

Smart Fire Surveillance and Alarm Generation System Using Raspberry Pi and Yolo-Based Detection

Prof. Vikas Tupe, Karan Santosh Gade, Aditya Vinod Gaikwad, Avishkar Sandip Gondkar

Department Of Electronics and Telecommunication Engineering JSPM Narhe Technical Campus, Pune, Maharashtra, India

Abstract - Fire hazards remain a major threat to human life, industrial infrastructure, and residential safety due to delayed detection and limited response capabilities of conventional fire alarm systems. Traditional systems primarily rely on smoke or heat sensors, which may result in delayed alerts and reduced reliability. This paper presents a smart fire surveillance and alarm generation system based on Raspberry Pi 4 Model B that integrates computer vision, IoT, and multi-sensor monitoring for real-time fire detection. The proposed system utilizes YOLO (You Only Look Once) object detection and OpenCV image processing to analyze live video streams, while fire and smoke sensors provide additional verification. Upon detecting fire-related events, the system activates a local buzzer alarm and generates remote notifications through a network-based interface for rapid response. A web dashboard developed using Flask and React.js enables live monitoring, alert management, and remote accessibility. By combining vision-based detection with sensor feedback, the system enhances detection accuracy and minimizes false alarms. The proposed solution is low-cost, scalable, and energy-efficient, making it suitable for both industrial and residential environments. Experimental results indicate improved reliability, faster response time, and enhanced safety compared to conventional fire detection approaches.

Key Words: Fire Detection, Raspberry Pi, YOLO, Internet of Things (IoT), OpenCV, Smart Surveillance, Multi-Sensor Monitoring, Alarm Generation, Real-Time Monitoring, Computer Vision

1. INTRODUCTION

Fire accidents are among the most dangerous hazards affecting human life, residential buildings, commercial establishments, and industrial facilities. Rapid fire outbreaks can cause significant loss of life, property damage, and environmental impact if not detected and controlled at an early stage. Conventional fire detection systems primarily depend on smoke detectors, heat sensors, or flame sensors to identify fire-related incidents. Although these systems provide basic protection, they often experience delays in detection because smoke or heat must physically reach the sensors before an alarm is activated. Such delays can allow fires to spread rapidly, reducing the effectiveness of emergency response measures.

Recent advancements in embedded systems, Internet of Things (IoT), and Artificial Intelligence (AI) have enabled the development of intelligent fire surveillance systems capable of providing faster and more reliable

detection. Computer vision-based fire detection techniques can analyze live video streams to identify flames and smoke in real time, allowing earlier detection compared to traditional sensor-only approaches. Deep learning algorithms, particularly the YOLO (You Only Look Once) object detection model, have demonstrated high accuracy and speed in identifying objects within images and videos, making them suitable for real-time fire monitoring applications.

The Raspberry Pi 4 Model B offers a cost-effective and energy-efficient platform for implementing edge-based intelligent surveillance systems. With sufficient computational power and support for camera modules, sensors, and network communication, it can perform real-time image processing and monitoring tasks without requiring expensive hardware infrastructure. By integrating computer vision with IoT technologies, fire detection systems can provide continuous surveillance, automated alert generation, and remote accessibility.

This paper presents a Smart Fire Detection and Surveillance System based on Raspberry Pi 4 Model B. The proposed system combines YOLO-based fire and smoke detection with OpenCV image processing and sensor-based verification using fire and smoke sensors. The hybrid approach improves detection accuracy and minimizes false alarms by validating visual observations with physical sensor data. Upon detecting a fire event, the system activates a local buzzer alarm and generates remote notifications through a network interface.

To facilitate real-time monitoring, a web-based dashboard developed using Flask and React.js provides live video streaming, alert management, and remote access capabilities. The proposed system is designed to be scalable, affordable, and suitable for both industrial and residential environments. Experimental results demonstrate that the integration of computer vision and sensor feedback enhances detection reliability, reduces response time, and improves overall fire safety compared to conventional fire alarm systems.

