Modern Applied Science; Vol. 19, No. 2; 2025
ISSN 1913-1844 E-ISSN 1913-1852
Published by Canadian Center of Science and Education

# An IoT-Based Framework for Wildfire Detection Using Multi-Sensors Integration and CNN Image Classification

Nurul Azma Zakaria[1], Hani Safwan Mohd Isha[1], Fairul Azni Jafar[2], Zaheera Zainal Abidin[3], Mohd Rizuan Baharon[1], Wan Faezah Abbas[4] & Nor Hidayah Arsyad[5]

[1] Fakulti Teknologi Maklumat dan Komunikasi, Universiti Teknikal Malaysia Melaka, Hang Tuah Jaya, 76100 Durian Tunggal, Melaka, Malaysia

[2] Fakulti Teknologi dan Kejuruteraan Industri dan Pembuatan, Universiti Teknikal Malaysia Melaka, Hang Tuah Jaya, 76100 Durian Tunggal, Melaka, Malaysia

[3] Fakulti Kecerdasan Buatan dan Keselamatan Siber, Universiti Teknikal Malaysia Melaka, Hang Tuah Jaya, 76100 Durian Tunggal, Melaka, Malaysia

[4] Fakulti Sains Komputer & Matematik, Universiti Teknologi MARA, 40450 Shah Alam, Selangor, Malaysia

[5] SB Tape Group Sdn. Bhd., Lot 1812, Jalan KPB 1, Kampung Baru Balakong, 43300 Seri Kembangan, Selangor, Malaysia

Correspondence: Nurul Azma Zakaria, Fakulti Teknologi Maklumat dan Komunikasi, Universiti Teknikal Malaysia Melaka, Hang Tuah Jaya, 76100 Durian Tunggal, Melaka, Malaysia. E-mail: azma@utem.edu.my

Received: August 19, 2025 Accepted: October 25, 2025 Online Published: October 27, 2025

doi:10.5539/mas.v19n2p97 URL: https://doi.org/10.5539/mas.v19n2p97

**Abstract**

Wildfires are a growing threat to ecosystems, property, and human lives, especially in rural and forest-adjacent areas where monitoring infrastructure is limited. Traditional detection methods, such as satellite imaging and human surveillance, often suffer from delayed response and low precision during early fire stages. This study proposes a novel IoT-based wildfire detection framework that combines multi-sensor data with deep learning for rapid and localized fire identification. The system integrates smoke and flame sensors with a YOLOv4-based convolutional neural network (CNN) for image classification, all deployed on a Raspberry Pi 5 platform. A dual-layer detection mechanism enables immediate threshold-based alerts and visual confirmation via AI-driven analysis. Real-time notifications are delivered through a Telegram bot, while environmental data are logged and visualized using the ThingSpeak dashboard. The system, developed in Python, is optimized for deployment in low-resource environments. Experimental results demonstrate high detection accuracy and reliable performance across diverse conditions. This work demonstrates the practical potential of lightweight, AI-enhanced IoT systems for early wildfire detection and offers a scalable solution for remote monitoring. Future enhancements will explore more efficient CNN architectures and predictive analytics for proactive fire management.

**Keywords:** wildfire detection, Internet of Things (IoT), multi-sensors, Convolutional Neural Networks (CNN), YOLOv4, environmental monitoring

## 1. Introduction

Wildfires are increasingly recognized as one of the most devastating environmental disasters, driven by climate change, prolonged droughts, and deforestation. Their impact extends beyond ecological destruction to include severe economic loss, displacement of communities, and long-term health consequences due to poor air quality. According to global reports, wildfire events are increasing in frequency and severity, often surpassing the capacity of traditional mitigation and response strategies. This escalation reflects a critical gap in current detection and alert mechanisms that, if unaddressed, will continue to result in preventable damage and loss. Current fire detection methods, such as satellite imaging (Wickramasinghe et al., 2018; Mo et al., 2023; Lakshmanaswamy et al., 2024; Yu et al., 2024; Wu et al., 2025; Narvaez Luces et al., 2025), manual surveillance, and camera-based monitoring (Pundir et al., 2018; Shakhnoza et al., 2022; Deepika et al., 2025) are limited by delayed responsiveness, high maintenance costs, and geographic coverage issues. While there have been significant advancements in wildfire modeling and prediction, these solutions frequently depend on data-intensive platforms or high-powered computing infrastructure, making them impractical for deployment in

remote or under-resourced regions.

There is a growing consensus that timely detection and rapid alerting can substantially reduce the spread and impact of wildfires. This has led researchers to explore IoT-based systems as a means of creating responsive, autonomous fire monitoring solutions. However, much of the existing literature either focuses on advanced machine learning models, which are computationally expensive, or presents systems without real-time, actionable alert capabilities. Several recent studies have addressed the integration of sensor-based monitoring with cloud platforms to detect environmental hazards, including fire. This study responds to the urgent need for practical, low-power, and field-deployable wildfire detection solutions by synthesizing a lightweight, real-time alert system that operates on Raspberry Pi using multi-sensors with a convolutional neural network (CNN) image classification algorithm. Unlike previous work, it emphasizes operational simplicity, edge computing compatibility, and integration with real-time communication platforms like Telegram and ThingSpeak.

This study is guided by the following research questions: (RQ1) How effectively does the proposed algorithm, implemented on a Raspberry Pi, detect wildfire indicators in real time? (RQ2) To what extent does the IoT-based system deliver timely and actionable alerts through Telegram and ThingSpeak across different environmental conditions? The research design is application-oriented. The system was implemented and tested in two scenarios (indoor and outdoor) to answer the research questions.

The paper is organized as follows. Section 2 presents the related works in monitoring and alert systems; Section 3 elaborates on the methodology used in developing the proposed system. Section 4 analyses the results and discusses them accordingly. Section 5 concludes the paper with an outline of contributions and future work.

## 2. Related Works

Numerous studies have explored the development of monitoring and alert systems using various technologies. However, an analysis of recent literature, as summarized in Table 1 and Figure 1, reveals several critical limitations in functionality and system integration. This section evaluates 20 selected research works based on the integration of six (6) key functionalities: Cloud integration, Notification systems, Machine Learning, Alarm systems, Multi-sensor integration, and Web-based platforms. These features are pivotal in developing intelligent, responsive, and scalable solutions, particularly within IoT-based systems for monitoring, safety, and decision support. The analysis reveals varying levels of implementation across these functionalities, highlighting both prevalent practices and significant research gaps.

