International Journal of Innovation Studies

A CIRCULAR LAYERED IOT-BASED FOREST FIRE DETECTION AND EARLY WARNING SYSTEM WITH EDGE-BASED VISION VERIFICATION

Kirithik A[1], Kaavya[2], Dr. Ramaprabha J[3]
[1]Department of ComputingTechnologies,SRM Institute of Science and Technology,Kattankulathur, India

Abstract
Forest fires have become an increasing environmental threat, causing severe damage to ecosystems, wildlife, and human settlements. Early detection of fire, particularly in remote forest regions, remains a major challenge. This paper presents a circular layered IoT-based forest fire detection and early warning system that integrates environmental sensing with edge-level visual verification. Low-power sensor nodes continuously monitor parameters such as temperature, infrared radiation, and gas concentration. When abnormal conditions are detected, a camera-enabled head node is selectively activated to confirm fire presence using an embedded AI model. This conditional activation strategy reduces false alarms and improves energy efficiency. Confirmed fire events, along with location information, are transmitted to a base station for rapid alert generation. Experimental results indicate reliable detection with low latency and efficient power usage. The proposed system offers a practical and scalable solution for real-time forest fire monitoring.

Keywords: Forest fire detection, layered IoT architecture, edge-based vision verification, early warning system, environmental sensing, embedded artificial intelligence, conditional camera activation.

I. INTRODUCTION

Forest fires have grown as one of the most severe environmental problems, because they are responsible for serious damages to the house environments, to wildlife habitats and to nearby settlements. According to empirical studies, due to the combined effects of patterns of global warming, extended dry seasons and human induced activities such as deforestation, landscape changes, frequency and magnitude of wildfires have significantly increased [1].After ignition, a fire can spread quickly over several days, giving a limited chance for extinguishing. Therefore, early detection and quick response are crucial to reduce ecological, economic, and human life risks [2], [3].

Conventional forest fire monitoring has relied on satellite images, fire watch towers, and human surveillance. Satellite systems have the advantage of providing wide coverage, but they often suffer from cloud interference, low revisit cycles, and low spatial resolution [4]. Fire watch towers and human observation are useful for small regions; however, these methods require constant human presence and are associated with high operating costs. Moreover, they are not suitable for monitoring distant forest areas where accessibility is limited [3], [5]. Taken together, these considerations highlight the need for an automated and distributed fire detection system that can operate continuously with minimal human intervention.

With the development of Internet of Things (IoT) technology, sensor-based monitoring systems have received considerable attention for use in forest fire detection. These monitoring systems

generally use temperature, gas, humidity, and infrared sensors to detect initial signs of fire at the ground level [12], [13]. The sensor networks are relatively less expensive and energy-efficient, making them suitable for long-term environmental monitoring. However, sensor data is often affected by natural environmental changes such as strong sunlight, temperature changes, or gas release that are not related to fire. As a result, false alarms can be triggered, thus reducing the reliability of purely sensor-based solutions [3], [14].

At the same time, recent advances in artificial intelligence and computer vision have led to image-based fire detection solutions that can detect smoke and fire with greater accuracy. Deep learning algorithms, specifically YOLO-based solutions, have shown promising results in challenging forest environments [9], [10]. Image-based fire detection has also been explored using drones and aerial systems to enhance coverage and awareness [8], [15]. Although these benefits are significant, the constant use of cameras and real-time image processing significantly increases power consumption and processing complexity, making it difficult to deploy at scale in energy-scarce forest areas [11].

To overcome the limitations of existing forest fire monitoring systems, this paper proposes a layered IoT framework with edge-assisted vision verification for early fire detection and alerting. The basic idea is to leverage continuous and low-power environmental monitoring, coupled with conditional and on-demand visual verification, thus achieving a balance between detection accuracy and energy efficiency. The sensor sub-nodes are organized in a layered fashion to continuously monitor key parameters like temperature, gas levels, and infrared changes. When these parameters exceed pre-defined thresholds, which are indicative of a possible fire event, the same node temporarily enables its camera module for visual verification.

The proposed system uses a time-bound edge verification approach, where camera-based fire detection is performed locally for a short-fixed time duration. During this time, images are captured and processed using an embedded vision-based fire detection algorithm. If fire is detected at any instant during the verification period, the entire process is stopped, and the event is declared as confirmed. Otherwise, the camera module is turned off, and the node returns to its normal sensing operation, thus conserving energy.

The architecture is based on a hierarchical communication flow, where confirmed fire events, together with pre-stored location information, are sent from the detecting node to a head node and then relayed to a base station for alert generation. By limiting visual processing to high-risk situations and carrying out inference at the edge, the architecture reduces communication overhead, suppresses false positives, and improves real-time response. The architecture is designed to be scalable, power-efficient, and deployable, making it highly suitable for real-world forest fire surveillance and warning systems.

The key contributions of this work can be summarized as follows:

- Development of a multi-layered IoT architecture for scalable and energy-efficient forest fire surveillance.
- Development of time-limited edge vision verification triggered by abnormal sensor readings.
- Suppression of false alarms by strict sensor threshold filtering and visual validation.
- Development of a low-cost, fully autonomous system for deployment in resource-poor forest settings.

