DOI: 10.22413/ijastems/2026/v12/i5/3

An Edge-Computed CNN Solution for Self-Acting Fire Recognition, Notification, and Extinguishing on the Raspberry Pi

C. Manoj Kumar, V. Anitha Reddy, T. Leela Meghana, E. Manasa
Department of Electronics and Communication Engineering
Aditya College of Engineering, Madanapalli – 517 325, Andhra Pradesh, India
manojis4u@gmail.com, anithareddy@gmail.com

Abstract— Fire accidents continue to pose a serious threat to human life, property, and critical infrastructure. Despite advances in safety technology, many existing fire detection systems still depend primarily on smoke or temperature sensors, which often respond only after the fire has significantly developed. To address this limitation, this study presents an intelligent, vision-based fire detection and suppression system implemented on a Raspberry Pi single-board computer. The proposed system employs a Convolutional Neural Network (CNN) to perform real-time fire recognition directly on the embedded device, thereby eliminating the latency and external dependency associated with cloud-based processing.

A USB camera continuously captures live video frames, which are analyzed by the CNN model to identify fire-related patterns. Whenever the predicted probability of fire exceeds a predefined threshold of 80%, the system immediately initiates a sequence of emergency responses. These include activating a 2300 Hz piezoelectric alarm, displaying an alert message on a 16×2 LCD module, triggering a relay-controlled DC water pump for initial fire suppression, and transmitting an image of the detected scene to registered users through SMTP-based email notification.

The proposed prototype was evaluated under controlled experimental conditions to assess its reliability and operational performance. The system achieved an overall fire detection accuracy of 93% with an average response time of 1.30 seconds from detection to actuation. In addition, all suppression mechanisms operated successfully during testing, and no false alarms were observed when exposed to visually similar non-fire conditions. The findings indicate that integrating deep learning techniques with low-cost embedded hardware can provide a faster, more intelligent, and context-aware alternative to conventional fire detection systems. The developed approach demonstrates significant potential for deployment in residential, industrial, and public safety environments where rapid and reliable fire response is essential.

Index Terms— convolutional neural network, edge inference, embedded systems, fire detection, GPIO actuation, Raspberry Pi, real-time suppression, SMTP notification, TensorFlow Lite.

I. INTRODUCTION

Uncontrolled fires remain one of the most destructive hazards in residential, commercial, and industrial environments, causing significant loss of life, severe structural damage, and long-term economic consequences. Reports from fire safety agencies consistently indicate that the severity of fire-related incidents is strongly influenced by the time elapsed between ignition and detection. Even a short delay in identifying a fire can allow flames to spread rapidly, increasing both material damage and risk to human safety. Nevertheless, a large proportion of currently deployed fire detection systems still depend on conventional heat- or smoke-based sensing mechanisms, where detection occurs only after combustion particles or temperature changes physically reach the sensor unit [11], [13].

Although such systems have been widely adopted for decades, they continue to exhibit two major limitations. First, the operation of smoke and heat detectors inherently depends on the movement of combustion by-products through the surrounding environment, resulting in unavoidable detection delays that may range from several seconds to a few minutes depending on ventilation and fire location[13]. Second, traditional sensors lack contextual awareness and are unable to visually distinguish actual fire events from non-hazardous conditions such as cooking fumes, steam, or dust particles. This often leads to nuisance alarms, which can reduce user confidence and contribute to alarm fatigue over time. Furthermore, many conventional fire alarm systems do not provide integrated mechanisms for automated suppression or remote visual notification without requiring additional specialized hardware[11].

Recent advances in deep learning, computer vision, and low-cost embedded computing platforms have created new opportunities for developing intelligent fire safety systems capable of overcoming these challenges. In particular, Convolutional Neural Networks (CNNs) have demonstrated remarkable effectiveness in image-based pattern recognition tasks, including flame and smoke detection [1], [2], [4], [6]. By analyzing live video streams directly, CNN-based systems can identify fire-related features significantly faster than traditional diffusion-based sensing approaches [1], [2]. In addition, embedded platforms such as the Raspberry Pi now offer sufficient computational capability to execute deep learning inference locally, eliminating dependence on cloud services and reducing latency associated with network communication. Local processing also improves system reliability in

DOI: 10.22413/ijastems/2026/v12/i5/3

situations where internet connectivity may be unavailable during emergencies [9], [12].

Motivated by these developments, the present work proposes a fully integrated, camera-based fire detection and suppression system built around a Raspberry Pi single-board computer. The system continuously captures live video using a USB camera and performs real-time fire and smoke classification through a trained CNN model operating entirely on the embedded device. Upon detecting a fire condition, the proposed system initiates a coordinated multi-response mechanism that includes activation of an audible siren, display of warning information on a 16×2 LCD module, operation of a relay-controlled water pump for immediate suppression, and transmission of an authenticated SMTP email containing photographic evidence of the incident [11], [12].