Cloud integration was implemented in 12 of the 20 studies, including those by Arun et al. (2023), Pavitra et al. (2020), and Saranya et al. (2020). These studies leveraged cloud technologies to support scalability, remote access, and centralized data management. However, other works, such as Lee et al. (2023) and Mokhtar'udin and Lias (2024), did not include cloud capabilities, likely due to constraints such as data privacy, latency, or infrastructure cost. This suggests a growing need for hybrid architectures that combine local processing with cloud-based services to optimize performance in resource-constrained or mission-critical environments.

Notification systems were among the most widely implemented features, appearing in 18 studies. Works such as Ehsan et al. (2022), Dasari et al. (2019), and Abdul Rahman et al. (2025) emphasized the importance of real-time alerts in improving user responsiveness and system interactivity. Notifications serve as a crucial communication bridge between system intelligence and end-user action. Despite their widespread use, future systems could benefit from more intelligent, context-aware, or predictive notification mechanisms to tailor alerts based on user behavior and environmental conditions.

In contrast, Machine Learning (ML) was the least utilized functionality, found in only one study, i.e Lee et al. (2023). This underrepresentation may stem from implementation complexity, limited access to large datasets, or a tendency to rely on rule-based or threshold-based logic, which restricts their adaptability and intelligence. Nevertheless, the potential of ML to support intelligent decision-making, anomaly detection, and system personalization presents a compelling research opportunity. With the increasing availability of lightweight ML models and edge computing solutions, future work should explore ML integration to enhance adaptability and predictive capabilities.

Alarm systems were present in 14 studies, including those by Parida (2021), Praveenkumar and Ram (2021), and Mahalingam et al. (2021). Alarms play a vital role in safety-critical applications, providing immediate alerts in emergencies. Unlike standard notifications, alarms are more disruptive and demand immediate user attention. However, several studies, such as Krishnamoorthy et al. (2023) and Kaura et al. (2020), did not include alarm functionality, which may reduce system effectiveness in time-sensitive scenarios. Future enhancements should focus on developing adaptive alarm mechanisms with escalation protocols and user customization to improve

responsiveness.

Multi-sensor integration emerged as one of the most consistently incorporated features, appearing in 18 studies. Examples include Ajith et al. (2020), Kadir et al. (2019), and Jyoshana et al. (2021). These systems utilize sensor fusion to provide richer, more accurate data, reducing false positives and increasing environmental awareness. The frequent implementation of multi-sensor setups highlights their value in enhancing system robustness and reliability. Nevertheless, future research could explore advanced features such as dynamic sensor selection, self-calibration, and adaptive fusion algorithms to further optimize sensor performance in complex and changing environments.

Lastly, web-based platforms were featured in only 7 studies, including Arun et al. (2023), Pavitra et al. (2020), and Mokhtar'udin and Lias (2024). While mobile-based apps are sometimes utilized, the absence of a web interface limits accessibility, especially in institutional or multi-user scenarios where centralized monitoring is required. The limited integration suggests a continued focus on embedded or local systems, possibly due to concerns over connectivity, device constraints, or security. However, this represents a key area for development. Future systems could benefit from cross-platform, web-based dashboards that enhance user engagement, support real-time monitoring, system visualization, and provide scalable interaction capabilities.

Overall, the critical analysis reveals an imbalance in feature integration across existing works, with none offering a fully comprehensive hybrid monitoring system. While some functionalities like multi-sensor systems and notifications are commonly employed, others, particularly ML and web platforms, are underexplored. The proposed study addresses these gaps by holistically integrating all six (6) core features. By integrating these essential features into a unified and cohesive IoT-based framework, the proposed work addresses the clear gaps in previous studies while establishing a new benchmark for intelligent, scalable, and user-centric wildfire detection systems. It offers a well-rounded solution that balances local responsiveness with global accessibility, demonstrating strong potential for real-world deployment in high-risk and remote environments.

Table 1. Functionality Comparison of Related Works

| Author (Year) | Cloud | Notification | Machine Learning | Alarm | Multi-sensors | Web-based |
|---|---|---|---|---|---|---|
| Abdul Rahman et al. (2025) | - | / | - | / | / | - |
| Ajith et al. (2020) | / | - | - | - | / | - |
| Arun et al. (2023) | / | / | - | / | / | / |
| Dasari et al. (2019) | - | / | - | / | / | - |
| Ehsan et al. (2022) | - | / | - | / | / | - |
| Jyoshana et al. (2021) | / | / | - | - | / | - |
| Kadir et al. (2019) | / | / | - | - | / | - |
| Kaura et al. (2020) | - | / | - | - | / | - |
| Krishnamoorthy et al. (2023) | / | - | - | - | / | / |
| Lakkar and Narayan (2020) | / | / | - | / | / | - |
| Lee et al. (2023) | - | - | / | - | - | - |
| Mahalingam et al. (2021) | / | / | - | / | / | - |
| Mokhtar'udin and Lias (2024) | - | / | - | / | / | / |
| Parida, D. (2021) | / | - | - | / | - | - |
| Pavitra et al. (2020) | / | / | - | - | / | / |
| Praveenkumar and Ram (2021) | / | / | - | / | / | / |
| Sahal et al. (2021) | - | - | - | - | / | - |
| Saranya et al. (2020) | / | / | / | / | / | - |
| Sonti et al. (2022) | - | / | - | / | - | - |
| Srinithi and Dhivya (2020) | - | / | - | / | / | - |

