Article

A Lightweight Fire Detection Framework Based on YOLOv11 for Deployment on Embedded UAV Systems

Venkat Bhardwaj * , Akansha Akansha , G. Renuka Devi

Posted Date: 13 February 2026

doi: 10.20944/preprints202602.1002.v1

Keywords: drone; forest fire detection; arduino UNO; raspberry-pi; object detection; YOLOv11n

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 February 2026 doi:10.20944/preprints202602.1002.v1

Article

A Lightweight Fire Detection Framework Based on YOLOv11 for Deployment on Embedded UAV Systems

Venkat Bhardwaj [1,*], Akanksha [1] and G. Renuka Devi [1,2]

[1] School of Engineering, Jawaharlal Nehru University, New Delhi, India
[2] IEEE Senior Member
* Correspondence: venkat37_soe@jnu.ac.in

Abstract

Forest fire poses a significant global threat, endangering ecosystems, biodiversity and the harmony of natural habitat. As the forest fires becomes more frequent and intense, the loss of natural habitat has increased rapidly. Therefore, it becomes important to detect forest fire at an early stage so that prompt and effective measures can be implemented immediately. To address this, we propose a low-cost drone-based forest fire detection and monitoring system that utilizes computer vision and deep learning for real-time forest fire detection. The proposed system uses a custom-built Quadcopter, equipped with the Raspberry pi camera module, to capture the real-time feed. These feeds are analyzed by the YOLOv11n object detection algorithm to accurately detect the fire. The proposed work utilizes an open-source dataset from Roboflow which contains 3426 images. This developed system is tested in various controlled environments demonstrating high mAP score of 93.6% in detecting wildfire. The YOLOv11n model achieves an accuracy of 92% and an approximate 8 frames per second for the test experiment. Therefore, it can be an effective tool for early wildfire detection, notifying timely alerts and aiding in rapid response, which can significantly enhance wildfire prevention and control efforts.

Keywords: drone; forest fire detection; arduino UNO; raspberry-pi; object detection; YOLOv11n

---

1. Introduction

Forest fires have emerged as one of the most critical environmental challenges of the 21st century, severely impacting ecosystems, burning vast expanses of land, threatening biodiversity, disrupting habitats, and contributing to global climate instability. One of the major causes of forest fires is human-induced causes accounting for approximately 90% of all reported incidents. These include agricultural practices like slash-and-burn, negligence like unattended campfires or discarded cigarettes, or sparks due to power lines or machineries operating near forest area. Natural causes, though less frequent, includes light striking, especially during dry seasons, or by friction between trees causing sparks due to high velocity wind.

As per the reports of Copernicus Atmosphere Monitoring Service (CAMS)[1], a total of 3.9 million square kilometers of land was burnt down globally during the 2023-24 wildfire season releasing approximately of 8.6 billion ton of Carbon dioxide in the atmosphere. The forest fires lead to a loss of 9 billion hectares of trees cover worldwide, making it the worst recorded wildfire season since the beginning of the century.

The most extensive wildfires were experienced by Bolivia and Guyana. Bolivia reported over 90,000 wildfire incidents, burning more than 10 million hectares of land—marking its worst fire season on record. Guyana recorded over 3,200 fire outbreaks, with approximately 1.3 million hectares of land affected. Venezuela recorded the highest carbon emissions in 2024 with over 44,000 wildfire outbreaks occurred affecting approximately 2 million hectares of land. Australia's Black Summer

(2019–2020) scorched approximately 18.6 million hectares, leading to the displacement/ death of over 3 billion species. The Amazon rainforest continues to experience recurring fire outbreaks, with over 100,000 incidents reported in 2020 alone, primarily due to deforestation activities. According to the India state of Forest report (ISFR-2023) [2], the number of fire hotspots detected by SNPP-VIIRS sensor were 2,03, 544 for the season 2023-24. The most affected states were Uttarakhand, Odisha and Chhattisgarh. Uttarakhand, in 2024, recorded 653 fire incidents and over 10,000 fire points detected via satellite imaging. These fires have significantly impacted forest cover, biodiversity-rich zones, and critical habitats including the Simlipal Biosphere Reserve in Odisha, a home to unique flora and fauna.

The rest of the paper is structured as follows. Section 2 reviews and summarizes relevant literature on forest fire detection systems. Section 3 describes the proposed drone network and deep learning model framework in detail. Section 4 describes evaluation and experimental results. Section 5 concludes the paper.

2. Literature Review

Therefore, it becomes important to detect the forest fires at early stages to minimize the ecological damage and protect the biological assets. Numerous works have been proposed to mitigate the effect of forest fires by early detection. Some of the traditional fire detection methods include manual patrolling, fire watch towers and satellite-based fire-detection that have been employed for monitoring for a long time. While these approaches are effective for early forest fire detection up to some extent, they do suffer some drawbacks. Manual surveillance and watch require substantial human effort, constrained by line-of-visibility, and may not be effective in remote or mountain terrains. Fire watch towers also suffer the same limitations.