2. Body of Paper

2.1 Existing Fire Detection Systems and Their Limitations

Traditional fire detection systems have served as the primary means of fire safety in residential, commercial, and industrial environments for several decades. These systems typically rely on photoelectric or ionisation-based smoke detectors, heat detectors, manual call

points, and fixed-temperature thermal sensors to identify fire-related events. While they remain the industry standard for basic fire protection, several well-documented shortcomings restrict their effectiveness in modern surveillance applications.^[1,2]

The fundamental limitation of sensor-only systems lies in their passive nature — they can detect fire only after sufficient smoke or heat has physically migrated to the sensor's location. In large open spaces such as warehouses, server rooms, and manufacturing floors, this delay can span several critical minutes.^[3] Furthermore, environmental conditions such as dust accumulation, steam from industrial processes, and rapid ambient temperature changes frequently trigger false alarms, reducing operational reliability and increasing emergency response costs.^[2]

The major limitations of conventional fire detection systems include:

- Delayed detection due to physical sensor proximity dependency
- Inability to visually verify or locate the fire source
- High false alarm rates caused by dust, steam, cooking fumes, and temperature fluctuations
- Limited spatial coverage in large-area environments
- No remote monitoring or real-time notification capabilities
- Absence of fire severity assessment prior to emergency dispatch
- Lack of integration with modern IoT or cloud-based management platforms
- Inability to differentiate between fire stages for proportional response

These shortcomings, particularly the combination of delayed response and high false alarm rates, underline the urgent need for intelligent, multi-modal fire surveillance systems that leverage computer vision and sensor fusion to deliver faster, more accurate, and remotely accessible fire detection.^[4]

2.2 Role of Computer Vision in Fire Detection

Computer vision has emerged as one of the most promising technologies for automated fire surveillance. Unlike traditional sensor-based approaches, vision-based

systems analyse continuous video streams to identify visual cues associated with fire and smoke — including colour patterns, texture characteristics, motion dynamics, and spatial spread — in real time.^[5]

Early vision-based fire detection approaches employed classical image processing techniques such as colour thresholding in HSV or YCbCr colour spaces, temporal frame differencing, and background subtraction to isolate fire pixels within a video frame.^[6] Although these methods demonstrated acceptable performance under controlled conditions, they suffered from poor robustness in dynamic lighting environments and exhibited elevated false positive rates in the presence of fire-coloured objects such as sunsets, orange warning lights, and heated metal surfaces.^[5]

OpenCV (Open Source Computer Vision Library) plays a central role in the image processing pipeline of the proposed system. It handles frame acquisition from the Zebtronics ZEB-Crystal Pro USB Webcam, RGB-to-BGR colour space conversion, spatial resolution normalisation, Gaussian noise reduction, and frame-by-frame buffering before model inference. These preprocessing steps are essential for improving the signal quality of input data fed to the deep learning model, directly impacting detection accuracy and speed.^[7]

The advantages of computer vision over conventional sensor-based fire detection include:

- Visual confirmation and spatial localisation of fire events
- Monitoring of large, complex environments from a single camera node
- Detection of early-stage flame and smoke before heat or particles reach sensors
- Ability to operate continuously without physical degradation over time
- Integration capability with existing CCTV and surveillance infrastructure

The integration of computer vision with physical sensor feedback in the proposed system creates a complementary detection architecture that reduces the weaknesses inherent to either approach used in isolation.^[8]

2.3 Deep Learning Based Fire Detection Using YOLOv8

Deep learning-based object detection has fundamentally transformed the capability of visual surveillance systems. Convolutional Neural Networks (CNNs) trained on large annotated datasets learn hierarchical feature representations that generalise far beyond the capability of manually engineered features, enabling accurate classification and localisation of complex visual patterns such as flame and smoke.^[9]

Among the available real-time object detection architectures, the YOLO (You Only Look Once) family of models is widely recognised for its combination of speed and accuracy, making it particularly suitable for embedded deployment.^[10] YOLOv8, the version adopted in the proposed system, introduces a decoupled detection head that separates classification and regression tasks, an anchor-free detection strategy for improved small-object localisation, and a C2f backbone for enhanced feature extraction with reduced computational overhead compared to its predecessors.^[11]

The YOLOv8n (nano) variant was selected for deployment on the Raspberry Pi 4, offering the optimal balance between inference speed and detection accuracy within the hardware's computational constraints. The model was fine-tuned using a custom dataset comprising approximately 5,000 annotated fire and smoke images sourced from publicly available fire detection datasets, including the FIRE dataset and the Furg-Fire-Dataset, augmented with synthetic images to improve class balance and robustness.^[12,13]

Key technical attributes of YOLOv8n relevant to this application are summarised in Table I.