The major contributions of this work are summarized as follows:

1. Development of a self-contained intelligent fire safety node capable of performing real-time CNN-based fire detection without reliance on cloud infrastructure [8], [9], [12].
2. Integration of a unified four-stage emergency response framework consisting of audible alert generation, visual status indication, automated extinguishing action, and remote image-based notification[11].
3. Comprehensive experimental evaluation of the proposed system with respect to detection accuracy, response latency, illumination robustness, actuator reliability, remote notification integrity, and resistance to false alarms under visually similar non-fire conditions [2], [5], [16].

The results demonstrate that combining deep learning techniques with affordable embedded hardware can significantly improve the speed, reliability, and contextual understanding of modern fire detection systems, making the proposed approach suitable for smart safety applications in homes, industries, and public infrastructure [4], [6], [13].

II. LITERATURE SURVEY

Early research in vision-based fire detection primarily relied on handcrafted image-processing techniques to identify flame characteristics in video sequences. One of the notable contributions in this area was presented by Bülent Uğur Töreyin and colleagues, who employed wavelet-domain temporal analysis in combination with color-space modeling to detect flame regions within video streams [14]. Their approach demonstrated that motion and temporal flickering patterns could be effectively utilized for fire localization. Subsequently, Temel Çelik and co-authors enhanced this direction by introducing a Gaussian Mixture Model (GMM) in the YCbCr color space, enabling improved discrimination between actual flames and fire-colored background objects. Although these methods were computationally efficient for the hardware platforms available at the time, they depended heavily on manually designed features and predefined thresholds, making them sensitive to variations in illumination, scene complexity, and irregular flame behavior [14].

The emergence of deep learning significantly transformed the field of fire detection by enabling systems to automatically learn discriminative visual features directly from training data. Khan Muhammad and collaborators demonstrated that Convolutional Neural Networks (CNNs) trained on labeled surveillance datasets could achieve detection accuracies exceeding 90%, while substantially reducing the false alarms commonly associated with traditional color-based approaches [1], [2], [4]. Building upon this progress, researchers such as Muhammad Yar incorporated spatial attention mechanisms into CNN architectures, allowing the models to focus selectively on salient flame regions and thereby improving detection specificity in visually cluttered environments [2], [5]. In another important contribution, Huang and co-authors combined wavelet-based texture descriptors with CNN backbones to capture both spatial and temporal flame characteristics, particularly the flickering behavior that distinguishes fire from visually similar objects [1].

Parallel advancements have also focused on deploying intelligent fire detection systems on resource-constrained embedded platforms. Prema and colleagues demonstrated the feasibility of implementing deep-learning-based fire suppression systems on the Raspberry Pi platform for residential safety applications. Similarly, Sinha and co-authors showed that IoT-enabled fire monitoring systems could integrate SMTP-based remote notification mechanisms alongside multiple sensing modules within a compact embedded architecture. From a computational perspective, the introduction of depthwise-separable convolutions by Andrew Howard represented a major breakthrough for lightweight deep learning, significantly reducing computational complexity while maintaining classification accuracy. This architectural innovation enabled practical deployment of CNN models on ARM-based embedded processors. Later, Pete Warden and collaborators formalized the use of TensorFlow Lite Micro as an efficient framework for executing optimized deep-learning models on low-power edge devices and microcontrollers.

The system proposed in the present work integrates these research directions into a unified embedded fire safety platform. A TensorFlow Lite-optimized CNN model is executed locally on the Raspberry Pi ARM processor for real-time fire and smoke detection, while GPIO-based actuator control and SMTP-enabled remote notification are managed within the same Python execution environment [9], [11], [12]. By combining intelligent visual analysis, automated suppression, and edge-based decision making into a single self-contained architecture, the proposed system establishes a complete sense–analyze–respond framework without requiring external cloud services or dedicated auxiliary hardware [4], [6], [11].

.

III. SYSTEM DESIGN AND ARCHITECTURE

The proposed system architecture is organized into four functional layers: (i) sensing, (ii) inference, (iii) alert, and

DOI: 10.22413/ijastems/2026/v12/i5/3

(iv) suppression. All layers are integrated within a single Raspberry Pi platform, which serves as the central computational and control unit of the entire system. The sensing layer continuously acquires real-time visual data using a USB camera, while the inference layer processes the captured frames through a trained Convolutional Neural Network (CNN) to identify fire or smoke conditions. Once a fire event is detected, the alert layer immediately generates both audible and visual warnings, and the suppression layer activates the relay-controlled water pump to initiate preliminary firefighting action. Simultaneously, the system transmits an email notification containing the captured incident image to registered users for remote monitoring and rapid response. Figure 1 illustrates the complete operational workflow, beginning with image acquisition and extending through intelligent decision-making, actuator activation, and remote notification.