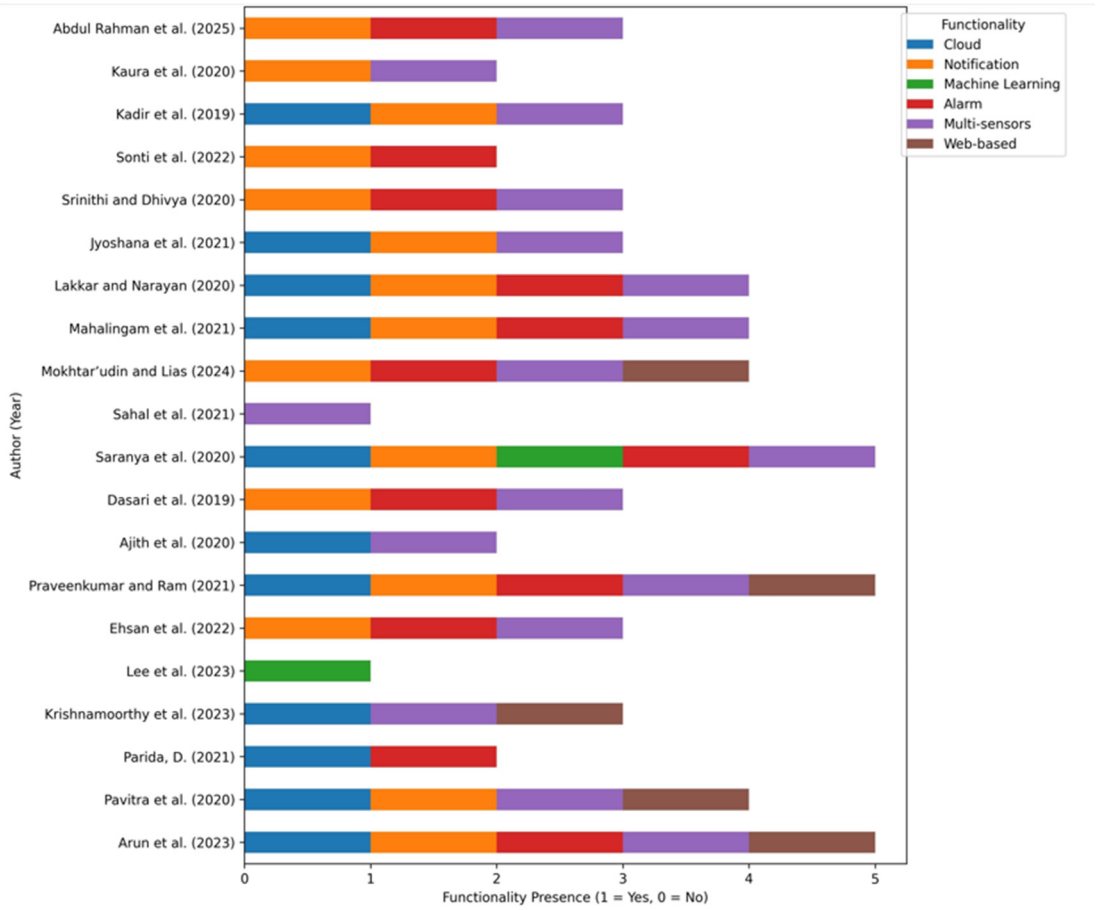



Figure 1. Functionality Comparison of Related Works

## 3. Method

This section presents the design and development of the proposed system, detailing its structural and functional components. The system architecture is described to illustrate the interaction between hardware and software components, as well as the data flow across modules. A flowchart is included to visualize the logical sequence of operations, while pseudocode is used to convey the core algorithms and decision-making processes implemented within the system. These elements collectively support a comprehensive understanding of the system's design and implementation.

### *3.1 System Design*

The system architecture, as shown in Figure 2, presents the system architecture, detailing the structural layout and the interaction between hardware and software components, including data communication and processing mechanisms. In addition, Figure 3 illustrates the hardware design of the system, showing how the flame sensor, smoke sensor, camera module, and other peripherals are connected to the Raspberry Pi via GPIO pins; whereas Figure 4 provides the circuit diagram, providing a schematic representation of the electrical connections and components involved. This detailed design supports the integration of multi-sensor input and image-based classification, enabling reliable and responsive wildfire detection.