Parameter	YOLOv5n (Baseline)	YOLOv8n (Proposed)
Model Size (MB)	3.9	3.2
mAP@0.5 (Fire)	78.4%	86.7%
Inference Speed (ms/frame)	310	248
Detection Head	Coupled	Decoupled
Anchor Strategy	Anchor-based	Anchor-free

Table I: Comparison of YOLOv5n and YOLOv8n on Fire Detection Task

The model was trained using transfer learning on YOLOv8n pretrained weights from the COCO dataset, followed by 100 epochs of fine-tuning on the fire-specific dataset with an input resolution of 416×416 pixels, achieving a mean Average Precision (mAP@0.5) of 86.7% on the test split. Confidence and IoU thresholds were set to 0.50 and 0.45 respectively to balance detection sensitivity and false alarm suppression.^[11]

2.4 System Architecture and Hardware Components

The proposed Smart Fire Surveillance and Alarm Generation System is built around a layered architecture comprising a sensing layer, a processing layer, a decision layer, and an alert/notification layer. This modular design ensures scalability, ease of maintenance, and adaptability to different deployment environments.^[14]

The hardware components are as follows:

Component	Specification	Function
Raspberry Pi 4B	4 GB RAM, Quad-core 1.8 GHz	Central processing unit
ESP8266 NodeMCU Module	80 MHz, Wi-Fi 802.11 b/g/n	Sensor data acquisition and wireless transmission
Zebronics ZEB-Crystal Pro USB Webcam	1080p @ 30 fps, USB 2.0	Video stream acquisition
MQ-2 Gas/Smoke Sensor	Analog & Digital output	Smoke and gas detection
Flame Sensor Module (IR)	760–1100 nm IR range	Physical flame detection
Active Piezo Buzzer	5 V DC, 85 dB	Local audible alarm
MicroSD Card	32 GB Class 10	OS and model storage

Table II: System Hardware Components and Specifications

The Raspberry Pi 4 Model B was selected as the central processing unit due to its 64-bit quad-core ARM Cortex-A72 processor and 4 GB LPDDR4 SDRAM, which provide sufficient computational headroom for concurrent YOLO inference, OpenCV frame processing, sensor polling, Flask server operation, and Wi-Fi communication for the web dashboard without requiring an external GPU. The ESP8266 NodeMCU module operates as a dedicated sensor node, collecting readings from the MQ-2 smoke sensor and IR flame sensor and transmitting the data to the Raspberry Pi over the local Wi-Fi network via HTTP GET requests, offloading GPIO interrupt handling from the main processing unit.^[15]

The software stack comprises: Raspberry Pi OS (64-bit Bullseye), Python 3.9, OpenCV 4.8, Ultralytics YOLOv8, Flask 3.0, React.js 18, and Arduino IDE

(ESP8266 firmware). The web dashboard is served locally by the Flask backend and accessible over the local network or through port-forwarded remote access.

2.5 Working Principle of the Proposed System

The proposed system employs a hybrid fire detection mechanism that combines real-time visual analysis with physical sensor verification, enabling multi-modal corroboration before triggering alerts. The detection pipeline operates as a continuous loop across four functional stages: acquisition, preprocessing, inference, and decision-action.^[16]

The four stages are described in detail below:

Stage 1 — Video Acquisition and Sensor Polling: The Zebronics USB webcam continuously captures live video at 1080p and 30 fps, accessed by OpenCV via the standard Video4Linux (V4L2) interface on Raspberry Pi OS. Simultaneously, the ESP8266 NodeMCU — connected to the MQ-2 smoke sensor and IR flame sensor — transmits sensor readings every 500 ms to the Raspberry Pi over Wi-Fi via HTTP GET requests to a lightweight endpoint on the Flask server. Sensor readings are normalised and stored in a shared state buffer accessible by the decision module.