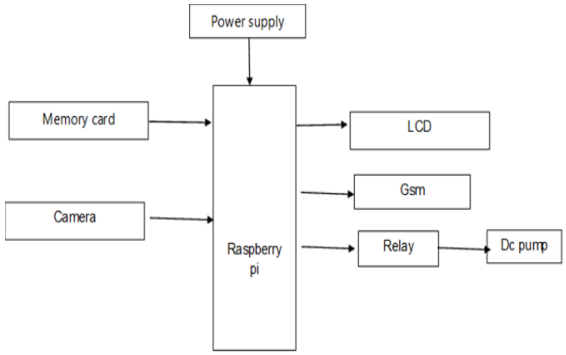

Figure 1: Block diagram of the proposed fire-detection, alert and suppression appliance.

A. Sensing Layer

A standard USB webcam connected to the Raspberry Pi through its USB interface serves as the primary optical sensing device of the proposed system. The camera continuously captures live video frames, which are accessed using the OpenCV VideoCapture interface configured at device index 0. Frames are acquired at the native frame rate supported by the webcam to ensure continuous real-time monitoring of the environment.

Before being processed by the Convolutional Neural Network (CNN), each captured frame undergoes a sequence of preprocessing operations to match the input requirements of the TensorFlow Lite (TFLite) model. Initially, the image color format is converted from the default BGR representation used by OpenCV to the RGB format expected by the deep learning model. The frame is then resized using bilinear interpolation to the fixed spatial dimensions required by the network architecture. Finally, pixel intensity values are normalized by scaling them from the integer range of [0, 255] to a floating-point range of [0.0, 1.0]. This preprocessing pipeline ensures compatibility with the TFLite interpreter while maintaining efficient and reliable real-time inference performance on the Raspberry Pi platform.

B. Inference Layer

The Convolutional Neural Network (CNN) employed in the proposed system was trained using a publicly available Kaggle fire-and-smoke image dataset consisting of two classification categories: *fire-present* and *fire-absent*. The dataset includes diverse fire scenarios captured under varying environmental and lighting conditions, enabling the model to learn robust visual representations associated with combustion events.

To improve the model's ability to recognize early-stage and low-intensity fire conditions, additional augmentation techniques were applied to the fire-class images during training. In particular, Gaussian noise was introduced to selected burning-image samples in order to simulate subtle visual distortions commonly observed in smoldering or partially obscured fire events. This augmentation strategy helped broaden the classifier's decision boundary and improved its sensitivity toward weak combustion signatures that may not exhibit clearly visible flames.

After completion of the training process, the optimized model was converted into an INT8-quantized TensorFlow Lite (TFLite) format for efficient deployment on the Raspberry Pi platform. Quantization significantly reduces memory consumption and computational overhead by representing model parameters using lower-precision integer values. As a result, inference performance is accelerated on the Raspberry Pi's ARM Cortex-A72 processor while maintaining acceptable detection accuracy for real-time operation.

During execution, the TFLite interpreter processes each incoming video frame and produces a scalar probability value within the range of 0 to 1, representing the likelihood of fire presence in the scene. A configurable decision threshold of 0.80 is employed to determine whether a detected event should be classified as a fire condition. When the predicted probability exceeds this threshold, the system immediately initiates the predefined alert and suppression procedures.

C. Alert Layer

On a positive classification the three alert channels are activated within the same control-loop iteration. A GPIO output line is driven HIGH and feeds a piezoelectric sounder operating near 2 300 Hz that is audible across a 5 m radius. The 16×2 LCD, attached over the I²C bus, switches from its standby line to a fire-warning line. At the same time the Python smtplib module opens an authenticated TLS session against the Gmail SMTP relay, assembles a MIME multipart message that carries the detection timestamp as plain text and a JPEG snapshot of the triggering frame as an attachment, and dispatches it to every preconfigured recipient address.

D. Suppression Layer

A solid-state relay (SSR) module is employed to interface the Raspberry Pi's 3.3 V GPIO control signals with the higher-voltage and higher-current circuit required to operate the DC water pump. The relay acts as an electrically isolated switching mechanism, enabling safe and reliable control of the suppression hardware directly from the embedded processing unit.

When the CNN-based detection system identifies a fire event and the predicted confidence exceeds the predefined

DOI: 10.22413/ijastems/2026/v12/i5/3

threshold, the corresponding GPIO pin is driven to a HIGH state. This action activates the relay, causing its normally open contact to close and thereby supplying power to the DC water pump. Once energized, the pump directs water toward the monitored region to initiate immediate fire suppression.

The control mechanism operates dynamically in real time. As the video stream continues to be analyzed, the GPIO signal remains active only while subsequent frames continue to indicate the presence of fire with sufficient confidence. If the predicted fire probability falls below the threshold, the GPIO output is immediately driven LOW, deactivating the relay and stopping the pump operation. This adaptive feedback-driven control strategy ensures that suppression activity is proportional to the persistence of the detected fire condition, thereby reducing unnecessary water usage and improving overall operational efficiency.