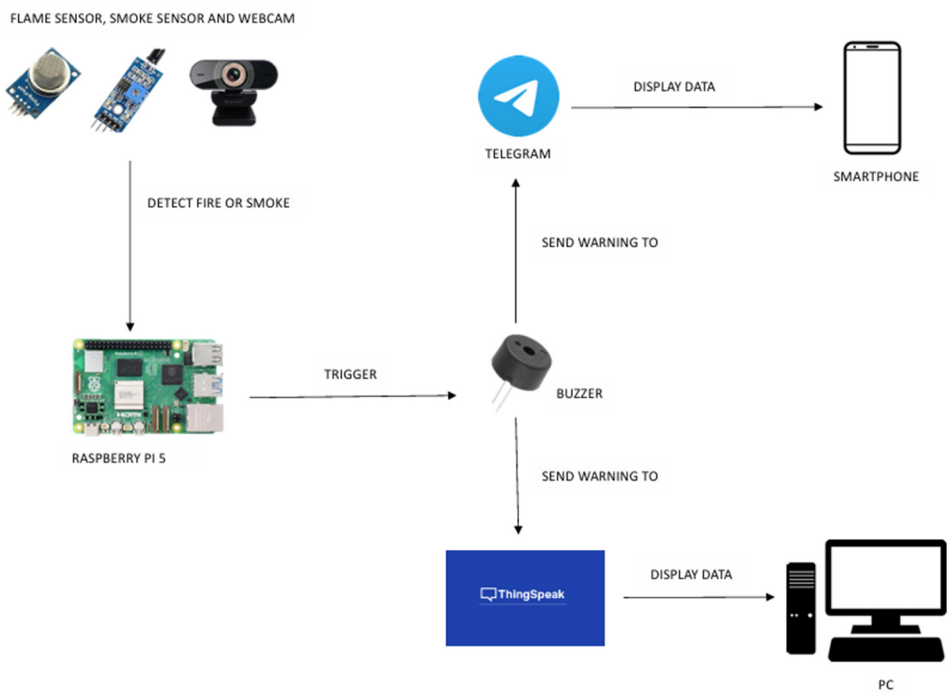



Figure 2. System Architecture

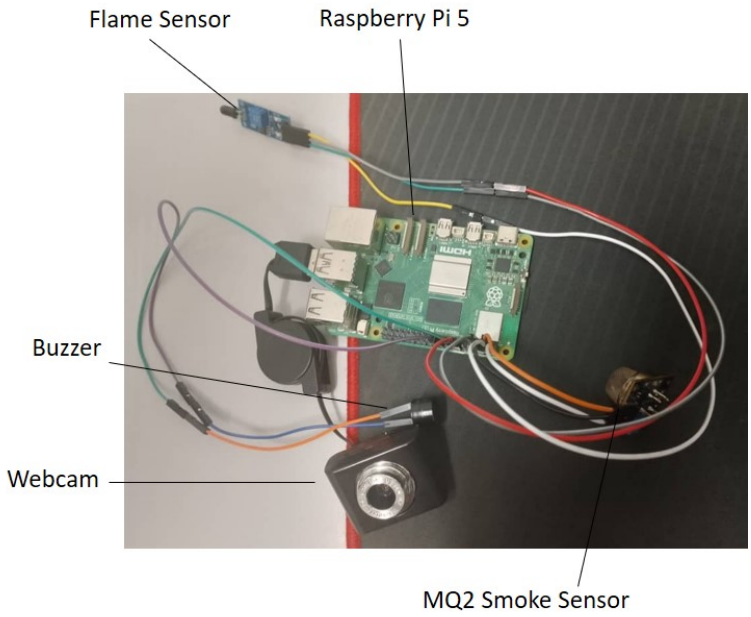



Figure 3. Hardware Design of the System

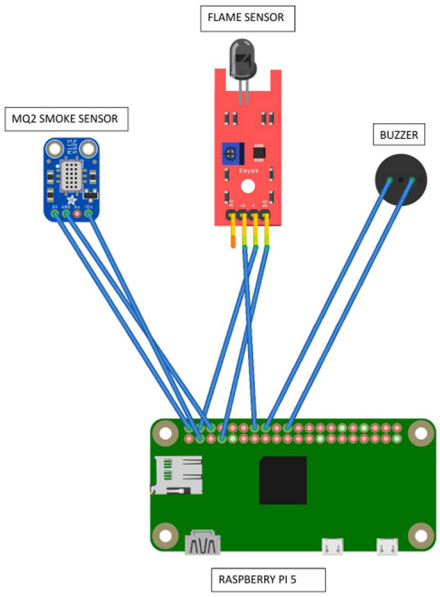



Figure 4. Circuit Diagram

*3.2 User Interface Design*

The user interface components of the system consist of several integrated elements designed to enhance accessibility and usability. A web-based visualization dashboard, as shown in Figure 5, powered by ThingSpeak, provides real-time data and system status and is accessible via both PC and mobile devices. The system also includes a reporting interface, as illustrated in Figure 6, which supports real-time data logging and displays sensor readings along with their corresponding timestamps. Specifically, Field 1 records flame detection using a flame sensor, while Field 2 captures smoke levels using a smoke sensor. Users can view these data reports for monitoring purposes and export them in CSV or Excel file formats for further analysis. In addition, a Telegram-based alert notification interface, as depicted in Figure 7, enables users to receive timely updates and critical alerts directly on their mobile devices.

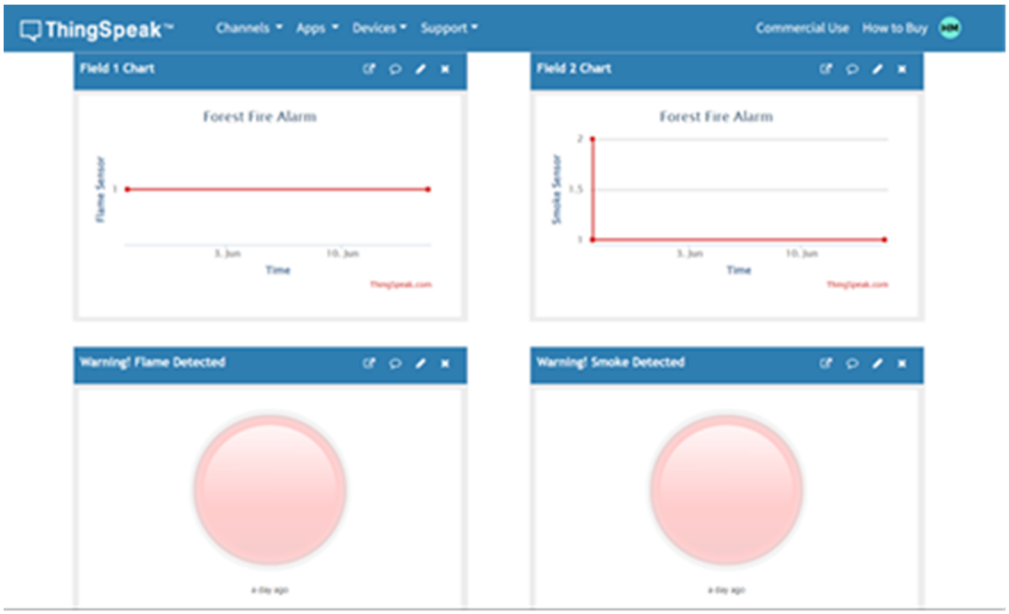



Figure 5. Sample of Visualization using ThingSpeak Dashboard

| | A | B | C | D |
|---|---|---|---|---|
| 1 | created_at | entry_id | field1 | field2 |
| 2 | 2024-05-27 15:12:26 UTC | 226 | 1 | 1 |
| 3 | 2024-05-27 15:12:46 UTC | 227 | 1 | 1 |
| 4 | 2024-05-27 15:13:05 UTC | 228 | 1 | 1 |
| 5 | 2024-05-27 15:13:31 UTC | 229 | 2 | 1 |
| 6 | 2024-05-27 15:13:47 UTC | 230 | 2 | 1 |
| 7 | 2024-05-27 15:14:02 UTC | 231 | 1 | 1 |
| 8 | 2024-05-27 15:14:21 UTC | 232 | 1 | 1 |
| 9 | 2024-05-27 15:14:40 UTC | 233 | 1 | 1 |
| 10 | 2024-05-27 15:14:59 UTC | 234 | 1 | 1 |
| 11 | 2024-05-27 15:15:18 UTC | 235 | 1 | 1 |
| 12 | 2024-05-27 15:15:33 UTC | 236 | 1 | 1 |
| 13 | 2024-05-27 15:15:52 UTC | 237 | 1 | 1 |
| 14 | 2024-05-27 15:16:12 UTC | 238 | 1 | 1 |
| 15 | 2024-05-27 15:16:31 UTC | 239 | 1 | 1 |
| 16 | 2024-05-27 15:16:50 UTC | 240 | 1 | 1 |

Figure 6. Sample of Data Displayed on Reporting Interface




Figure 7. Sample of Alert Notification Displayed on Telegram Channel

*3.3 System Flowchart*

A clear understanding of the system's internal workflow is essential for interpreting its functionality and behavior. This subsection introduces the operational flowchart, which outlines the logical sequence of activities and decision-making steps carried out by the system. As presented in Figure 8, the flowchart visually represents the progression from sensor input to data processing, condition evaluation, and alert generation. Each step corresponds to a specific function or decision point that defines how the system responds under various conditions.

The proposed framework combines multi-sensor input with image classification using a pre-trained YOLOv4 convolutional neural network (Bayegizova et al., 2024) to improve detection accuracy and responsiveness. The system initializes flame and smoke sensors via GPIO, a camera module through OpenCV, and connects to external services like Telegram for alerts and ThingSpeak for cloud-based data updates. It continuously captures and analyzes images for fire or smoke while simultaneously monitoring sensor readings. Detection from either source triggers alerts via Telegram, activates a local alarm, and uploads data to ThingSpeak. Detected images are stored for future model training, and the system operates in a continuous loop, making it suitable for autonomous deployment in wildfire-prone areas.