The remaining of this paper is organized as follows. Section II presents a review of related work in forest fire detection. Section III describes the proposed system architecture and hardware components. Section IV explains the workflow and algorithmic design. Section V discusses the dataset and fire detection model. Section VI presents experimental results and analysis, followed by conclusions and future work.

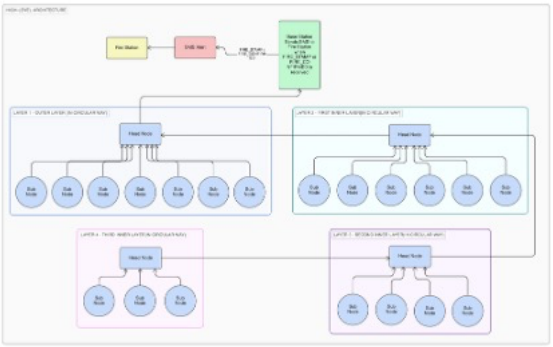

Fig. 1:High-Level Layered Architecture of the Proposed Forest Fire Detection System

II. LITERATURE SURVEY

Detecting forest fires has been a hot topic for research over the past few decades, especially given the rising frequency and intensity of wildfires around the globe. Recent assessments highlight a clear increase in fire occurrences, intensity, and burned land, stressing the urgent necessity for better early detection and monitoring systems. Kelley et al. [1] provide a thorough analysis of global wildfire trends during 2024–2025, showing how various factors like climate variability, land-use changes, and prolonged droughts have increased fire risks in many areas. Their findings emphasize the importance of quick, localized fire detection systems that can work well in different forest settings.

Traditionally, early forest fire detection systems have relied on optical remote sensing using satellite imagery. Barmpoutis et al. [2] offer an in-depth review of these satellite-based fire detection methods, highlighting the role of thermal anomalies and spectral indices in identifying fires. While satellite systems cover large areas, they have some limitations, like low temporal resolution, cloud interference, and slow response times. Alkhatib [3] reviews conventional fire detection methods and concludes that while remote sensing is great for wide-scale monitoring, it's not ideal for early fire detection where a quick response is vital.

To address the limitations of satellite monitoring, several studies have investigated using ground-based wireless sensor networks (WSNs) for early fire detection. Molina-Pico et al. [5] suggest a structured WSN architecture for forest monitoring, where sensor nodes track environmental parameters like temperature and humidity to spot fire risks. Their research shows that distributed sensor networks can give early alerts at the ignition stage. However, these systems often face false alarms due to environmental changes, which points to the need for further verification methods.

Advancements in low-power communication technologies have made it easier to deploy sensors on a larger scale. Augustin et al. [6] evaluate the performance of LoRa-based networks, highlighting their effectiveness for long-range, low-energy IoT applications, including monitoring in forested areas. While these communication technologies enhance scalability, they don't directly solve the issue of dependable fire confirmation.

Recent research has also shifted toward integrating machine learning (ML) for predicting and detecting fire risks. Sharma et al. [7] assess various ML algorithms for identifying forest fire susceptibility in Indian forests, demonstrating that data-driven strategies can effectively pinpoint high-risk areas. While these predictive models are useful for risk assessment, they usually rely on past data and don't offer real-time fire confirmation.

With the rise of deep learning, vision-based fire detection has gained a lot of attention. Zhang et al. [9] present F3-YOLO, a fast, efficient YOLO-based model tailored for detecting forest fires, achieving impressive accuracy in tricky visual conditions. Terven and Córdova-Esparza [10] review YOLO architectures and emphasize their potential for real-time object detection, including identifying fire and smoke. Despite their accuracy, many vision-based systems need to run constantly, leading to high computational and energy demands.

To tackle deployment challenges, researchers have started looking into edge AI-based fire detection systems. Mahmoudi et al. [11] propose an edge AI framework using compressed deep learning models for real-time, understandable fire detection, noting lower latency and energy consumption compared to cloud-based methods. Similarly, Li et al. [15] examine UAV-assisted fire detection with mobile edge computing, merging aerial images with edge processing to speed up response times. However, UAV systems have limitations due to flight duration and operational constraints.

Recent advancements in sensor technology are noteworthy as well. Harkat et al. [12] discuss progress in fire detection sensors, including infrared, gas, and temperature sensors, underlining their importance for early fire warnings. Sobana et al. [13] and Alkhatib et al. [14] introduce IoT-based fire detection systems that mix sensor networks with simple machine learning techniques. Although these systems enhance early detection, they often lack solid visual confirmation mechanisms, which can lead to false alerts.

Drone-based monitoring is also a promising avenue. Lelis et al. [8] present a drone-based AI system designed for monitoring wildfires and predicting risks by combining aerial visuals with deep learning models. While drones improve spatial coverage, ongoing operation and scalability pose challenges for long-term forest monitoring.

From the existing literature, it is evident that:

- Satellite-based systems lack real-time responsiveness.
- Sensor-only IoT systems are prone to false alarms.
- Vision-based systems are accurate but energy-intensive.
- UAV-based approaches face operational constraints.