Stage 2 — Frame Preprocessing (OpenCV): Each frame captured from the USB webcam is passed through an OpenCV preprocessing pipeline. Steps include: BGR to RGB conversion, bilinear resizing to 416×416 pixels, Gaussian blur (kernel 3×3) for noise reduction, and pixel normalisation to the [0, 1] range. Preprocessed frames are queued for model inference. Frame skipping is applied (every 3rd frame processed) to maintain real-time throughput on the constrained hardware.^[7]

Stage 3 — YOLOv8 Inference: The preprocessed frame is fed to the YOLOv8n model deployed using the Ultralytics Python SDK. The model performs a single forward pass and returns bounding box coordinates, class labels (fire / smoke), and confidence scores for all detected objects. Detections with confidence scores below 0.50 are discarded. Detected regions are highlighted with annotated bounding boxes rendered back on the live video stream for the web dashboard.^[11]

Stage 4 — Decision and Alert Generation: The system applies the following decision logic:

(i) If the YOLO model detects fire or smoke with confidence ≥ 0.50 , and the sensor module confirms elevated smoke level or IR flame signal, a HIGH CONFIDENCE alert is generated. The buzzer activates immediately and a push notification is transmitted via the Flask API to connected dashboard clients.

(ii) If only the vision model detects fire/smoke with confidence ≥ 0.65 but sensors do not confirm, a VISUAL ALERT is raised. The buzzer is activated after a 3-second delay, allowing the sensor to confirm or deny, reducing false positives from visually fire-like stimuli.

(iii) If only the sensors are triggered (e.g., smoke from cooking outside camera view), a SENSOR ALERT is raised without relying on visual confirmation, ensuring no single modality is a single point of failure.

This three-tier decision framework improves overall system reliability by ensuring that either a confirmed dual-modal detection or a high-confidence single-modal detection is required before irreversible alarm actions are triggered, directly reducing false alarm rates compared to single-sensor systems.^[8,16]

2.6 Experimental Results and Observations

The proposed system was evaluated in a controlled laboratory environment under three test conditions: open flame (candle, distance 0.5 m to 3 m), simulated smoke (non-toxic theatrical fog machine), and combined flame and smoke events. A total of 120 test trials were conducted across these conditions to assess detection accuracy, response time, and false alarm rate.^[17]

The key performance metrics recorded during testing are summarised in Table III.

Performance Metric	Conventional System	Proposed System
Detection Accuracy (%)	71.2%	89.4%
Avg. Detection Time (s)	14.6	2.3
False Alarm Rate (%)	18.3%	4.7%
Smoke Detection Range (m)	1.5	4.0
Remote Alert Capability	No	Yes (Web + Buzzer)
Power Consumption (W)	Varies	~8.5 (avg.)

Table III: Comparative Performance of Proposed System vs. Conventional System

The results demonstrate that the proposed system achieves a detection accuracy of 89.4% and an average detection time of 2.3 seconds, representing an 83.7% reduction in response time compared to the conventional sensor-only baseline. The false alarm rate was reduced from 18.3% to 4.7%, validating the effectiveness of the dual-modal verification approach. The system maintained stable performance at distances up to 4 metres from the fire source under typical indoor illumination conditions.^[17]

Limitations observed during testing include reduced visual detection accuracy under dense smoke conditions that obscure the camera field of view (accuracy dropped to 73.1% in high-density smoke trials), and a marginal increase in inference latency (~310 ms/frame) under peak CPU load when the Flask server is simultaneously handling multiple concurrent client connections. These limitations indicate areas for future improvement, including the integration of thermal imaging for smoke-obscured scenarios and the deployment of a lightweight edge inference optimiser such as TensorRT or ONNX Runtime.^[11,15]