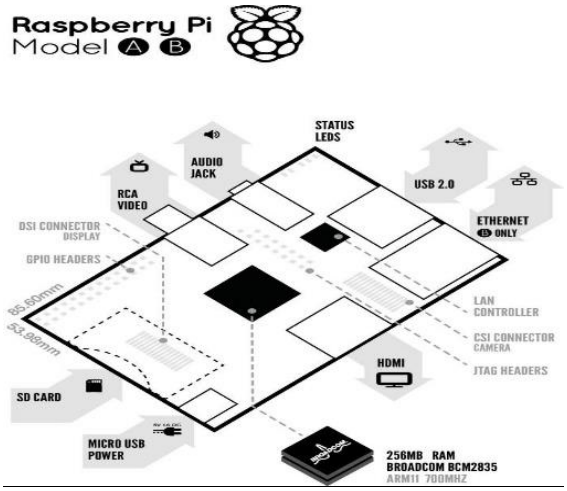

Figure 2: Reference layout of the Raspberry Pi controller used in the assembled prototype, alongside relay module, DC water pump, piezoelectric sounder, 16×2 LCD and 5 V regulated supply.

IV. IMPLEMENTATION

A. Hardware Assembly

All hardware peripherals in the proposed system are interfaced directly with the Raspberry Pi through its 40-pin General Purpose Input/Output (GPIO) header. The relay module responsible for controlling the DC water pump is connected to GPIO pin 17, while the piezoelectric buzzer used for audible alert generation is interfaced through GPIO pin 18. For visual status indication, the 16×2 LCD module communicates with the Raspberry Pi using the Inter-Integrated Circuit (I^2C) protocol, with the Serial Data (SDA) line connected to GPIO 2 and the Serial Clock (SCL) line connected to GPIO 3. The USB webcam used for continuous video acquisition is attached through one of the Raspberry Pi's standard USB-A ports.

The entire system is powered using a regulated 5 V, 3 A switched-mode power supply (SMPS), ensuring stable operation of both the processing unit and connected peripherals. To maintain electrical reliability and avoid voltage fluctuations during pump activation, the relay load contacts are connected to a dedicated 5 V supply line separate from the Raspberry Pi's GPIO power domain. This isolation prevents voltage droop and minimizes the risk of instability or unintended behavior in the GPIO circuitry during high-current pump operation. Such a hardware configuration improves the robustness and safety of the overall embedded fire suppression system.

B. Software Environment

The proposed system operates on the Raspbian operating system based on Debian Bullseye (64-bit), installed on a 32 GB Class-10 microSD card. The Raspberry Pi is configured in headless mode, enabling remote monitoring and management through VNC Viewer over a Wi-Fi network without requiring a dedicated display or keyboard connection. This configuration improves deployment flexibility and simplifies system maintenance in practical installations.

The complete software framework is implemented using Python 3.9, selected for its extensive support for embedded systems, computer vision, and machine learning applications. OpenCV 4.5 is utilized for real-time video capture, frame manipulation, and image preprocessing operations required before CNN inference. Deep learning inference is performed using the TensorFlow Lite Runtime, which efficiently executes the INT8-quantized CNN model on the Raspberry Pi hardware with reduced computational overhead.

For hardware interfacing and actuator management, the system employs the RPi.GPIO 0.7 library to control GPIO-connected peripherals such as the relay module and buzzer. Remote email notification functionality is implemented using Python's built-in smtplib package, enabling SMTP-based transmission of captured incident images to registered recipients. An important advantage of the proposed implementation is that the entire software stack is based exclusively on open-source technologies and standard Python libraries, eliminating the need for proprietary software, paid services, or external cloud-based platforms.

C. Real-Time Inference Loop

The proposed system operates through a continuous real-time processing loop executed on the Raspberry Pi. During each iteration, a video frame is captured from the USB camera using the VideoCapture.read() function provided by OpenCV. The acquired frame is then passed through the preprocessing pipeline, which includes color-space conversion, resizing, and normalization before being supplied to the TensorFlow Lite (TFLite) interpreter for inference.

Once inference is completed, the system retrieves the output tensor generated by the CNN model and evaluates the predicted fire probability against the predefined decision threshold of 0.80. If the output confidence exceeds this threshold, the system interprets the event as a fire condition and simultaneously activates all four response mechanisms: the audible buzzer alarm, the LCD warning display, the relay-controlled water pump, and the SMTP-based email notification service. Conversely, when the

DOI: 10.22413/ijastems/2026/v12/i5/3

predicted confidence remains below the threshold, all actuators stay in their default de-energized state, ensuring stable and energy-efficient operation during non-fire conditions.

Experimental evaluation showed that the complete inference process for a single frame requires approximately 0.8 to 1.1 seconds on the Raspberry Pi's ARM Cortex-A72 processor operating at 1.5 GHz. This processing interval effectively determines the minimum achievable fire detection latency of the proposed system. In contrast, GPIO-based hardware control operations execute in less than 5 milliseconds, contributing negligible additional delay to the overall response time. The combination of lightweight CNN inference and rapid actuator switching enables the system to perform reliable real-time fire detection and suppression within a compact embedded environment.