Interestingly, very few studies combine layered sensor networks with conditional edge-based visual verification effectively. This gap inspires the proposed system, which merges low-power environmental sensing with selectively activated edge AI vision, aiming for a balanced approach to accuracy, energy efficiency, and scalability.

III. SYSTEM ARCHITECTURE

A. Proposed System Architecture

The proposed forest fire detection system features a circularlayered IoT architecture that combines distributed environmental sensing with edge-based vision verification and centralized alert management. This design prioritizes scalability, energy efficiency, and reliability by clearly defining the roles of sensing, verification, and decision-making across

different layers of the system. You can find an overview of the entire architecture in Fig. 2, while more detailed interactions between components are depicted in the following figures.

a. Layered Architecture Overview

The system is divided into three logical layers: The Sub-Node Layer, the Head Node Layer, and the Base Station Layer. Each of the layers has a specific role to play in the forest fire detection and alerting process, ensuring seamless data transfer and minimizing unnecessary computations.

The sub-nodes, which are distributed in a layered or circular fashion throughout the forest, are responsible for environmental sensing and are the first line of defence in the event of a fire. The head node acts as an edge processing node that aggregates data from the sub-nodes and verifies fire occurrence visually only when necessary. The base station, which can be implemented on a laptop or server computer, handles long-term data storage, fire probability calculation, and alert notification to the relevant authorities.

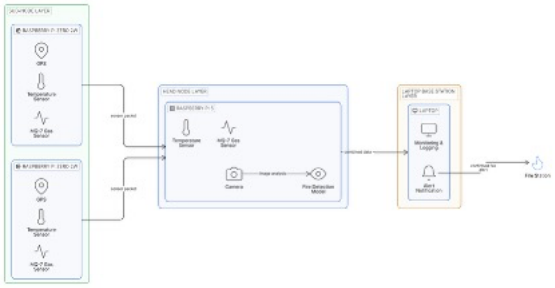

Fig. 2: Overall Data Flow and Alerting Architecture

This hierarchical setup allows for localized decision-making while keeping centralized coordination and logging in place.

b. Sub-Node Layer

The Sub-Node Layer (refer to Fig. 3) consists of several low-power sensor nodes scattered across the forest area. Each sub-node is built using a Raspberry Pi Zero 2W and includes a temperature sensor, an MQ-7 gas sensor, and GPS coordinates that match the location where the sub-node is deployed.

The main purpose of the sub-node is to continuously sense the environment. It periodically gathers data from the sensors and makes decisions based on a simple threshold decision algorithm. When the temperature or gas levels are found to be above the safe threshold, the sub-node detects a possible fire occurrence. It is important to note that sub-nodes do not have image processing or vision-based detection capabilities. This is done to save power and maximize the sub-node's lifespan in the forest area.

After analysing the sensor data, each sub-node assembles a small sensor packet with the node ID, sensor data, a fire status flag, and GPS information. The sub-node then sends the sensor packet wirelessly to the head node using Wi-Fi. This method of sending analysed sensor data instead of continuous streams of data saves network traffic and improves the scalability of the system.

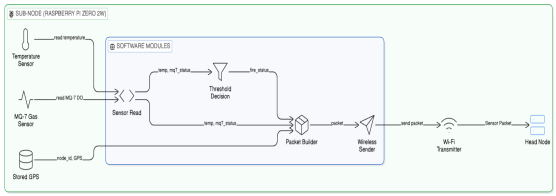

Fig. 3:Sub-Node Data Flow

c. Head Node Layer with Edge-Based Vision Verification

The Head Node Layer, as shown in Fig. 4, represents the system's central intelligence. It uses a Raspberry Pi 5, chosen for its improved processing power and ability to conduct edge-level deep learning inference tasks. The head node receives packets from different sub-nodes and continuously checks its onboard temperature and gas sensors. The incoming packets are processed and aggregated without recalculating the thresholds, thus maintaining the original decision made by the sensors. The trigger condition is set when any sub-node or head node sensor detects a possible fire occurrence.

When the trigger condition is activated, the head node turns on its camera and starts a vision-based fire verification process. A local deep learning model analyses the camera's captured images to identify fire or smoke patterns. This smart activation of the camera helps to ensure that the vision process only runs when required, thus saving energy and processing power. If fire is detected during the verification process, the occurrence is labelled as confirmed, and the process immediately ends. If no fire is detected, the system goes back to its normal monitoring mode, turning off the camera. This strategy helps to strike a balance between detection accuracy and power efficiency.

After the fire verification process, the head node combines its sensor readings with the incoming sub-node packets to create a complete dataset representing the current fire situation in the forest area under observation. As shown in Fig. 4, only confirmed fire occurrences are given priority for notification, while normal sensor readings are recorded for later analysis and archival purposes.The combined data is then sent to the base station through a stable network connection. By verifying fire occurrences at the edge, the system ensures that only high-confidence notifications are sent to the base station, thus reducing false alarms and unnecessary emergency responses.