2.7 Advantages of the Proposed System

The proposed Smart Fire Surveillance and Alarm Generation System offers a comprehensive set of advantages over both conventional single-sensor systems and prior computer vision-only approaches.^[14,16]

- Early-stage detection: Visual detection of small flames and incipient smoke before sensor thresholds are triggered
- Dual-modal verification: Sensor fusion reduces false alarm rate to 4.7%, a 74% improvement over conventional systems
- Remote monitoring: React.js web dashboard provides real-time live stream, detection logs, and alert history accessible over Wi-Fi
- Low-cost hardware platform: Total component cost under ₹8,000, significantly below commercial CCTV-integrated fire detection systems
- Energy efficient: Average power draw of ~8.5 W, enabling continuous 24/7 deployment on standard power or UPS
- Scalable architecture: Multiple camera-Pi nodes can be networked to a central Flask server for multi-zone coverage
- No cloud dependency: Entire inference pipeline runs locally on the Raspberry Pi, ensuring operation during internet outages
- Dual alert mechanism: Simultaneous local buzzer and remote web notification ensures alerts are not missed
- Extensible framework: YOLO model can be retrained for specific environments (e.g., chemical plants, server rooms) with custom datasets

2.8 Applications of the Proposed System

The modular, low-cost, and network-accessible design of the proposed system makes it suitable for deployment across a broad spectrum of environments where fire safety is a priority.^[14]

- Industrial plants and manufacturing facilities: High-risk zones with flammable materials benefit from early visual detection
- Warehouses and cold storage facilities: Large open areas where sensor proximity coverage is insufficient

- Residential buildings and apartments: Affordable, compact deployment for home fire safety
- Educational institutions: School and college campuses requiring non-disruptive, 24/7 monitoring
- Healthcare facilities and hospitals: Environments where false alarms cause critical operational disruption
- Data centres and server rooms: Where early smoke detection is critical for equipment protection
- Shopping malls and commercial complexes: Multi-zone deployments with centralised dashboard monitoring
- Smart city surveillance infrastructure: Integration with existing urban CCTV networks for city-wide fire detection
- Agricultural storage units: Remote monitoring of grain stores and crop drying facilities prone to self-ignition

The system's open-source software stack and standard hardware interfaces also facilitate adoption in research and academic environments, enabling further development of fire detection algorithms and multi-sensor fusion strategies.

3. CONCLUSIONS

The proposed Smart Fire Surveillance and Alarm Generation System successfully demonstrates a reliable, low-cost, and intelligent approach to real-time fire detection by integrating YOLOv8-based computer vision, OpenCV image processing, and ESP8266-based sensor communication on a Raspberry Pi 4 platform. The hybrid detection mechanism, combining visual analysis from a USB webcam with physical verification from MQ-2 smoke and IR flame sensors over Wi-Fi, effectively addresses the core limitations of conventional fire alarm systems, including delayed detection, high false alarm rates, and the absence of remote monitoring capabilities.

Experimental results confirm that the system achieves a detection accuracy of 89.4%, an average response time of 2.3 seconds, and a false alarm rate of 4.7%, representing significant improvements over traditional sensor-only approaches. The three-tier decision logic ensures proportional and reliable alert generation, while the Flask and React.js-based web dashboard enables live surveillance and remote accessibility over a local network. The distributed architecture, with the ESP8266 handling sensor acquisition independently of the Raspberry Pi's main processing pipeline, further improves system responsiveness and scalability. Future work will focus on integrating thermal imaging to improve detection accuracy under dense smoke conditions, deploying ONNX Runtime for optimised model inference on the Raspberry Pi, and extending the system to support multi-

node fire zone monitoring with a centralised alert management interface.

ACKNOWLEDGEMENT

The authors sincerely thank the project guide and faculty members of the Department of Electronics and Telecommunication Engineering, JSPM Narhe Technical Campus, Pune, for their continuous guidance, constructive feedback, and encouragement throughout the course of this work. Their technical insights significantly contributed to shaping the design and implementation of the proposed system. The authors also extend their gratitude to the institution for providing the laboratory infrastructure and computational resources necessary for system development and testing. Finally, heartfelt thanks are due to family and friends for their constant motivation and moral support during the completion of this research.