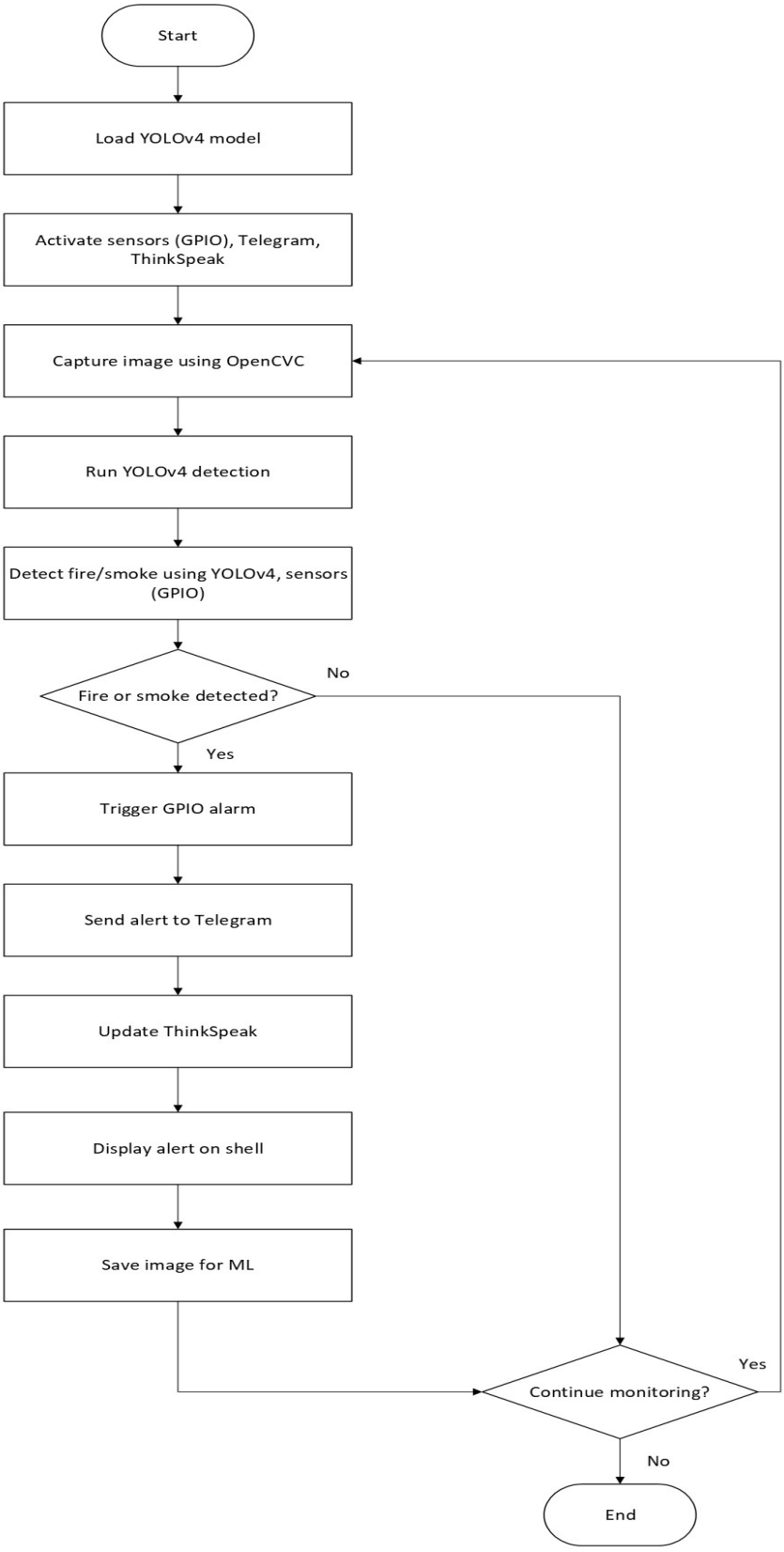



Figure 8. Flow of the System

*3.4 Detection Algorithm Implementation*

The detection algorithm was implemented in Python and deployed on a Raspberry Pi running Raspbian OS. It integrates multi-sensor input, specifically flame and smoke sensors, with image-based classification using the

pre-trained YOLOv4 convolutional neural network. This hybrid approach enables the system to perform parallel evaluation of environmental conditions and visual cues, improving detection reliability. The algorithm is designed to process sensor readings and image data in real time, applying conditional logic to determine whether a fire or smoke event has occurred. Upon detection, the system executes predefined actions such as triggering alerts, activating alarms, and logging data. The algorithm also includes routines for image storage to support future model refinement. The overall computational logic is structured for efficiency and responsiveness, and is further detailed through pseudocode in Figure 9.

```
BEGIN
  // Initialize and Load Components
  Initialize GPIO pins for flame and smoke sensors
  Initialize camera module (OpenCV)
  Load YOLOv4 object detection model (pre-trained on fire/smoke images)
  Initialize Telegram API
  Initialize ThinkSpeak API

  LOOP
    // Step 1: Capture image from camera
    image ← CaptureImageUsingOpenCV()
    // Step 2: Detect fire/smoke in image using YOLOv4
    detections ← RunYOLOv4Detection(image)
    isFireOrSmokeInImage ← CheckDetections(detections, classes=["fire", "smoke"])
    // Step 3: Read flame and smoke sensor values
    flameSensorStatus ← ReadFlameSensor(GPIO_PIN)
    smokeSensorStatus ← ReadSmokeSensor(GPIO_PIN)
    // Step 4: Decision Making
    IF isFireOrSmokeInImage == TRUE OR flameSensorStatus == HIGH OR smokeSensorStatus == HIGH
THEN

        // Step 5: Alert and Response
        SendAlertToTelegram("Wildfire Detected!")
        TriggerAlarm(GPIO_ALARM_PIN)
        UpdateThinkSpeak(flameSensorStatus,smokeSensorStatus,    isFireOrSmokeInImage)
        DisplayAlertOnShell("Wildfire Detected!")
        SaveImageForML(image)

    ELSE
        DisplayOnShell("Monitoring... No fire/smoke detected.")
    END IF

    // Step 6: Ask if system should continue monitoring
    continue ← CheckSystemStatus()
    IF continue == FALSE THEN
        BREAK
    END IF
  END LOOP
END
```