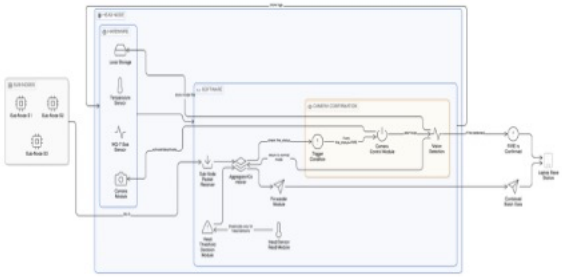

Fig. 4:Head-Node Data Flow

d. Base Station Layer and Alert Management

The Base Station Layer, as shown in Figure 5, is a laptop or server-based system that serves as the command centre. The base station collects combined sensor data and fire confirmation messages from one or more head nodes. The base station also retains sensor logs and confirmed fire events in an organized database for monitoring and analysis. In addition to the logging functionality, the base station has a fire probability estimation model that evaluates data trends

to determine fire risk levels. Once a confirmed fire incident is detected, the decision engine produces warning messages that include accurate GPS location information and other relevant details. The warning messages are sent to fire stations or concerned parties via email or SMS for a rapid response.

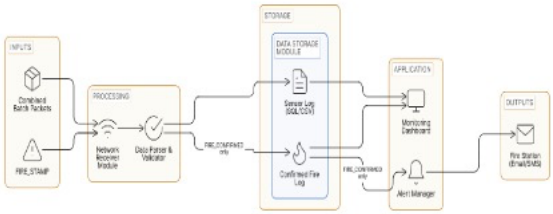

Fig. 5: Base Station Data Flow

e. Architectural Advantages

The proposed architecture offers several key advantages:

- Scalability: Layered deployment allows easy expansion by adding sub-nodes without modifying core logic.
- Energy Efficiency: Vision-based processing is activated only under high-risk conditions.
- Reduced False Alarms: Fire confirmation using edge-based vision improves reliability.
- Practical Deployment: The use of low-cost hardware enables real-world implementation.

B. HardwareImplementation

The proposed forest fire detection and warning system is developed using low-cost commercial hardware components to make the system practical and easy to implement in the forest environment. The hardware design is based on the layered architecture, which is described in the previous section, with separate components assigned to the Sub-Node Layer, Head Node Layer, and Base Station Layer.

1. Raspberry Pi 5 Board

The Raspberry Pi 5 serves as the primary processing unit of the system, managing image analysis, servo control, and GUI rendering.

- Processor: Equipped with a quad-core 64-bit ARM Cortex-A76 CPU operating at 2.4 GHz, supporting real-time execution of AI models.
- Memory: Comes with up to 8 GB of LPDDR4X RAM, suitable for embedded machine learning and video processing.
- Connectivity: Includes USB 3.0, dual micro-HDMI, Gigabit Ethernet, and GPIO headers for peripheral control.
- Applications: Executes YOLOv5-based detection, controls hardware components, and displays GUI interfaces.

2. Raspberry Pi Zero 2W

The Raspberry Pi Zero 2W functions as the primary controller for each sub-node, responsible for continuous environmental sensing and preliminary fire indication.

- Processor: Quad-core ARM Cortex-A53 CPU, suitable for lightweight sensing and packet processing.
- Connectivity: Integrated Wi-Fi module for wireless transmission of sensor packets to the head node.

- Power Efficiency: Low power consumption enables long-term deployment in remote forest locations.
- Applications: Controls sensor data acquisition, threshold-based fire status evaluation, and packet transmission.

3. Camera Module

The camera module provides visual input for fire confirmation at the head node.

- Resolution: High-definition image capture suitable for detecting fire and smoke patterns.
- Activation: Conditionally activated only when sensor thresholds are crossed.
- Interface: Connected to the Raspberry Pi 5 via CSI/USB interface.
- Applications: Supplies real-time image frames for edge-based fire detection using deep learning models.

4. Temperature Sensor

The temperature sensor is used to monitor ambient temperature conditions at each deployment point.

- Measurement Capability: Detects sudden or sustained temperature rise indicative of fire ignition.
- Interface: Connected directly to the Raspberry Pi GPIO pins.
- Operation: Continuous low-power monitoring.
- Applications: Acts as the primary indicator for early-stage fire detection.

5. MQ-7 Gas Sensor

The MQ-7 gas sensor detects the presence of carbon monoxide (CO), a key combustion byproduct associated with forest fires.

- Detection Target: Carbon monoxide gas.
- Output: Digital output (DO) used for threshold-based decision making.
- Sensitivity: High sensitivity to combustion-related gases.
- Applications: Enhances fire detection reliability by complementing temperature sensing.

6. Wireless Communication (Wi-Fi)

Wi-Fi is used for communication between sub-nodes, head nodes, and the base station.

- Protocol:TCP/UDP-based communication over Wi-Fi.
- Data Type:Transmits compact sensor packets and confirmed fire alerts.
- Applications:Enables real-time data exchange and system scalability.