REFERENCES

- [1] W. Zheng, Z. Li, Y. Liu, and C. Li, "A review of fire detection methods based on deep learning," *IEEE Access*, vol. 10, pp. 64760–64781, 2022.
(Supports: traditional fire detection limitations, photoelectric /ionisation detectors)
- [2] K. Muhammad, J. Ahmad, I. Mehmood, S. Rho, and S. W. Baik, "Convolutional neural networks based fire detection in surveillance videos," *IEEE Access*, vol. 6, pp. 18174–18183, 2018.
(Supports: false alarms, environmental triggers, emergency response costs)
- [3] B. U. Toreyin, Y. Dedeoglu, U. Gudukbay, and A. E. Cetin, "Computer vision based method for real-time fire and flame detection," *Pattern Recognition Letters*, vol. 27, no. 1, pp. 49–58, 2006.
- [4] K. Muhammad, J. Ahmad, and S. W. Baik, "Early fire detection using convolutional neural networks during surveillance for effective disaster management," *Neurocomputing*, vol. 288, pp. 30–42, 2018.
- [5] K. Muhammad, M. Sajjad, F. Mehmood, and S. W. Baik, "Efficient fire detection for uncertain surveillance environment," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 5, pp. 3113–3122, 2019.
- [6] T. Celik and H. Demirel, "Fire detection in video sequences using a generic colour model," *Fire Safety Journal*, vol. 44, no. 2, pp. 147–158, 2009.
- [7] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision with the OpenCV Library*, 1st ed. Sebastopol, CA: O'Reilly Media, 2008.
- [8] K. Muhammad, J. Ahmad, Z. Lv, P. Bellavista, P. Yang, and S. W. Baik, "Efficient deep CNN-based fire detection and localization in video surveillance applications," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 7, pp. 1419–1434, 2019.
- [9] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.
- [10] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, 2016, pp. 779–788.
- [11] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLOv8," GitHub, 2023. [Online]. Available: <https://github.com/ultralytics/ultralytics>
- [12] A. Jadon, M. Omama, A. Varshney, M. S. Ansari, and R. Sharma, "FireNet: A Specialized Lightweight Fire & Smoke Detection Model for Real-Time IoT Applications," *arXiv preprint arXiv:1905.11922*, 2019.
- [13] V. Hüttner, C. R. Steffens, and S. S. da Costa Botelho, "First response fire combat: Deep learning based visible fire detection," in *Proc. 2017 Latin American Robotics Symposium (LARS) and 2017 Brazilian Symposium on Robotics (SBR)*, IEEE, 2017, pp. 1–6, doi: 10.1109/SBR-LARS-R.2017.8215312.
- [14] Z. Zhang, L. Tan, and T. L. K. Robert, "An improved fire and smoke detection method based on YOLOv8n for smart factories," *Sensors*, vol. 24, no. 15, p. 4786, 2024, doi: 10.3390/s24154786.
- [15] N. N. N. Zaid and N. Sazali, "Investigation of IoT devices for fire alarm detection systems using Raspberry Pi," in *Proc. 6th Asia Pacific Conference on Manufacturing Systems and 4th International Manufacturing Engineering Conference (iMEC-APCOMS 2022)*, Lecture Notes in Mechanical Engineering, Springer, Singapore, 2023, pp. 85–93, doi: 10.1007/978-981-99-1245-2_12.
- [16] S. N. Saydirasulovich, M. Mukhiddinov, O. Djuraev, A. Abdusalomov, and Y.-I. Cho, "An improved wildfire smoke detection based on YOLOv8 and UAV images," *Sensors*, vol. 23, no. 20, p. 8374, 2023, doi: 10.3390/s23208374.
- [17] D. Zhang, S. Han, and C. Zhao, "Vision-based forest fire detection in aerial images for preventing disasters," *Energies*, vol. 11, no. 8, p. 2013, 2018.