Figure 9. Pseudocode of the Detection Algorithm

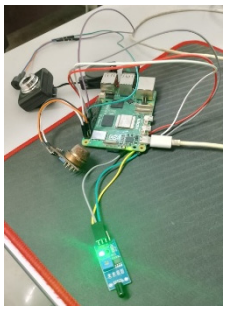
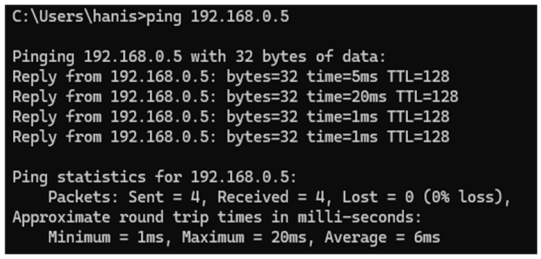
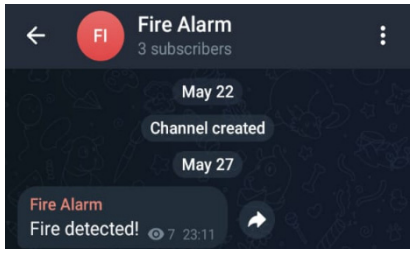
## 4. Results and Discussion

A comprehensive testing phase was conducted to assess the functionality, performance, and usability of the proposed wildfire detection system. Functional tests verified the integration and operation of hardware components, software modules, and communication interfaces. Performance testing was carried out under two distinct environmental scenarios, i.e indoor and outdoor settings, to evaluate the system's responsiveness, stability, and detection accuracy in varying conditions. Usability testing was performed using the System Usability Scale (SUS) (Harum et al., 2023), a standardized tool that measures user satisfaction, ease of use, and interface effectiveness. The results of all tests are presented in the following table, with each test successfully producing the intended outcomes and meeting the system's design requirements.

### *4.1 Functionality and Connectivity Tests*

Table 2 summarizes the functional and connectivity tests conducted during the evaluation phase. As part of the prototype-level assessment, successful connections were established between the Raspberry Pi and key components, including the flame and smoke sensors, buzzer, and webcam, as illustrated in Figure 10. Furthermore, network connectivity was validated through stable Wi-Fi access, enabling consistent communication with cloud services, as shown in Figure 11. In addition, notification tests confirmed that both the Telegram channel and the ThingSpeak dashboard received alerts as expected, as depicted in Figure 12. Finally, machine learning functionality was validated, with Figure 13 demonstrating that the YOLOv4 model operated correctly within the system.

Table 2. Functionality and Connectivity Tests

| Test Type | Description | Results |
|---|---|---|
| Prototype | To ensure reliable communication between the Raspberry Pi and the connected sensors and buzzer via the specified GPIO interfaces. |  Figure 10. Sensors and Buzzer Connectivity |
| Network Connectivity | To verify the Raspberry Pi's connection to the Wi-Fi network. |   Figure 11. Connectivity Test using Ping Command |
| Notification | To verify that alert messages generated by the Python script are successfully received by both the Telegram channel and the ThingSpeak dashboard. |   |

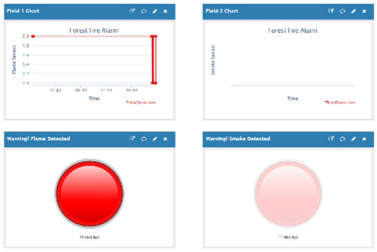


Figure 12. Notification on Telegram and Dashboard

| | |
|---|---|
| Machine Learning (CNN) | To ensure that the Webcam and YOLOv4 algorithm are functioning correctly with the Raspberry Pi. |




Figure 13. Image Detection using Webcam and YOLOv4

*4.2 Scenario-based Performance Test*

A scenario-based performance test was conducted in both indoor and outdoor environments. To evaluate the reliability and responsiveness of the fire detection system. The test aimed to measure the system's alert generation time based on inputs from two sensors, i.e flame and smoke, with a controlled interval distance of 10 cm between the fire source and the sensors. The primary metric assessed was the average response time for alert generation, which reflects the system's efficiency in detecting potential hazards and triggering timely notifications.

In the indoor environment, the performance test revealed that the flame sensor had an average response time of 16.94 seconds, while the smoke sensor responded in 22.84 seconds. In contrast, the outdoor environment showed a different pattern: the flame sensor recorded a slower average response time of 28.9 seconds, whereas the smoke sensor responded slightly faster at 21.276 seconds. The results indicate that the system functions as intended, with successful alert generation across both environments. However, notable differences in response times were observed:

Indoor vs Outdoor Flame Detection: The flame sensor performed significantly faster indoors (16.94s) compared to outdoors (28.9s). This discrepancy may be attributed to environmental factors such as wind, light interference, or temperature variations that affect flame visibility and sensor sensitivity outdoors.

Smoke Sensor Consistency: Interestingly, the smoke sensor showed relatively consistent performance across both settings, with only a slight improvement outdoors (21.276s vs 22.84s). This suggests that smoke dispersion patterns may be less affected by environmental conditions than flame detection, or that the sensor is more robust to such variations.

Sensor Sensitivity and Environmental Influence: The flame sensor appears more sensitive to environmental

changes, which could impact its reliability in open or ventilated areas. Conversely, the smoke sensor demonstrates more stable behavior, making it a critical component for outdoor deployment.

Overall, the system demonstrates effective performance in both settings, but the results highlight the importance of environmental calibration and sensor placement to optimize detection accuracy and response time. Future enhancements could include adaptive thresholding or sensor fusion techniques to further improve reliability under varying conditions.

*4.3 Usability Test*

The usability evaluation of the developed system was conducted using the System Usability Scale (SUS) (Harum et al., 2023), which yielded high scores, indicating that the interface is intuitive, efficient, and well-suited for real-time monitoring and alert management. Feedback was collected from eight (8) respondents, who assessed the system based on a standardized questionnaire consisting of 10 items. Each item was rated using a 5-point Likert Scale, where R represents the respondent and Qx denotes the question number (x = 1 to 10). The SUS scoring method involves adjusting responses as follows: for odd-numbered questions, 1 is subtracted from the score; for even-numbered questions, the score is subtracted from 5. The adjusted scores are then summed and multiplied by 2.5 to obtain the final SUS score. As illustrated in Figure 14, the results reflect a positive user experience, confirming that the system is user-friendly and effectively supports its intended functionality.

Table 3 summarizes the analysis of SUS data collected from the respondents, which meets the recommended sample size for SUS evaluations. In the table, each respondent is denoted by ‘R’, each question by ‘Q’, the raw score by ‘RS’, and the final calculated score by ‘FS’. The scoring process follows the standard SUS methodology: for odd-numbered questions, 1 is subtracted from the score, while for even-numbered questions, the score is subtracted from 5. The adjusted scores are then summed and multiplied by 2.5 to obtain the final SUS score. Based on this analysis, the developed system achieved an overall SUS score of 85.63%, corresponding to a usability grade of A, indicating excellent user satisfaction and interface quality.