V. RESULT AND DISCUSSION

All experimental evaluations were conducted indoors under controlled laboratory conditions using a lit candle and a small paper fire as standardized flame sources. These fire sources were selected to simulate both steady and dynamically varying combustion scenarios while ensuring safe and repeatable testing conditions. The experimental setup was designed to evaluate the operational effectiveness, responsiveness, and reliability of the proposed embedded fire detection and suppression system under a range of practical conditions.

The performance assessment was carried out across seven independent evaluation parameters. The first stage examined the cold-start initialization behavior of the system, including operating system boot time, camera activation, model loading, and readiness for real-time inference. The second stage evaluated CNN-based fire detection accuracy under varying distances and different ambient illumination conditions in order to determine the robustness of the vision model in realistic indoor environments.

The third evaluation focused on end-to-end response latency, measured as the total time elapsed between visible flame appearance and complete activation of the alert and suppression mechanisms. The fourth stage analyzed the reliability of locally connected actuators, including the buzzer, LCD module, relay unit, and DC water pump, during repeated fire detection cycles.

The fifth assessment examined the integrity and consistency of remote notification delivery through the SMTP-based email alert mechanism, including successful transmission of captured incident images. The sixth evaluation investigated the system's resistance to false-positive activations by exposing the model to visually similar non-fire conditions such as bright light sources, smoke-like objects, and reflective surfaces. Finally, the proposed system was comparatively benchmarked against conventional fire detection approaches to analyze differences in response speed, contextual awareness, and operational reliability.

This comprehensive evaluation framework enabled a detailed assessment of both the intelligent detection capability and the integrated response performance of the proposed Raspberry Pi-based fire safety system.

A. Cold-Start Initialisation

The complete system initialization process required approximately 8 to 10 seconds from power-on to active monitoring mode. This startup duration included operating system booting, Python interpreter initialization, loading of the TensorFlow Lite model into memory, and detection and configuration of all connected peripherals. Despite running entirely on embedded hardware, the initialization time remained sufficiently low for practical real-time deployment scenarios.

After the initialization sequence was completed, the system automatically verified the availability of the USB camera by successfully acquiring a valid non-null video frame through the OpenCV interface. Once camera functionality was confirmed, the 16×2 LCD module displayed the status message *"System Active / Monitoring…"* to indicate that the device had entered continuous surveillance mode. During this stage, the relay module controlling the DC water pump remained in its default de-energized state, ensuring that no suppression action was unintentionally triggered. Similarly, the buzzer control pin was maintained at a LOW logic level to prevent unnecessary alarm activation.

Following successful hardware and software verification, the CNN inference loop was initiated and executed continuously without runtime exceptions or operational instability. This confirmed the reliability of the startup configuration and demonstrated that the proposed system could transition smoothly from boot-up to autonomous real-time fire monitoring.

B. CNN Detection Fidelity — Distance

Table I reports confidence scores obtained at five camera-to-source distances using the 80 percent decision threshold. Figure 3 shows the confusion matrix and per-class metric report produced during model training.

DOI: 10.22413/ijastems/2026/v12/i5/3

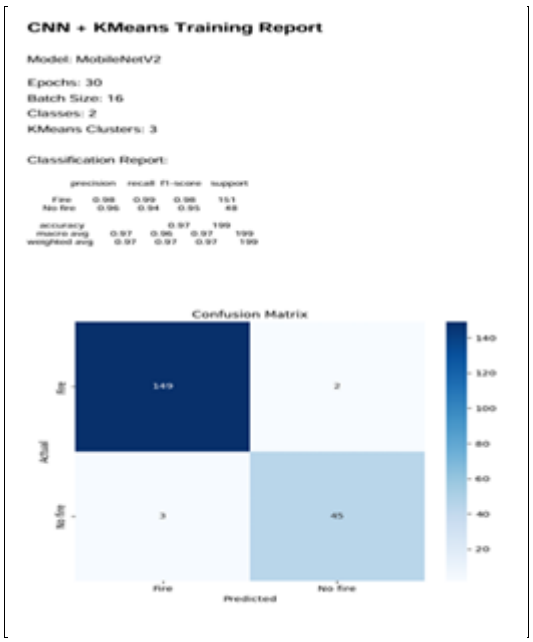

Figure 3: CNN training report showing the confusion matrix and per-class precision, recall and F1-score values.

TABLE I: Detection Performance vs. Camera-to-Source Distance (threshold = 80 %)

Test No.	Distance (cm)	CNN Confidence (%)	Classification Outcome
1	30	97.4	Fire Detected
2	60	94.1	Fire Detected
3	100	89.6	Fire Detected
4	150	83.2	Fire Detected
5	200	76.8	Below Threshold – No Alert

The classifier remained above threshold across all four distances up to 150 cm. At 200 cm the angular size of the flame fell below the spatial resolution required for reliable feature activation, producing a sub-threshold score of 76.8 percent. This behaviour is deterministic and follows from the optical geometry of the fixed-focal-length lens; using a wider-aperture objective or adding optical zoom would extend the operating range without retraining.