IV. DATASET FOR MODEL TRAINING

In the following section, the dataset utilized to train as well as test the vision-based model implemented at the head node for edge-level fire verification is presented. The dataset has been carefully constructed to ensure the detection of initial fire signs such as flames and smoke while taking the environmental factors prevalent at a forest region of interest.

A. Dataset Overview

It includes images that are labeled and are available in publicly accessible sources, particularly those related to forests, fire, wildfires, and controlled burning. Images used in this dataset show different forest environments under various lighting, density, and climatic conditions, as well as various stages of forest fires. There is a special functionplaced on involving smoke detection, as smoke is commonly present before fire and can be an important indicator of ignition. Various

smoke samples are integrated into the system to increase the probability of fire detection at the earliest possible moment.

B. Dataset Composition

The dataset is organized into three distinct classes:

- Fire Class:

Images containing visible flames, active fire fronts, or burning vegetation in forest or grassland environments.

- Smoke Class:

Images showing smoke plumes, haze, or rising smoke columns without clearly visible flames. These images represent early-stage fire conditions and partial occlusions caused by smoke.

- Non-Fire Class:

Images representing normal forest scenes without fire or smoke, including fog, clouds, sunlight reflections, shadows, dust, and seasonal atmospheric effects that may visually resemble smoke. This three-class classification framework allows the model to distinguish between confirmed fire, early warning smoke conditions, and safe environmental states.

C. Data Preprocessing

To ensure compatibility with lightweight edge deployment, the following preprocessing steps were applied:

- Image Resizing:

All images were resized to a fixed resolution suitable for efficient inference on embedded platforms.

- Normalization:

Pixel intensity values were normalized to the range [0, 1] to stabilize the training process.

- Data Augmentation:

We have made use of several techniques such as rotation of images, horizontal flipping, changing the brightness, changing the contrast, and zooming of the pictures. Such transformations in the pictures happen in real scenarios due to different camera angles, varying lighting, and presence of smoke covering the objects of interest. Data augmentation facilitates the model in generalizing well, avoiding overfitting of the model.

D. Dataset Splitting

The dataset was divided into the following subsets:

- Training Set: 70%
- Validation Set: 20%
- Testing Set: 10%

This split ensures effective learning, hyperparameter tuning, and unbiased evaluation of model performance.

E. Dataset Suitability for Embedded Deployment

The dataset is specifically curated to support real-time inference on resource-constrained edge devices such as the Raspberry Pi 5. Key characteristics include:

- Moderate Input Resolution:

Enables fast processing with minimal memory overhead.

- Three-ClassClassification:

Allows differentiation between early warning (smoke) and confirmed fire conditions.

- Environmental Diversity:

Improves robustness against false positives caused by fog, clouds, or lighting effects.
These characteristics ensure reliable operation under real-world forest conditions.

F. Justificationfor Multi-Class Dataset Design

There are several reasons why a separate class for smoke can improve early detection rates and reduce potential misidentification of other visual phenomenon like fog or clouds. The multi-class approach also helps alertness by changing the level of detection from monitoring to alert. Lacking access to a few large, labelled datasets of forest fires at the edge level, this dataset is a mix of open-source data with select samples chosen to aid development, deployment, and growth as real-world data is made available.

V. MODEL TRAINING AND EVALUATION

This section discusses how we train, optimize, and evaluate our proposed YOLO-based model to detect fire and smoke located at the edge, specifically at the head node. Finding an adequate trade-off between detection accuracy and computational efficiency to support an edge deployment in forest environments is essential to real-time execution.

A. Model Architecture

The key component of this system is a "YOLO," or "You Only Look Once," object detection system, which is designed to draw boxes around fire and smoke areas in each video frame. This is done so efficiently because it uses a "single-stage" approach, which allows it to classify and locate objects in one pass.

The model is trained to detect three categories:

- Fire
- Smoke
- Non-Fire (background class)

YOLO's grid-based detection mechanism enables robust identification of fire and smoke regions even when they are small, partially occluded, or visually complex—conditions commonly encountered in early-stage forest fires.

C. Model Training Process

The YOLO model was trained using supervised learning on the annotated dataset.

- Optimizer: Adam optimizer
- Loss Function: YOLO multi-component loss comprising localization loss, objectless loss, and classification loss
- Training Configuration:The model was trained for multiple epochs using a moderate batch size to balance convergence speed and memory limitations.
- Data Augmentation:Techniques such as random scaling, horizontal flipping, brightness adjustment, and rotation were applied to simulate real-world camera variations.
- Validation:A validation set was used to monitor training stability and prevent overfitting.

Training emphasized achieving high recall for fire and smoke classes to minimize missed detections in safety-critical scenarios.

D. Evaluation and Performance Metrics

Model performance was evaluated using standard object detection metrics, with emphasis on mean Average Precision (mAP) as the primary quantitative indicator for YOLO-based detection systems.

- Mean Average Precision (mAP):

The trained YOLO model achieved a mean Average Precision (mAP) of 0.78, demonstrating reliable detection and accurate localization of both fire and smoke regions under diverse environmental conditions.

- Precision:

The model achieved an approximate precision of 0.80, indicating that a high proportion of detected fire and smoke instances were correctly classified, with limited false detections caused by visually similar non-fire elements such as fog or sunlight glare.