Alexander et al. [3] proposed GOES-EFD (GOES – Early Fire Detection), which utilizes NOAA's GOES thermal infrared imagery to detect forest fires. The multitemporal anomaly detection flags pixels with significantly increased brightness temperatures and contextual comparison with neighboring pixels to reduce false positives. Zhanqing et al. [4] deployed the AVHRR (Advanced Very High-Resolution Radiometer) data at 3.8 μm and 11 μm bands and extracted their radiance values for each pixel to compared them to its background. Kaufman et al. [5] deployed MODIS, NASA’s EOS satellite to detect the thermal anomalies by comparing brightness temperatures at 4 μm and 11 μm for each pixel. Detected fires were further characterized by their size, intensity, and radiative energy including smoldering vs. flaming stages. Schroeder et al. [6] deployed VIIRS-based detection using I4 (3.74 μm) for high-temperature anomalies and I5 (10.76 μm) for background reference to reduce false positives from sun glint or urban heat. Quayle et al. [7] utilized shortwave infrared (SWIR) and near-infrared (NIR) bands of Landsat-8/OLI (Operational Land Imager) to identify Hot pixels using reflectance difference and band ratios, which were validated against surrounding pixels. Satellite-based fire detection, even though being a highly effective method providing large-scale monitoring capabilities, can be hindered by climatic conditions like cloud cover or low-quality images which reduce the reliability of detection. Moreover, most of the satellite-based systems do not offer continuous real-time monitoring and are ineffective during night-time or low light conditions. These limitations may lead to critical delays in identifying the forest fire allowing the fire to spread further.

To address the need for ground-level accuracy and real-time responsiveness, researchers began developing terrestrial solutions such as Wireless Sensor Networks (WSNs). It is a system of small, energy-efficient sensor devices that communicate wirelessly to monitor and collect data from a specific area providing real-time data. Hefeeda et al. [8] proposed a detection algorithm by organizing sensor nodes in the forest into a grid to measure temperature, humidity wind, rainfall and this data is used to calculate FMCC (ignition risk) and FWI (Fire Weather Index). Zhang et al. [9] proposed a cluster-based WSN, in which the data collected by sensors is sent to cluster heads to classify nodes into Coordinator and RFD nodes, using ZigBee (CC2430) for communication. The system integrated satellite data, aerial patrols, and manual observation, forming an effective ground-air detection network.

Though WSNs have improved wireless connectivity for environmental monitoring but they also suffer from certain drawbacks like limited power supply, spatial reach, processing and storage capacity. Their utility could be further amplified into evolving into Internet of Things (IoT) systems. IoT allows real-time monitoring and alerts, remote access, cloud-based analytics and decision making by further integrating it with Artificial Intelligence algorithms. Sutikno et al. [10] proposed an early forest fire detection method which consists of two main systems working together. The Sensor Nodes (with Arduino UNO) deployed in forests, consisting of various sensors (flame, DHT-22, MQ-7) to collect data and transmitted via RF-XBee (ZigBee protocol) to a Gateway Node (with NodeMCU ESP8266) which receives, processes, and further forwards the data to the Thinger.io cloud platform. Kanakaraja et al. [11] deployed the Ubidots IoT platform to enable real-time remote monitoring. A combination of sensors and an ESP-32 cam is used to collect data and capture images. A buzzer alert is generated, and authorities are notified via cloud whenever sensor values exceed preset thresholds. Morchid et al. [12] developed a three -layer architecture. The IoT device layer utilizes flame and gas sensors connecting to Raspberry- Pi 3 B+ for real-time data collection, the Cloud Layer utilizes ThingSpeak platform which receives data via MQTT and analyses it, and application layer to visualize data, integrating MATLAB and ThingSpeak. Though these systems being highly efficient in environmental sensing and cloud-based monitoring, they are often limited in spatial coverage and mobility. To overcome this, UAV (Unmanned Aerial Vehicle)-based systems are being explored, providing greater flexibility, coverage, and real-time aerial surveillance, which when integrated with deep learning algorithms, intelligently detect fire, smoke, and even human presence with high accuracy.

Yuan et al. [13] presented an image-processing based method by analyzing Infrared images captured by UAVs. It analyses heat distribution and employed Histogram-based Segmentation, Optical flow analysis to detect motion and Directional variation analysis to differentiate actual fire from other hot objects. Govil et al. [14] integrated terrestrial camera-based detection with satellite detection methods by using InceptionV3 architecture (via TensorFlow-Slim) to output smoke probability for each tile. Liu et al. [15] used GAN (Generative Adversarial Network) to generate synthetic fire and non-fire images. A Multi-stage Classification Pipeline then follows- HOG (Histogram of Oriented Gradients) and Adaboost classifier for high recall primary detection, CNN to extract features and SVM for classification which were trained using misclassified samples from Adaboost for improved accuracy. Sudhakar et al. [16] used Color-Based Segmentation to isolate fire regions from background, Motion-Based Filtering to distinguish between fire from static moving objects, Optical Flow Refinement to distinguish fire motion to drone-induced movements and IR–visual image fusion to retain overlapping regions. Zhao et al. [17] applied saliency detection (Bayesian method with CRF) for segmentation of core fire regions extracted using color moments, ASM energy, and entropy, and then classified into smoke or fire via logistic regression which are then fed into FireNet CNN. Pan et al. [18] fine-tuned the MobileNet-V2 model and applied discrete fourier transform to discard filters with nearly-zero magnitude. The trained model is converted to TensorFlow Lite format and probability of fire is predicted block-wise. Chen et al. [19] trained Deep learning models using early-fusion to concatenate RGB+IR before network and late-fusion with two separate CNN streams, merged at high-level features. Flame localization was done using MSER and non-max suppression on IR images. Allauddin et al. [20] also used transfer learning on the MobileNet model. Video streams from drones are sent to the server for processing. Alerts sent to the authorities via mobile app. Kumaran et al. [21] proposed an IoT-based autonomous drone system which can also be used for disaster management. A Raspberry-Pi camera scans for fire as drone moves in predefined path and as fire is detected it descends and activates fire extinguisher, then ascends and continues its patrol.