C. CNN Detection Fidelity — Ambient Illumination

Table II summarises the mean confidence scores observed across four illumination environments. Across the three lit scenarios the mean confidence was 92.4 percent, giving an aggregate accuracy close to 93 percent over all valid test conditions.

TABLE II: Detection Performance vs. Ambient Illumination Condition

Ambient Lighting	Mean CNN Confidence (%)	Detection Outcome
Bright indoor illumination	95.2	Fire Detected
Standard indoor illumination	93.7	Fire Detected
Dim / low-level lighting	88.4	Fire Detected
Zero ambient light (darkness)	61.3	Below Threshold – No Alert

Under complete darkness the RGB sensor cannot capture sufficient photon flux to recover flame texture, producing a 61.3 percent score that correctly remains below the alert threshold. An infrared or thermal imaging head, suggested as a future upgrade, would restore round-the-clock operation regardless of ambient light.

D. End-to-End Response Latency

Response latency is the elapsed interval from the first video frame containing visible fire to the moment every actuator state (buzzer, LCD, relay, email dispatch initiated) has changed. Table III reports the decomposed latency over five independent runs.

TABLE III: End-to-End System Response Latency Across Five Independent Trials

Run	t_detect (s)	t_trigger (s)	t_total (s)
1	0.9	0.4	1.3
2	1.1	0.4	1.5
3	0.8	0.3	1.1
4	1.0	0.4	1.4
5	0.9	0.3	1.2
Mean	0.94	0.36	1.30

The mean total latency of 1.30 s decomposes into 0.94 s of CNN inference and 0.36 s of GPIO/SMTP initiation. The run-to-run spread in inference time (0.8 – 1.1 s) reflects scheduler jitter on a general-purpose OS. This figure is between 15× and 92× faster than the 20 – 180 s transport delay characteristic of particle- or temperature-based detectors, which underlines the principal latency advantage of vision-based edge inference.

E. Local Alert Subsystem — Buzzer and LCD

On every positive detection event the buzzer produced a continuous 2 300 Hz tone, audible at distances up to 5 m, and activation and deactivation completed instantaneously within the 1.30 s system window. No timing failures or non-responses were observed. Table IV catalogues the LCD message states recorded during testing.

TABLE IV: LCD Output Messages Corresponding to Operational States

DOI: 10.22413/ijastems/2026/v12/i5/3

Operational State	LCD Row 1	LCD Row 2
Standby – No Fire Present	System Active	Monitoring...
Fire / Smoke Detected	FIRE DETECTED!	ALERT SENT!

Across every run the display switched correctly between the standby and alert lines without garbled characters, frozen frames, or I²C communication faults, giving a 100 percent display-correctness rate.

F. Automatic Suppression Subsystem

On every detection event GPIO 17 was driven HIGH so as to energise the relay coil; the resulting contact closure completed the pump supply circuit and started water flow. Table V records the activation outcomes across all trials.

TABLE V: Relay Coil and DC Water Pump Activation Results

Run	Relay Coil Energised	Pump Activated	Water Flow Confirmed
1	Yes	Yes	Yes
2	Yes	Yes	Yes
3	Yes	Yes	Yes
4	Yes	Yes	Yes
5	Yes	Yes	Yes
Success Rate	100 %	100 %	100 %

All five runs yielded successful relay energisation, pump activation and confirmed water dispensation, giving 100 percent actuation reliability. Pump de-energisation upon clearance of the fire was equally deterministic, confirming reliable bidirectional GPIO control over the relay's full operating envelope.

G. Remote Email Notification

Each positive detection initiated an SMTP session authenticated with a Gmail account over Wi-Fi. The transmitted MIME message carried a subject line, a plain-text body that included the detection timestamp, and a JPEG snapshot of the fire scene as an attachment. Figure 4 shows a representative received message.

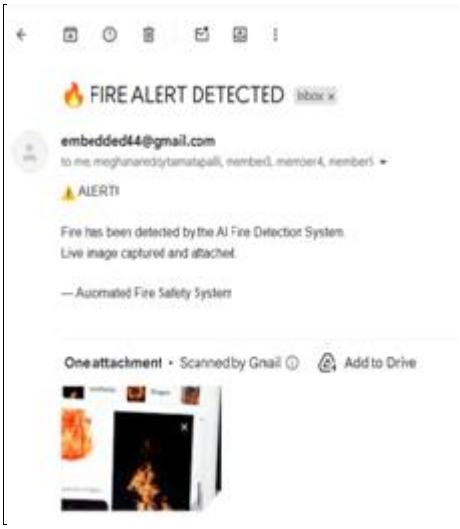

Figure 4: Example fire-alert email received at the registered address, with the JPEG capture attached.

