, A. (2017). Image Processing based fire detection system using Raspberry Pi system. SSRG International Journal of Computer science and Engineering (JCSE), 4(4): 18-22.
[18] Wong, A.K.K., Fong, N.K. (2014). Experimental study of video fire detection and its applications. Procedia Engineering, 71: 316-327. https://doi.org/10.1016/j.proeng.2014.04.046
[19] Almoallem, Y.D., Moghimi, M.J., Jiang, H. (2017). Black silicon based iris with reduced light scattering and reflection. In 2017 International Conference on Optical MEMS and Nanophotonics (OMN), Santa Fe, NM, USA, pp. 1-2. https://doi.org/10.1109/OMN.2017.8051469
[20] Wu, Q., Cao, J., Zhou, C., Huang, J., et al. (2017). Intelligent smoke alarm system with wireless sensor network using ZigBee. Wireless Communications and Mobile Computing, 2018(1): 8235127. https://doi.org/10.1155/2018/8235127
[21] Alsayaydeh, J.A.J., Irianto, Ali, M.F., Al-Andoli, M.N.M., Herawan, S.G. (2024). Improving the robustness of IoT-powered smart city applications through service-reliant application authentication technique. IEEE Access, 12: 19405-19417. https://doi.org/10.1109/ACCESS.2024.3361407
[22] Al-Andoli, M.N., Irianto, Alsayaydeh, J.A., Alwayle, I.M., Che Ku Mohd, C.K.N., Abuhoureyah, F. (2024). Robust overlapping community detection in complex networks with graph convolutional networks and fuzzy C-means. IEEE Access, 12: 70129-70145. https://doi.org/10.1109/ACCESS.2024.3399883
[23] Keano, N.L.S., Jose, C.A.J. (2024). IoT-enabled fire alarm system with cloud-based storage and monitoring. International Journal of Latest Technology in Engineering Management & Applied Science, 13(9): 188-196. https://doi.org/10.51583/ijltemas.2024.130919
[24] Alsayaydeh, J.A.J., Irianto, Zainon, M., Baskaran, H., Herawan, S.G. (2022). Intelligent interfaces for assisting blind people using object recognition methods. International Journal of Advanced Computer Science and Applications (IJACSA), 13(5): 84-92. https://doi.org/10.14569/IJACSA.2022.0130584
[25] BRE Group. (2019). Effectiveness of multi-sensor fire detectors in reducing false alarms. Building Research Establishment, UK. https://www.bregroup.com.
[26] Shkarupylo, V., Alsayaydeh, J.A.J., Yusof, M.F.B., Oliinyk, A., Artemchuk, V., Herawan, S.G. (2024). Exploring the potential network vulnerabilities in the smart manufacturing process of Industry 5.0 via the use of machine learning methods. IEEE Access, 12: 152262-152276. https://doi.org/10.1109/ACCESS.2024.3474861
[27] Alsayaydeh, J.A.J., Irianto, Aziz, A., Xin, C.K., Hossain, A.K.M.Z., Herawan, S.G. (2022). Face recognition system design and implementation using neural networks. International Journal of Advanced Computer Science and Applications (IJACSA), 13(6): 63-69. https://doi.org/10.14569/IJACSA.2022.0130663
[28] Shkarupylo, V., Blinov, I., Chemeris, A., Dusheba, V., Alsayaydeh, J.A.J., Oliinyk, A. (2021). Iterative approach to TLC model checker application. In 2021 IEEE 2nd KhPI Week on Advanced Technology (KhPIWeek), Kharkiv, Ukraine, pp. 283-287. https://doi.org/10.1109/KhPIWeek53812.2021.9570055
[29] Devi, K.L.S.R., Kumar, G.N., Narayana, P.A., Ramana, K.V., Amarendra, K., Gullipalli, T.R. (2024). Forest fire prediction and management using AI (artificial intelligence), ML (machine learning) and deep learning techniques. In 2024 8th International Conference on Inventive Systems and Control (ICISC), Coimbatore, India, pp. 324-327. https://doi.org/10.1109/ICISC62624.2024.00062
[30] Sholanke, A.B., Ekhaese, E.N., Ekundayo, P.A. (2024). Users' knowledge of fire safety measures in educational environment: A case study of a college building in Nigeria. International Journal of Safety & Security Engineering, 14(1): 135-143. https://doi.org/10.18280/ijsse.140113