Yandouzi et al. [22] used DJI Mavic Air drone to capture video and tested several deep learning models, YOLOv6, YOLOv7, YOLOv8 and Faster R-CNN. Detected fire coordinates are sent to

Geographic Information System for mapping and alert generation. Jiao et al. [23] deployed YOLOv3-tiny model and their UAV was equipped with a visible/IR camera and DJI Manifold on-board computer utilizing Edge computing. Mohamed et al. [24] proposed AI-MEFDES: AI-Based Multi-Class Early Fire Detection and Extinguishing System. They deployed Transfer learning from pre-trained YOLOv5 weights to fine tune the model for it to detect multiple classes (fire/ person/ electric fire). A logic controller turns on extinguishers if no person is detected and if person is detected, notifying alerts are sent. Goyal et al. [25] used APM 2.5 drone kit, Raspberry Pi 4 and YOLOv3 for accurate fire/ smoke detection. Histogram-based segmentation and optical flow analyzer is used to identify potential fire pixels and filter out static heat sources. The identified frames are passed through YOLOv3 for accurate detection of fire/smoke. Zhao et al. [26] proposed the Fire-YOLO, that improved YOLOv3 by replacing Darknet-53 with EfficientNet, for better detection of small flames and smoke under diverse lighting conditions. Titu et al. [27] used YOLOv8 on a custom labelled dataset and Autodistill to train lighter models like YOLOv8n and DETR. The final YOLOv8 model is deployed with a Raspberry-Pi 5, connected with a Pi camera mounted on the UAV. Bahhar et al. [28] first applied lightweight CNN to filter out normal frames, followed by YOLOv5s detect smoke and YOLOv5l to detect fire.

Sarkar et al. [29] proposed a two-step framework; sprint UAVs move in grid and sandclock-shaped pattern and survey UAVs circling the fire to identify its boundary and estimate the affected area. Afghah et al. [30] proposed a Leader-Follower Approach using UAV coalition. Fixed-wing UAVs were used as leaders for wide area surveillance and Rotary UAVs (coalition) hover at low altitudes to perform the actual sensing and video recording tasks. Bushnaq et al. [31] modelled forest as a 2D grid using Markov stochastic process. Sensors are randomly placed in the forest and if M or more sensors report fire, the UAV enters the verification mode to reduce false positives. Jahan et al. [32] used Pixhawk PX4 flight controller integrated with a sensor suite (MQ, flame, DHT22). A servo-controlled mechanism deploys fire-extinguishing balls and a fuzzy-based backstepping controller (FBSC) for stable trajectory tracking under environmental disturbances. Anitha et al. [33] integrated Cat Swarm Optimization (CSO) with Long Short-Term Memory (LSTM) neural networks. EfficientDet processes image/video input from surveillance cameras and passes this data to Recurrent LSTM network to classify as fire/no fire. Hristov et al [34] deployed LoRaWAN sensors, Ground-based cameras to visually monitor the area for smoke or fire and issuing an early warning. Rotary-wing UAVs verify the presence of fire and fixed-wing UAVs are launched for wide-area assessment and fire spread tracking.

3. Material and Methodology

This section outlines the step-by-step process followed in the design and implementation of a custom-built quadcopter drone. The objective was to develop a fully functional and stable aerial platform capable of manual flight, capturing real-time visual data, which will later be used for detecting early signs of fire or smoke using on-board machine learning algorithms. The entire system was assembled from individual components, with both hardware and software configured manually to achieve reliable flight control and expandability.

3.1. Drone Assembly

The drone construction followed the YMFC-AL design approach, originally by Joop Brokking [35]. It was constructed using a standard X-configuration quadcopter frame to ensure flight stability and symmetrical weight distribution. Four DJI brushless DC motors of 920kV were mounted on the arms using vibration-damping soft mounts to absorb vibrations which reduce mechanical noise during flight. Each motor was connected to a 30A Electronic Speed Controller (ESC), which is used to regulate the power sent to each motor. All ESCs received power from a Lithium Polymer (Li-Po) battery via a Power Distribution Board to simplify power management.

The flight controller was developed using an Arduino Nano microcontroller, an open-source microcontroller platform utilized extensively in embedded systems. It is paired with an MPU-6500, a six-axis inertial measurement unit (IMU), which provides real-time orientation data through an on-board gyroscope and accelerometer. Gyroscope measures angular acceleration in degrees per second and is less sensitive to vibrations. Accelerometer is used to measure the acceleration, and is more sensitive to vibrations. Together they detect both angular and linear motion. A custom-built PWM (Pulse width modulation) reader circuit was used to interpret input signals from a 6-channel RC receiver, enabling manual control of throttle, pitch, roll, and yaw. The Arduino was programmed using the Arduino IDE, followed by ESC calibration and IMU sensor tuning. Propellers were installed in a counter-rotating configuration to maintain balance and directional control. After completing the wiring and final integration, a series of indoor test flights were performed to verify flight stability and responsiveness. PID values were manually tuned to improve hover precision and overall control.