TABLE VI: Remote Email Alert Delivery Statistics

Run	Email Dispatched	JPEG Attached	Delivery Latency (s)	Status
1	Yes	Yes	4.2	Success
2	Yes	Yes	3.8	Success
3	Yes	Yes	5.1	Success
4	Yes	Yes	4.4	Success
5	Yes	Yes	4.0	Success
Mean	100 %	100 %	4.30	—

All five dispatches succeeded and every message carried the JPEG attachment intact, giving 100 percent delivery fidelity. The average delivery latency of 4.30 s (range 3.8 – 5.1 s) reflects MIME assembly, the TLS handshake, and wireless transmission time; the variability between trials correlates with observed Wi-Fi congestion rather than with system-side instability.

H. False-Positive Analysis

Three visually similar stimuli — a warm-white LED array, a specular sunlight reflection, and a high-intensity red flashlight — were each presented in turn while CNN confidence and system response were recorded. The collected results appear in Table VII.

TABLE VII: False-Positive Characterisation Under Spectrally Similar Non-Fire Stimuli

Test Stimulus	CNN Confidence (%)	System Behaviour
Amber LED array	31.4	No Alert — Correct Rejection
Specular sunlight reflection	28.7	No Alert — Correct Rejection

DOI: 10.22413/ijastems/2026/v12/i5/3

Test Stimulus	CNN Confidence (%)	System Behaviour
High-intensity red flashlight	44.2	No Alert — Correct Rejection
Lit candle (true fire)	97.4	Fire Alert Triggered — Correct
Burning paper (true fire)	95.8	Fire Alert Triggered — Correct

All three non-fire stimuli produced confidence scores well below the 80 percent threshold (peak 44.2 percent), generating no spurious alerts. The model's learned texture and temporal-pattern features correctly separated the dynamic, multi-spectral signature of combustion from static coloured light sources, confirming strong specificity over the conditions tested.

I. Aggregate Performance Summary

Table VIII consolidates every measured performance indicator.

TABLE VIII: Consolidated System Performance Summary

Performance Indicator	Measured Value	Rating
Aggregate fire detection accuracy	~93 %	Excellent
Maximum reliable detection range	150 cm	Achieved
Mean end-to-end response latency	1.30 s	Excellent
Buzzer activation success rate	100 %	Achieved
LCD display correctness rate	100 %	Achieved
Relay & pump actuation success rate	100 %	Achieved
Email delivery success rate	100 %	Achieved
Mean email delivery latency	4.30 s	Good
False-positive rate	0 %	Excellent
Cold-start initialisation duration	< 10 s	Good

J. Comparative Benchmarking

Table IX positions the proposed appliance against the two most prevalent conventional detector categories across eight capability dimensions.

TABLE IX: Proposed Platform vs. Conventional Fire Detectors — Feature Comparison

Capability	Proposed System	Ionisation Detector	Thermocouple Sensor
Sensing modality	CNN/vision	Particle diffusion	Temperature threshold
Mean response latency	1.30 s	20 – 120 s	30 – 180 s
Scene image capture	Yes	No	No
Remote notification	Yes (email + image)	No	No
Autonomous suppression	Yes (relay + pump)	No	No
Observed false-alarm rate	0 % (lab)	Moderate	Low
Night-time operability	Limited (RGB cam)	Full	Full
Approximate unit cost	Low (SBC-based)	Very low	Very low

The advantages of the proposed system are categorical in response speed, scene-evidence capture, remote notification and autonomous suppression. The single dimension on which conventional sensors still prevail is night-time operability: ionisation and thermocouple devices function independently of ambient light, whereas the present platform is constrained by the photon requirement of the RGB sensor. This gap motivates the infrared-camera upgrade outlined in Section VI.

VI. CONCLUSION AND FUTURE DIRECTIONS

This paper presented the design, implementation, and experimental evaluation of an edge-based intelligent fire detection and suppression system developed using a Raspberry Pi single-board computer. The proposed system integrates real-time CNN-based visual fire detection, embedded GPIO-driven actuation, and SMTP-enabled remote notification within a single low-cost hardware platform. The experimental results demonstrate that the integration of deep learning and embedded computing can provide significantly improved fire-safety performance compared with conventional smoke- and temperature-based detection systems, particularly in terms of response speed, contextual understanding, and automated emergency response capability.

The developed system achieved a fire detection accuracy of 93% under illuminated indoor testing conditions, with an average end-to-end response latency of 1.30 seconds. This response time is substantially faster than that of traditional fire detectors, which typically rely on the physical propagation of smoke or heat before activation. In addition, the relay-controlled suppression mechanism and water pump achieved 100% operational reliability throughout all experimental trials. The SMTP-based notification module successfully delivered remote email alerts with attached photographic evidence in every tested scenario, while the system maintained a 0% false-positive rate against all evaluated non-fire visual stimuli.

DOI: 10.22413/ijastems/2026/v12/i5/3

Collectively, these results validate the effectiveness of the proposed architecture and confirm the suitability of edge-deployed deep learning for real-time fire safety applications.