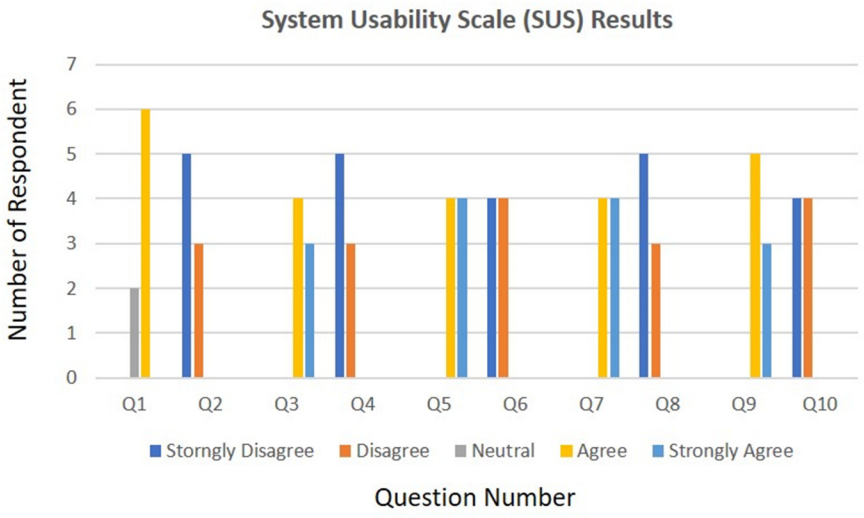



Figure 14. System Usability Scale (SUS) Test Results for Developed System

Table 3. Data Collection from SUS Test

| R | Q1 | Q2 | Q3 | Q4 | Q5 | Q6 | Q7 | Q8 | Q9 | Q10 | RS (%) | FS (%) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 4 | 1 | 4 | 2 | 4 | 2 | 4 | 2 | 4 | 2 | 29 | 77.5 |
| 2 | 4 | 1 | 5 | 2 | 4 | 1 | 5 | 1 | 4 | 2 | 27 | 87.5 |
| 3 | 3 | 1 | 4 | 1 | 5 | 1 | 5 | 1 | 4 | 1 | 26 | 90 |
| 4 | 3 | 1 | 4 | 1 | 5 | 1 | 5 | 1 | 5 | 2 | 28 | 90 |
| 5 | 4 | 2 | 4 | 1 | 5 | 2 | 5 | 2 | 4 | 1 | 30 | 85 |
| 6 | 4 | 2 | 5 | 1 | 4 | 1 | 4 | 1 | 5 | 1 | 26 | 90 |
| 7 | 4 | 2 | 4 | 2 | 4 | 2 | 4 | 2 | 4 | 2 | 30 | 75 |
| 8 | 4 | 1 | 5 | 1 | 5 | 2 | 4 | 1 | 5 | 1 | 29 | 90 |
| Average Score | | | | | | | | | | | | 85.63 |

## 5. Conclusion

Wildfire detection systems are essential for protecting lives and property, yet many existing solutions suffer from delayed response times, limited real-time communication, and poor adaptability to diverse environments. Conventional systems also tend to lack user-friendly interfaces and integration with modern IoT platforms, which limits their effectiveness in proactive hazard management. This study addresses these limitations by introducing an intelligent, sensor-based fire detection and alert system that combines real-time monitoring, automated alert generation, and a user-centric design approach.

The proposed system integrates flame and smoke sensors with a Raspberry Pi platform and leverages a pre-trained YOLOv4 convolutional neural network for visual fire detection. It enables real-time alert delivery via Telegram and provides data visualization through the ThingSpeak dashboard, ensuring timely and accessible hazard notifications. A comprehensive testing framework was employed to validate the system, including scenario-based performance testing in both indoor and outdoor environments, functionality testing of hardware and software components, and usability evaluation using the SUS method. The results confirmed the system's ease of use and suitability for end users.

This research contributes to the advancement of intelligent fire detection technologies by offering a practical and deployable solution that integrates deep learning with IoT capabilities, supports real-time communication, and emphasizes usability for non-technical users. The system's adaptability to low-resource environments further enhances its relevance for widespread application in residential, industrial, and public safety contexts. Future work may explore cloud-based analytics, mobile app integration, and adaptive algorithms to improve detection accuracy, scalability, and user engagement.

## 6. Acknowledgments

The authors extend sincere appreciation to the Universiti Teknikal Malaysia Melaka (UTeM) and the Fakulti Teknologi Maklumat dan Komunikasi (FTMK) for the valuable support provided, including financial assistance and access to essential resources, which facilitated the successful execution of this study. Gratitude is also expressed to all individuals and organizations whose contributions, collaboration, and encouragement significantly supported the completion of this research.

## References

Abdul Rahman, R., Daud, S., & Rajamoorthi, R. (2024). IoT-Based Smart Fire Detection System using ESP32, *Proceedings of the Liga Ilmu Serantau* (LiS 2024), 067-1077.

Ajith. S., Chandru. S., Boopalan. E., & Periyathambi. P. (202). Cloud Based Forest Fire Alert System using IoT, *International Journal of Trend in Scientific Research and Development (IJTSRD)*, ISSN: 2456-6470, Volume-4 | Issue-2, February 2020, 650-652.

Arun, K., R. P., Ram Kumar, Kappaganthu, Abhiram, & Samindla, Pruthvi. (2023). Forest Fire Detection with

GPS using IoT. *E3S Web of Conferences, 391*. https://doi.org/10.1051/e3sconf/202339101155

Bayegizova A., Abdikerimova G., Kaliyeva S., Shaikhanova A., Shangytbayeva G., Sugurova L., Sugur Z., & Saimanova Z. (2024). Fire Detection Using Deep Learning Methods, *International Journal of Electrical and Computer Engineering, 14*(1), 547-555. https://doi.org/10.11591/ijece.v14i1.pp547-555

Dasari, P., Reddy, G., & Gudipalli, A. (2020). Forest fire detection using wireless sensor networks. *International Journal on Smart Sensing and Intelligent Systems, 13*, 1-8. https://doi.org/10.21307/ijssis-2020-006

Deepika R., Srikanth M., Vignesh Adithva K., & Skanda Ganesh G. (2025). A Surveillance Camera Based Fire Detection and Localization, *2025 International Conference on Computing and Communication Technologies, ICCCT 2025*. https://doi.org/10.1109/ICCCT63501.2025.11019433

Ehsan, I., Mumtaz, Asia, Khalid, Muhammad, Iqbal, Jawaid, Hussain, Saddam, Ullah, Syed Sajid, & Umar, Fazlullah. (2022). Internet of Things Based Fire Alarm Navigation System: A Fire-Rescue Department Perspective. *Mobile Information Systems*. https://doi.org/10.1155/2022/3830372

Harum, N., Ahmad Mahin, N.A., Hamid, E., Emran, N.A., Anawar, S., & Asnawi, A.L. (2023). RFID Attendance System with Contagious Disease Prevention Module Using Internet-of-Things Technology. *International Journal of Interactive Mobile Technologies (iJIM), 17*(18), 4-15. https://doi.org/10.3991/ijim.v17i18.41603

Jyoshana, Y. S., Pooja, M. P., Arpitha, S., Manjunatha, C., & Farheen Saba, N. (2021). Automatic Fire Detection in Forest by Enabling IOT, *International Journal of Research Publication and Reviews, 2*(8), 527-534.