To enable wildfire detection, a Raspberry Pi and camera module are mounted on the drone. A Raspberry Pi is a single board computer which runs on Linux-based operating system. Power is fed into the Raspberry pi by using a LM2596 DC–DC converter to decrease the voltage so that the module works smoothly. The camera feed is processed in real-time on the Raspberry Pi using a custom-trained YOLOv11n model to detect signs of forest fires. Figure 1 depicts the hardware setup of the proposed drone Network.

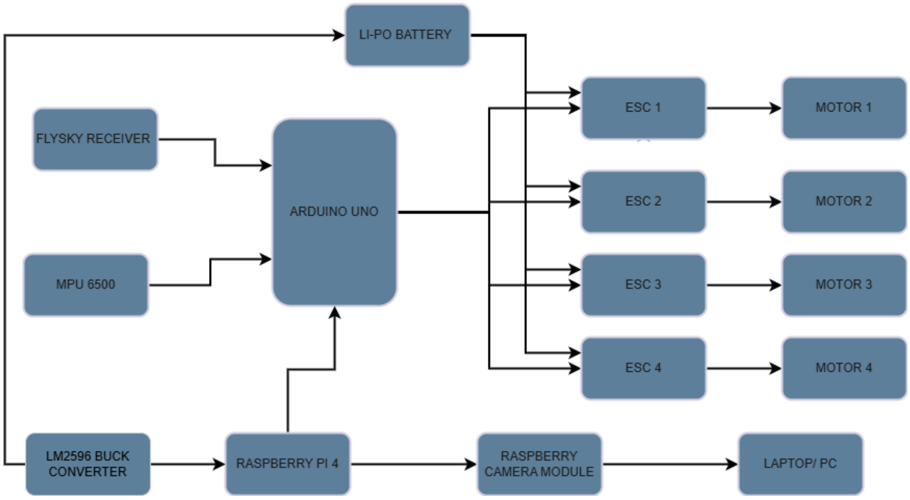


Figure 1. Hardware Setup of the Proposed drone Network.

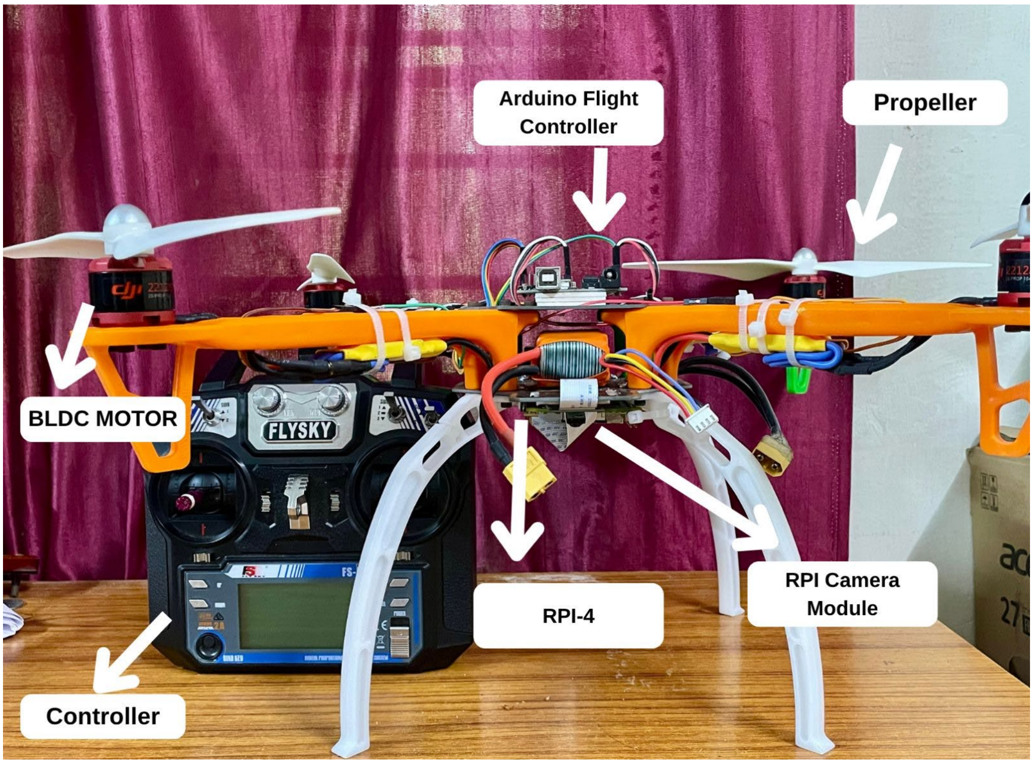


Figure 2. Proposed fire detection Drone with various components.