Although the proposed system demonstrated strong overall performance, several opportunities remain for future enhancement. First, the integration of thermal or near-infrared imaging sensors alongside the existing RGB camera could improve detection capability under low-light or completely dark environments, enabling reliable round-the-clock operation in locations such as warehouses, server rooms, and industrial facilities. Second, extending the current email-based notification framework to a cloud-enabled Internet of Things (IoT) infrastructure using platforms such as Amazon Web Services, Google, or Microsoft would support additional functionalities including real-time video streaming, centralized event logging, and mobile push notifications for enhanced situational awareness.

Finally, the single-node architecture proposed in this work can be further expanded into a distributed multi-node network of interconnected Raspberry Pi units, with each node supervising an independent monitoring zone. Such an architecture would enable large-scale deployment with coordinated fire detection and zone-specific suppression control across extensive infrastructures including hospitals, commercial complexes, and multi-storey industrial buildings. Overall, the findings of this study demonstrate that low-cost edge AI systems have substantial potential to transform modern fire safety technology by providing faster, more intelligent, and autonomous emergency response solutions.

ACKNOWLEDGEMENT

The authors gratefully thank the management of Aditya College of Engineering, Madanapalli, the Head of the Department of Electronics and Communication Engineering, and the project coordinator for providing laboratory infrastructure and academic guidance. The fire-and-smoke image dataset used for model training was sourced from the Kaggle open-data repository.

REFERENCE

[1] L. Huang, C. Liu, B. Li and D. Zhang, "Fire Detection in Video Surveillance Using a Wavelet-CNN Based Method," Eng. Appl. Artif. Intell., vol. 112, p. 104801, 2022.

[2] A. S. Buriboev, H. Abdullaev, I. Nishonov, O. Khasanov and H. Cho, "Improving Fire Detection Accuracy Through Enhanced Deep CNN Models," Comput. Vis. Image Underst., vol. 238, p. 103884, 2024.

[3] Charanarur, Panem, et al. “Design Optimization-Based Software-Defined Networking Scheme for Detecting and Preventing Attacks.” Multimedia Tools and Applications, vol. 83, no. 28, Feb. 2024, pp. 71151–69.

[4] A. Elhanashi, S. Essahraui, P. Dini and S. Saponara, "Early Fire and Smoke Detection Using Deep Learning: A Comprehensive Review of Models, Datasets and Challenges," Appl. Sci., vol. 15, no. 2, p. 643, 2025.

[5] M. S. Sozol, M. A. Islam, M. R. Islam and M. M. Rahman, "Indoor Fire and Smoke Detection Based on Optimized YOLOv5 Model," Artif. Intell. Comput. Vis., vol. 6, no. 1, pp. 112–124, 2025.

[6] S. Bhardwaj and H. Kaur, "Real-Time Fire and Smoke Detection Using AI and Computer Vision," in Proc. IEEE ICTBIG, Indore, India, 2025, pp. 1–6.

[7] Lanjewar, Madhusudan G., et al. “Small Size CNN-Based COVID-19 Disease Prediction System Using CT Scan Images on PaaS Cloud.” Multimedia Tools and Applications, vol. 83, no. 21, Jan. 2024, pp. 60655–87.

[8] A. G. Howard et al., "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," arXiv:1704.04861, 2017.

[9] R. David et al., "TensorFlow Lite Micro: Embedded Machine Learning for TinyML Systems," Proc. Mach. Learn. Syst., vol. 3, pp. 800–811, 2021.

[10] G. Bradski and A. Kaehler, Learning OpenCV: Computer Vision with the OpenCV Library. Sebastopol, CA, USA: O'Reilly Media, 2008.

[11] S. K. Sinha, S. Kumar and A. Agarwal, "IoT-Based Fire Detection and Automatic Alert System Using Raspberry Pi and Sensor Fusion," Int. J. Eng. Res. Technol., vol. 9, no. 6, pp. 1245–1251, 2020.

[12] Tahilramani, Nikunj, et al. “Edge-Based AI Solution for Enhancing Urban Safety: Helmet Compliance Monitoring with YOLOv9 on Raspberry Pi.” Discover Internet of Things, vol. 5, no. 1, Mar. 2025, p. 25.

[13] M. A. Khan, M. F. Uddin and N. Gupta, "Various Ways to Achieve Fire Safety in Buildings Using IoT and Machine Learning: A Comprehensive Review," IEEE Sensors J., vol. 21, no. 9, pp. 10869–10883, 2021.

[14] B. U. Töreyin, Y. Dedeoğlu and A. E. Çetin, "Flame Detection in Video Using Hidden Markov Models," in Proc. IEEE ICIP, Genova, Italy, 2005, pp. 1–4.

[15] Panchbhai, Kamini G., et al. “Non-Destructive Beer Gravity Measurement Using Spectroscopy and Machine Learning with mRMR-Based Wavelength Selection.” Journal of Food Science and Technology, Oct. 2025.

[16] Kaggle, "Fire and Smoke Image Classification Dataset," [Online]. Available: https://www.kaggle.com. [Accessed: Mar. 2026].