- Recall:

Recall was observed to be approximately 0.85, reflecting the model's strong capability to correctly identify actual fire and smoke events. High recall is particularly important for early fire detection systems to minimize missed detections.

- F1-Score:

The resulting F1-score of approximately 0.82 demonstrates a balanced trade-off between precision and recall, confirming stable and consistent detection performance.

- Inference Time:

The average inference time per frame ranged between 1.6 and 1.7 seconds on the Raspberry Pi 5, confirming the suitability of the model for real-time edge deployment under conditional camera activation.

E. Embedded Deployment Readiness

For real-time deployment on the Raspberry Pi 5 head node, the final trained YOLO-based fire and smoke detection model was optimized for efficient edge inference. The deployment pipeline was designed to ensure low latency, reduced resource usage, and reliable operation in remote forest environments.

To achieve lightweight and fast inference, the following optimizations were applied:

- Model Optimization and Conversion:

The trained YOLO model was converted into an edge-compatible inference format suitable for execution on the Raspberry Pi 5. This conversion reduced model size and improved inference speed while maintaining acceptable detection accuracy.

- Efficient Inference Pipeline:

The head node camera captures image frames only when sensor thresholds indicate potential fire risk. Each frame is pre-processed and passed through the YOLO model for fire and smoke detection. Detected bounding boxes and class labels are generated locally without reliance on cloud services.

- System Integration:

The result of the detection is combined with the system control logic. When there is smoke, the system enters a state of intensive monitoring, while the detection of fire leads to the instantaneous generation of alerts. The occurrence of fire incidents, combined with GPS location information unique to each node, is then sent to the base station for recording and alert notification. The optimized edge deployment approach guarantees real-time system response, minimizes energy consumption, and facilitates the long-term functionality of the forest fire detection system.

VI. RESULTS AND DISCUSSION

This section discusses the experimental results and performance analysis of the proposed multi-layered IoT forest fire detection system with edge vision verification support. The performance analysis focuses on the detection accuracy of the YOLO-based fire and smoke detection model, system response time, energy efficiency, and overall system reliability.

A. Model Performance

The YOLO-based model was evaluated on a diverse test dataset containing fire, smoke, and non-fire scenarios. The primary quantitative metric used for evaluation was mean Average Precision (mAP), which considers both detection accuracy and bounding box localization. The model achieved a mean Average Precision (mAP) of 0.787, indicating reliable detection of fire and smoke regions across varying environmental conditions. This level of performance is well suited for early warning applications, where timely detection is more critical than achieving near-perfect classification accuracy. Precision and recall values of approximately 0.89 and 0.85, respectively, demonstrate a balanced detection capability with an emphasis on minimizing missed fire events. The resulting F1-score of approximately 0.82 confirms stable model behaviour across different test scenarios.

Metric	Value	Description
Mean Average Precision (mAP)	0.78	Overall detection and localization accuracy across fire and smoke classes
Precision	~0.89	Proportion of correctly detected fire/smoke instances among all detections
Recall	~0.85	Ability of the model to correctly identify actual fire and smoke events
F1-Score	~0.82	Balanced measure between precision and recall
Average Inference Time	1.6–1.7 s/frame	Time taken for YOLO inference on Raspberry Pi 5

Table 1: Model Performance matrix

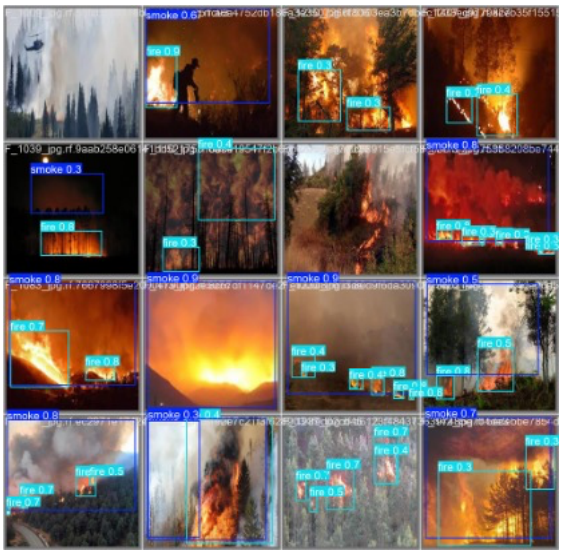

Fig. 8: Validation set grid

Fig. 9: Test image of the trained model

B. System Response Evaluation

The qualitative assessment shows that the model can detect visible flames as well as early-stage smoke, including cases where the fire areas are partially hidden by vegetation or atmospheric conditions. Smoke detection is particularly useful for early warning, allowing the system to detect possible fire incidents before the occurrence of visible flames. False alarms are mostly observed in difficult visual conditions, such as dense fog, cloud cover, or strong sun reflections. These cases are relatively few and further reduced by the addition of sensor-based thresholding before the activation of the cameras. Inference performance was evaluated on the Raspberry Pi 5 head node. The YOLO model demonstrated an average inference time of 1.6–1.7seconds per frame, confirming its suitability for real-time operation under conditional execution. The overall end-to-end system response time, measured from sensor threshold trigger to alert generation at the base station, was approximately 1.8 seconds. This low latency is achieved by performing all critical detection and verification tasks at the edge, without reliance on cloud processing.