3.2. Training the Model

The dataset employed in this work was acquired from Roboflow [36], a widely used platform that offers tools and curated datasets and data preprocessing for a wide range of computer vision tasks. The dataset comprised a total of 3426 images, with 2056 fire and 1370 non-fire images. The dataset was further split into Training set of 2740 images (80%), test (10%) and validation set (10%). After data augmentation was applied, the total count of training images increased to 6576, with 4438 fire images and 2138 non-fire images. Tables 1 and 2 describes the dataset and applied augmentation techniques.

Table 1. Training images before and after augmentation.

Class	Before Augmentation	After Augmentation
Fire	1644	4438
No-Fire	1096	2138
Total	2740	6576

Table 2. Applied augmentation techniques and parameters.

Augmentation Technique	Parameters
Flip	Horizontal and Vertical
Zoom	± 15° and 90° clockwise and counter clockwise
Rotation	0% to 20 %
Saturation	±20%
Noise addition	Upto 0.5% pixels

The YOLOv11n [37] object detection algorithm is used for this study, which is a real-time deep learning based object detection algorithm that processes the entire image using single Convolutional Neural Network (CNN). It segments the input image into a grid of cells and each cell detects the concerned object whose center falls into that grid. YOLO predicts multiple bounding boxes and confidence score for each cell and then uses threshold filtering and non-max suppression to eliminate duplicate detections by retaining the boxes with the highest confidence scores.

The model was trained on Google Colab, a cloud-based platform that offers hosted Jupyter notebook service and also provides free access to GPUs and TPUs. The model was initialized with pretrained weights (yolo11n.pt) and modified for a single-class object detection task. It was trained for 50 epochs using a batch size of 16 and an input image resolution of 640x640. AdamW optimizer was used with a learning rate of 0.002 and momentum of 0.9. The model architecture has 319 layers with approximately 2.59 million parameters, and layers were partially frozen to preserve pretrained feature representations. Performance evaluation was conducted using standard object detection metrics such as precision, recall, mean average precision (mAP), and intersection over union. Table3 and table 4 shows the characteristics and hyperparameters for the applied YOLOv11n model.

Table 3. Characteristics of the applied YOLOv11n model.

Model Characteristics	Value
Layers	319
Parameters	2590035
Gradient	2590019
GFLOPs	6.4
Size (MB)	5.73

Table 4. Hyperparameters for the applied YOLOv11n model.

Model Characteristics	Value
Epochs	50
Image size	640×640
Batch size	16
Optimizer	AdamW
Learning rate	0.002
Momentum	0.9
Weight Decay	0.0005
Intersection over Union	0.7

4. Results and Discussion

The performance of the YOLOv11n model was evaluated using various evaluation metrics, including classification loss, bounding box loss, accuracy precision, recall, and mean Average Precision (mAP) at IoU thresholds of 0.5 and 0.5:0.95. Classification loss function measures the difference between predicted class probabilities and true class labels. The bounding box loss function measures the error between predicted bounding box and actual bounding boxes. Precision measures out of all the true predictions, how many are actually true. Recall is a measure of out of all the actual positives; how many did the model correctly identified. Intersection over Union (IoU) measures the amount of overlap between the predicted bounding boxes and corresponding ground truth bounding boxes. Mean Average Precision (mAP) evaluates both classification and localization of the object. It is calculated by plotting the Precision vs Recall curve, calculating the area under the curve (AUC) and take the average for each IoU threshold.

$$Accuracy = \frac{TP + TN}{TP + FP + TP + TN}$$

$$\mathrm{Pr}\,e\,cision = \frac{TP}{TP + FP}$$

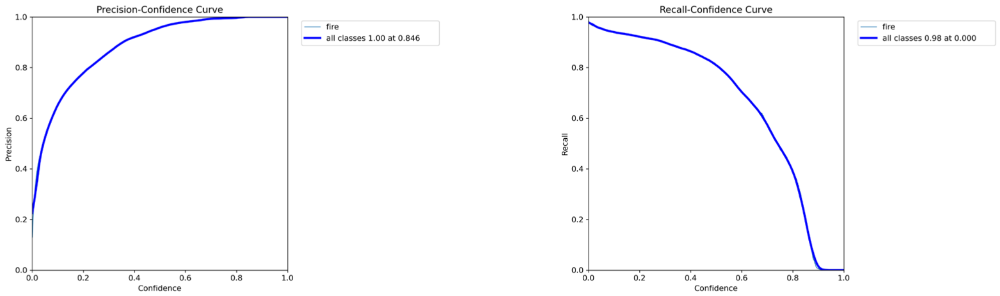
$$Recall = \frac{TP}{TP + FN}$$

$$F1\,Score = 2 \cdot \frac{\mathrm{Pr}\,e\,cision \times Recall}{\mathrm{Pr}\,e\,cision + Recall}$$