Krishnamoorthy, M., Md. Asif, Polamarasetty P. Kumar, Ramakrishna S. S. Nuvvula, & Baseem Khan. (2023). A Design and Development of the Smart Forest Alert Monitoring System Using IoT, *Journal of Sensors*. https://doi.org/10.1155/2023/8063524

Lakkar, P., & Narayan, M. (2020). Forest Fire Detection, Prevention and Response System using IoT, *International Research Journal of Engineering and Technology (IRJET), 7*(2), 2133-2136.

Lakshmanaswamy P., Sundaram A., & Sudanthiran T. (2024). Prioritizing the Right to Environment: Enhancing Forest Fire Detection and Prevention Through Satellite Data and Machine Learning Algorithms for Early Warning Systems, *Remote Sensing in Earth Systems Sciences, 7*(4), 472-485. https://doi.org/10.1007/s41976-024-00140-0

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

Mahalingam, S., Deep, M. S., & Krishna, K. S. (2021). Wireless Sensor Based Forest Fire Early Detection with Online Remote Monitoring, *International Journal of Engineering and Advanced Technology (IJEAT), 10*(5), 143-145. https://doi.org/10.35940/ijeat.E2670.0610521

Mo F., Deng L., Huang J., Zhang X., Zhang S., & Li H. (2023). Research on Remote Sensing Satellite Onboard Fire Detection and Real-Time Transmission Technology, *International Geoscience and Remote Sensing Symposium (IGARSS),* 4478-4481. https://doi.org/10.1109/IGARSS52108.2023.10281490

Mokhtar'udin, M.D., & Lias, J. (2024). IoT-Based Smart Fire Alarm Navigation System, *Evolution in Electrical And Electronic Engineering, 5*(2), 147-155.

Narvaez Luces O.D.R., Pham M.-T., Poterek Q., & Braun R. (2025). Burnt Area Extraction from High-Resolution Satellite Images Based on Anomaly Detection, *Communications in Computer and Information Science,* 2135 CCIS, 448-457. https://doi.org/10.1007/978-3-031-74633-8_33

Parida, D. (2021). *IoT Based Forest Fire Detection System using Arduino and GSM Module, Circuit Digest*, Retrieved from https://circuitdigest.com/microcontroller-projects/iot-based-forest-fire-detection-using-arduino-and-gsm-module

Pavitra, N., Khan, Sania, Jain, Seeksha, MN, Anusha, & Kalyan, Y. (2020). Forest Fire Detection System Using Iot. *International Journal of Engineering Applied Sciences and Technology, 5*, 331-336. https://doi.org/10.33564/IJEAST.2020.v05i04.050

Praveen, K. (2021). Forest Fire Detection System Using IOT. *International Journal of Innovative Research in Computer and Communication Engineering, 9*(4).

Pundir A.S., Buckchash H., Rajput A.S., Tanwar V.K., & Raman B. (2018). Fire Detection Using Dense

Trajectories, *Advances in Intelligent Systems and Computing, 704*, 211-221. https://doi.org/10.1007/978-981-10-7898-9_17

Sahal, R., Alsamhi, S. H., Breslin, J. G., & Ali, M. I. (2021). Industry 4.0 towards Forestry 4.0: Fire Detection Use Case. *Sensors, 21*(3), 694. https://doi.org/10.3390/s21030694

Saranya, M., Sahana, S., Suganthi, B., Vijaynigilesh, R., & Vivin, T. (2020). IOT Based Forest Fire Prediction and Detection. *International Journal of Innovative Technology and Exploring Engineering, 9*, 1835-1838. https://doi.org/10.35940/ijitee.F3914.049620

Shakhnoza M., Sabina U., Sevara M., & Cho Y.-I. (2022). Novel Video Surveillance-based Fire and Smoke Classification Using Attentional Feature Map in Capsule Networks, *Sensors, 22*(1), art. no. 98. https://doi.org/10.3390/s22010098

Sonti, S. M., Harika, A., Dileep, C., & Giri Raju, B. (2022). Fire Detecting And Alerting System Using Raspberry Pi Pico, *International Research Journal of Modernization in Engineering Technology and Science, 4*(6), 4565-4569.

Srinithi.P., & Dhivya.N. (2020). IoT Based Forest Fire and Deforestation Detection System, *International Journal of Innovative Research in Science, Engineering and Technology, 9*(3).

Wickramasinghe C., Wallace L., Reinke K., & Jones S. (2018). Implementation of a New Algorithm Resulting in Improvements in Accuracy and Resolution of SEVIRI Hotspot Products, *Remote Sensing Letters, 9*(9), 877-885. https://doi.org/10.1080/2150704X.2018.1484955

Wu W., Tang G., Liu C., Li D., Cui Q., Luo Y., Wan L., Wang X., Zhang T., Ding F., Li C., & Wang J. (2025). XingHuan Visible and Uncooled Multispectral Infrared Camera for Wildfire Detection: Algorithm Description and Initial Validation, *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 18*, 20149-20162. https://doi.org/10.1109/JSTARS.2025.3592843

Yu J., Jiang X., Zeng Z.-C., & Yung Y.L. (2024). Fire Monitoring and Detection Using Brightness-temperature Difference and Water Vapor Emission from the Atmospheric Infrared Sounder, *Journal of Quantitative Spectroscopy and Radiative Transfer, 317*, art. no. 108930. https://doi.org/10.1016/j.jqsrt.2024.108930

**Copyrights**

Copyright for this article is retained by the author(s), with first publication rights granted to the journal.

This is an open-access article distributed under the terms and conditions of the Creative Commons Attribution license (http://creativecommons.org/licenses/by/4.0/).