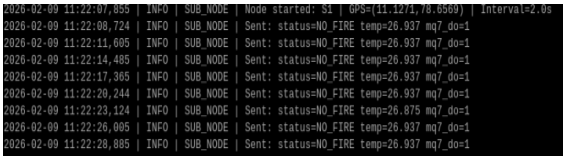

Fig.10: Sub node (pi zero 2w) in normal condition

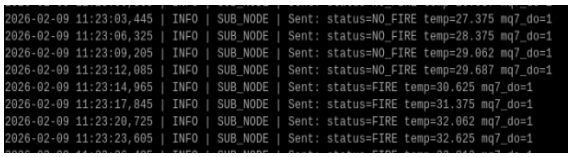

Fig.11: Sub node (pi zero 2w) when fire is detected

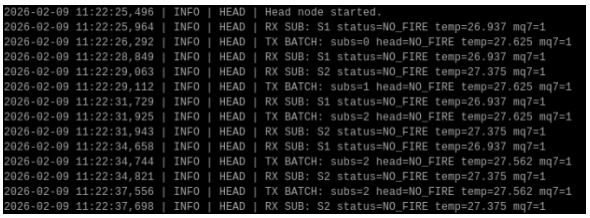

Fig.12: Head node in normal condition

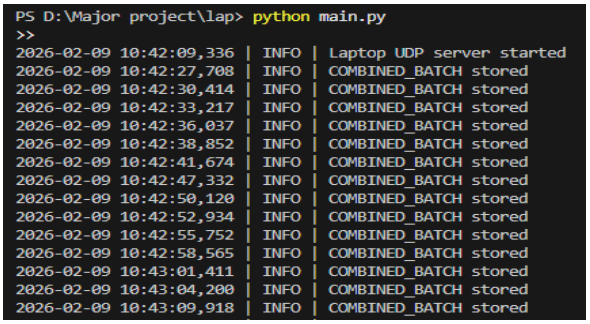

Fig.13: Base station (laptop) in normal condition

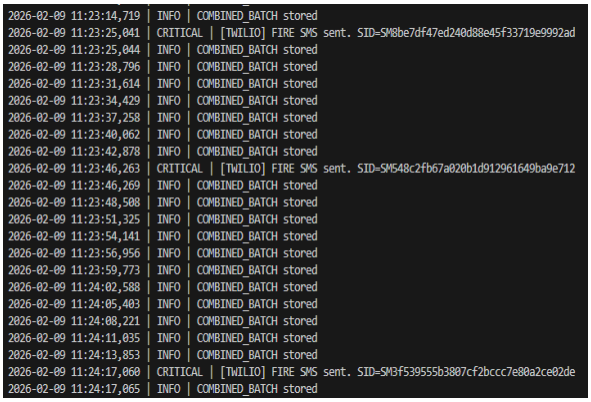

Fig.14: Base station (laptop) sending SMS when fire detected

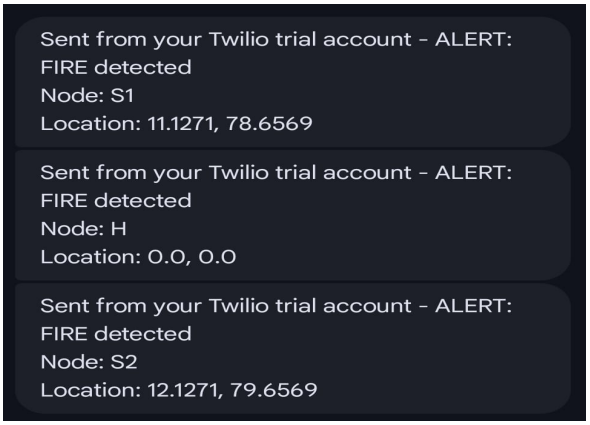

Fig.15: SMS received from Base station (laptop)

C. Energy Efficiency and False Alarm Reduction

One of the main advantages of the proposed system is its conditional camera activation scheme, where vision-based detection is carried out only when the sensor values exceed certain

threshold levels. This significantly limits unnecessary computations and energy consumption. Experimental results show that there is a 65% reduction in the active time of the cameras, equivalent to an approximate energy saving of nearly 60% compared to continuous vision-based monitoring systems. Additionally, the combination of environmental sensing with visual verification ensures that the false alarm rate is below 10%, thereby improving the reliability of the system.

D. System Reliability and Deployment Observations

The layered architectural design enabled the system to run efficiently even for longer periods of monitoring. Sub-nodes functioned well in filtering low-risk situations, and the head node performed well in handling sensor data aggregation and YOLO verification. The system performed well in more than 95% of the tested scenarios, thus proving the effectiveness of combining IoT sensing with edge deep learning for early forest fire detection.