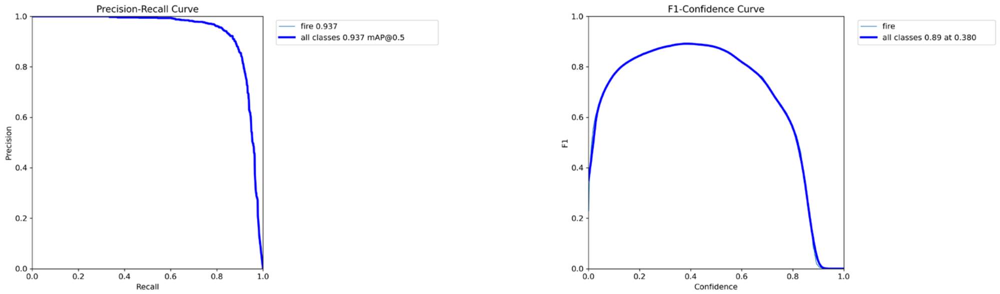
Figure 3 depicts precision, recall F1 score vs confidence for the YOLOv11n model. The maximum achieved value of precision is 0.93 indicating that the model can precisely detect the fire regions, reducing the false positives. The maximum achieved value of Recall is 0.869 indicating that the chance of missing actual fire instances is reduced. Other evaluation metrics such as training and validation losses, precision, recall, and mean Average Precision of the YOLOv11n model is shown in Figure 4. The maximum achieved value of mAP is 0.936 indicating that the model can accurately classify fire instances with both high precision and recall. The model is correctly classifying forest fire and accurately localizing it. Figure 5 depicts the confusion matrix of the trained YOLOv11n model.

Table 4. Evaluation metrics for YOLOv11n model.

Evaluation metric	Value
Precision	0.93
Recall	0.869
F1 score	0.898
Mean Average Precision	0.936



(a) Precision vs Confidence **(b)** Recall vs Confidence



(c) Precision Recall Curve **(d)** F1 Score vs Confidence

Figure 3. Evaluation metrics for YOLOv11n model.

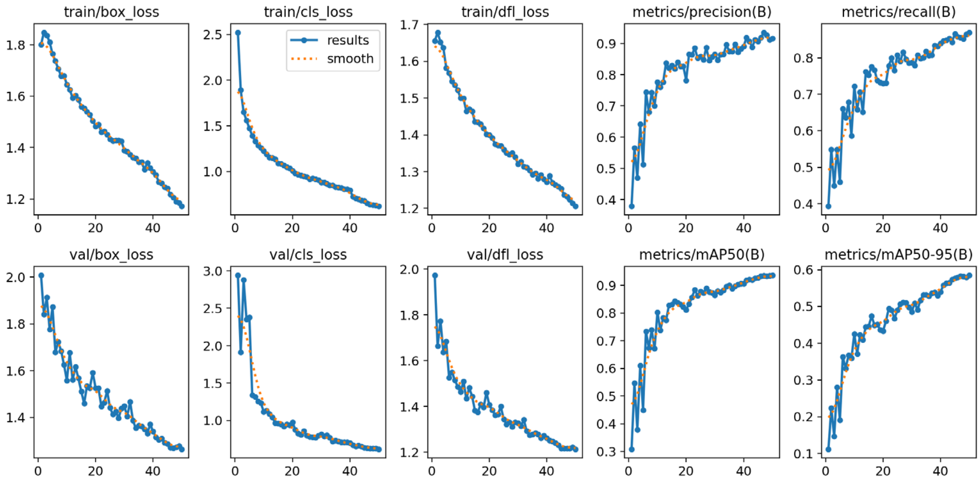


Figure 4. Various Performance metrics vs epochs for YOLOv11n model.

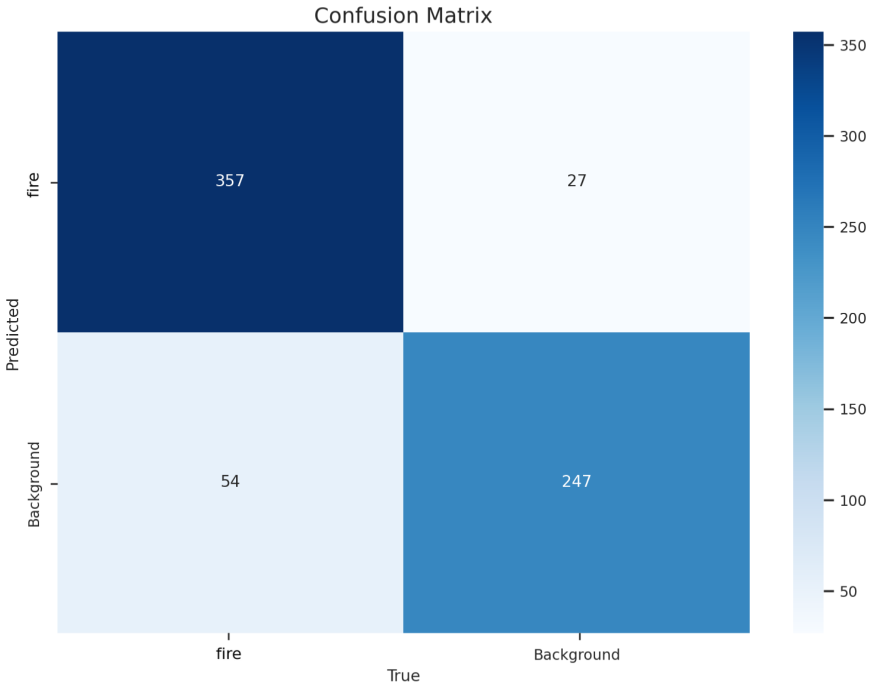
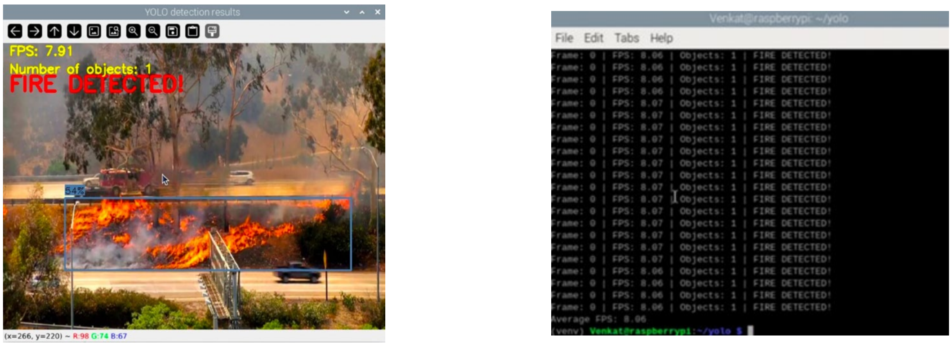