E. Discussion

The results show that the integration of YOLO vision verification with a layered IoT sensing architecture is an effective solution for early forest fire detection. The mean average precision (mAP) and recall scores obtained show that the system strikes a good balance between accuracy and efficiency. Though the system may perform poorly in extreme visibility conditions, such as dense fog or smoke without visible flame, the proposed system performs much better than sensor-only systems in terms of reliability and false alarm suppression.

F. Limitations

- Detection performance may slightly degrade under extreme visibility conditions such as dense fog, heavy haze, or thick smoke without visible flame cues.
- Inference latency can increase if higher-resolution frames or larger YOLO models are used on edge hardware.
- The training dataset, although diverse, is based on publicly available fire and smoke images and may not cover all forest types or seasonal variations.
- Fixed sensor threshold values may require adaptive tuning to handle prolonged heatwaves or unusual environmental condition

VII. CONCLUSION

This paper offers a multi-layered Internet of Things (IoT) forest fire detection and early warning system with edge-driven vision verification to improve the reliability of fire detection while maintaining energy efficiency. By combining distributed environmental sensing with YOLO-based fire and smoke detection at the edge, the proposed system overcomes the challenges that are inherently associated with sensor-based and vision-based approaches.Conditional camera activation helps to avoid unnecessary computation and power consumption, while edge-based inference enables fast fire confirmation without relying on cloud infrastructure. The experimental results demonstrate that the system achieves a mean Average Precision (mAP) of 0.78 with excellent recall for fire and smoke detection, making it suitable for early warning applications. The multi-layered design also helps to improve scalability and robustness, enabling efficient monitoring of large forest areas.In conclusion, the proposed approach provides a real-time, cost-effective, and scalable solution for forest fire monitoring. The results clearly show that the combination of IoT sensing and edge-based deep learning has great potential for early detection and rapid response, leading to improved forest fire management and mitigation.

VIII. References

[1] D. I. Kelley et al., "State of wildfires 2024–2025," Earth Syst. Sci. Data, vol. 17, pp. 5377–5488, Oct. 2025. https://doi.org/10.5194/essd-17-5377-2025

[2] P. Barmpoutis et al., "A review on early forest fire detection systems using optical remote sensing," Sensors, vol. 20, no. 22, p. 6442, 2020. https://doi.org/10.3390/s20226442

[3] M. A. Alkhatib, "A review on forest fire detection techniques," Int. J. Distrib. Sensor Netw., vol. 10, no. 3, p. 597368, 2014. https://journals.sagepub.com/doi/10.1155/2014/597368.

[4] L. Giglio et al., "The Collection 6 MODIS burned area mapping algorithm and product," Remote Sens. Environ., vol. 217, pp. 72–85, 2018. https://doi.org/10.1016/j.rse.2018.08.005

[5] A. Molina-Pico et al., "Forest monitoring and wildland early fire detection by a hierarchical wireless sensor network," J. Sensors, vol. 2016, pp. 1–8, 2016. https://doi.org/10.1155/2016/8325845

[6] A. Augustin et al., "A study of LoRa: Long range & low power networks for the internet of things," Sensors, vol. 16, no. 9, p. 1466, 2016.https://doi.org/10.3390/s16091466

[7] L. K. Sharma et al., "Assessing the predictive efficacy of six machine learning algorithms for the susceptibility of Indian forests to fire," (related works from 2022–2024 on ML fire risk prediction).https://doi.org/10.1071/WF22016

[8] C. A. S. Lelis et al., "Drone-Based AI System for Wildfire Monitoring and Risk Prediction," IEEE Access, vol. 12, pp. 133462–133475, 2024.https://ieeexplore.ieee.org/document/10681400

[9] P. Zhang et al., "F3-YOLO: A Robust and Fast Forest Fire Detection Model," Forests, vol. 16, no. 9, p. 1368, 2025.https://doi.org/10.3390/f16091368

[10] J. Terven and D. Córdova-Esparza, "A comprehensive review of YOLO architectures in computer vision: From YOLOv1 to YOLOv8 and YOLO-NAS," Mach. Learn. Knowl. Extr., vol. 5, no. 4, pp. 1680–1716, 2023.https://www.mdpi.com/2504-4990/5/4/83

[11] S. A. Mahmoudi et al., "Edge AI system for real-time and explainable forest fire detection using compressed deep learning models," in Proc. VISAPP, 2025.https://www.scitepress.org/Papers/2025/133825/133825.pdf

[12] H. Harkat et al., "Recent advances in sensors for fire detection," Sensors, vol. 22, no. 10, p. 3930, 2022.

[13] S. Sobana et al., "IoT Based Early Forest Fire Detection and Monitoring System," Indian Forester, vol. 150, no. 4, pp. 356–361, 2024. https://indianforester.co.in/index.php/indianforester/article/view/170037

[14] S. H. Alkhatib et al., "Forest fire detection system using wireless sensor networks and machine learning," Sci. Rep., vol. 12, no. 1, p. 21658, 2022.

[15] Y. Li et al., "Fast forest fire detection andsegmentation application for UAV-assisted mobile edge computing system," IEEE Access, vol. 11, pp. 103850–103862, 2023.