Figure 5. Confusion Matrix for the YOLOv11n model.

Due to regional restrictions between India and Pakistan, flying the drone was not permitted. So, we adopted an alternative strategy to evaluate the deployment feasibility of the forest fire detection system. The complete detection pipeline is tested using pre-recorded videos of forest fire sourced from publicly available datasets and online platforms. A total of 63 videos were fed into the system out of which the model was able to detect fire instances in 58 videos correctly, achieving an accuracy of 92.06%. The results are shown in the figures below. Figure 6 shows forest fire detection by the drone when a video is fed, and the output terminal confirming fire has been detected.



```
Frame: 0 | FPS: 8.06 | Objects: 1 | FIRE DETECTED!
Frame: 0 | FPS: 8.06 | Objects: 1 | FIRE DETECTED!
Frame: 0 | FPS: 8.06 | Objects: 1 | FIRE DETECTED!
Frame: 0 | FPS: 8.06 | Objects: 1 | FIRE DETECTED!
Frame: 0 | FPS: 8.07 | Objects: 1 | FIRE DETECTED!
Frame: 0 | FPS: 8.07 | Objects: 1 | FIRE DETECTED!
Frame: 0 | FPS: 8.07 | Objects: 1 | FIRE DETECTED!
Frame: 0 | FPS: 8.07 | Objects: 1 | FIRE DETECTED!
Frame: 0 | FPS: 8.07 | Objects: 1 | FIRE DETECTED!
Frame: 0 | FPS: 8.07 | Objects: 1 | FIRE DETECTED!
Frame: 0 | FPS: 8.07 | Objects: 1 | FIRE DETECTED!
Frame: 0 | FPS: 8.07 | Objects: 1 | FIRE DETECTED!
Frame: 0 | FPS: 8.07 | Objects: 1 | FIRE DETECTED!
Frame: 0 | FPS: 8.07 | Objects: 1 | FIRE DETECTED!
Frame: 0 | FPS: 8.07 | Objects: 1 | FIRE DETECTED!
Frame: 0 | FPS: 8.07 | Objects: 1 | FIRE DETECTED!
Frame: 0 | FPS: 8.07 | Objects: 1 | FIRE DETECTED!
Frame: 0 | FPS: 8.07 | Objects: 1 | FIRE DETECTED!
Frame: 0 | FPS: 8.06 | Objects: 1 | FIRE DETECTED!
Frame: 0 | FPS: 8.06 | Objects: 1 | FIRE DETECTED!
Frame: 0 | FPS: 8.06 | Objects: 1 | FIRE DETECTED!
Frame: 0 | FPS: 8.06 | Objects: 1 | FIRE DETECTED!
Average FPS: 8.06
(venv) Venkat@raspberrypi:~/yolo $
```

(a) Drone camera feed processed using YOLOv11n

(b) Terminal output on Raspberry-pi

Figure 6. Experimental results demonstrating forest fire detection on the videos fed into the model.

5. Conclusions

Early and efficient forest fire detection is essential to reduce environmental harm and economic losses. This work implements a low-cost drone-based forest fire detection method using computer vision tools. The YOLOv11n model was trained on a dataset of 3426 images covering various fire-scenarios using Google Colab. The model achieved a mean Average Precision (mAP) score of 93.6%, demonstrating its accuracy and effectiveness in detecting fire instances. A Raspberry pi camera was mounted on the drone to capture real-time data which is processed using YOLOv11n to detect fire instances. Due to current constraints, real-world drone deployment could not be performed. As an alternative, the detection system was tested using pre-recorded videos of forest fires. The proposed system offers an effective solution for early detection of forest fire, enabling rapid response and minimizing ecological damage. In future work, the system will be field-tested in actual forest environments, and integration with GPS, cloud connectivity, and alert systems will be explored for automated and scalable deployment.

Author Contributions: Conceptualization, V.B. and A.; Methodology, V.B. and A; Software, V.B.; Validation, V.B., A. and G.R.D.; Formal Analysis, A; Investigation, V.B.; Resources, V.B. and A.; Data Curation, V.B.; Writing – Original Draft Preparation, A.; Writing – Review & Editing, V.B. and G.R.D.; Visualization, V.B. and A.; Supervision, G.R.D ; Project Administration, G.R.D.; All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data will be made available on request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Copernicus Atmosphere Monitoring Service, "Home," [Online]. Available: https://atmosphere.copernicus.eu/. [Accessed: May 28, 2025].
2. Press Information Bureau, "Forest fire management and satellite tracking measures," Apr. 30, 2024. [Online]. Available: https://www.pib.gov.in/PressReleasePage.aspx?PRID=2086742. [Accessed: May 28, 2025].
3. Koltunov, A., Ustin, S. L., Quayle, B., Schwind, B., Ambrosia, V. G., & Li, W. (2016). The development and first validation of the GOES Early Fire Detection (GOES-EFD) algorithm. *Remote Sensing of Environment*, *184*, 436-453.
4. Li, Z. H. A. N. Q. I. N. G., Kaufman, Y. J., Ichoku, C., Fraser, R., Trishchenko, A., Giglio, L., ... & Yu, X. (2001). A review of AVHRR-based active fire detection algorithms: Principles, limitations, and recommendations. *Global and regional vegetation fire monitoring from space, planning and coordinated international effort*, 199-225.

5. Kaufman, Y. J., Justice, C. O., Flynn, L. P., Kendall, J. D., Prins, E. M., Giglio, L., ... & Setzer, A. W. (1998). Potential global fire monitoring from EOS-MODIS. *Journal of Geophysical Research: Atmospheres*, *103*(D24), 32215-32238.
6. Schroeder, W., Oliva, P., Giglio, L., & Csiszar, I. A. (2014). The New VIIRS 375 m active fire detection data product: Algorithm description and initial assessment. *Remote Sensing of Environment*, *143*, 85-96.
7. Schroeder, W., Oliva, P., Giglio, L., Quayle, B., Lorenz, E., & Morelli, F. (2016). Active fire detection using Landsat-8/OLI data. *Remote sensing of environment*, *185*, 210-220.
8. Hefeeda, M., & Bagheri, M. (2007, October). Wireless sensor networks for early detection of forest fires. In *2007 IEEE International Conference on Mobile Adhoc and Sensor Systems* (pp. 1-6). IEEE
9. .Zhang, J., Li, W., Yin, Z., Liu, S., & Guo, X. (2009, May). Forest fire detection system based on wireless sensor network. In *2009 4th IEEE conference on industrial electronics and applications* (pp. 520-523). IEEE.
10. Sutikno, T., Akbar, S. A., Bayusuta, T. A., Sabila, L. Y., & Satrian, P. H. (2024, September). Internet of Things-Based Early Forest Fire Detection System with Thinger. io Platform. In *2024 11th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)* (pp. 87-91). IEEE.
11. Kanakaraja, P., Sundar, P. S., Vaishnavi, N., Reddy, S. G. K., & Manikanta, G. S. (2021). IoT enabled advanced forest fire detecting and monitoring on Ubidots platform. *Materials Today: Proceedings*, *46*, 3907-3914.
12. Morchid, A., Oughannou, Z., El Alami, R., Qjidaa, H., Jamil, M. O., & Khalid, H. M. (2024). Integrated internet of things (IoT) solutions for early fire detection in smart agriculture. *Results in Engineering*, *24*, 103392.
13. Yuan, C., Liu, Z., & Zhang, Y. (2017, June). Fire detection using infrared images for UAV-based forest fire surveillance. In *2017*
14. Govil, K., Welch, M. L., Ball, J. T., & Pennypacker, C. R. (2020). Preliminary results from a wildfire detection system using deep learning on remote camera images. *Remote Sensing*, *12*(1), 166.
15. Liu, Z., Zhang, K., Wang, C., & Huang, S. (2020). Research on the identification method for the forest fire based on deep learning. *Optik*, *223*, 165491.
16. Sudhakar, S., Vijayakumar, V., Kumar, C. S., Priya, V., Ravi, L., & Subramaniyaswamy, V. (2020). Unmanned Aerial Vehicle (UAV) based Forest Fire Detection and monitoring for reducing false alarms in forest-fires. *Computer Communications*, *149*, 1-16.
17. Zhao, Y., Ma, J., Li, X., & Zhang, J. (2018). Saliency detection and deep learning-based wildfire identification in UAV imagery. *Sensors*, *18*(3), 712.
18. Pan, H., Badawi, D., & Cetin, A. E. (2020). Computationally efficient wildfire detection method using a deep convolutional network pruned via fourier analysis. *Sensors (Basel, Switzerland)*, *20*(10), 2891.
19. Chen, X., Hopkins, B., Wang, H., O'Neill, L., Afghah, F., Razi, A., ... & Watts, A. (2022). Wildland fire detection and monitoring using a drone-collected rgb/ir image dataset. *IEEE Access*, *10*, 121301-121317.
20. Allauddin, M. S., Kiran, G. S., Kiran, G. R., Srinivas, G., Mouli, G. U. R., & Prasad, P. V. (2019, July). Development of a surveillance system for forest fire detection and monitoring using drones. In *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium* (pp. 9361-9363). IEEE.
21. Kumaran, S., Raj, V. A., & Raman, V. R. (2023). IoT-based Autonomous Search and Rescue Drone for Precision Firefighting and Disaster Management. *International Journal of Advanced Computer Science & Applications*, *14*(11).
22. Yandouzi, M., Grari, M., Berrahal, M., Idrissi, I., Moussaoui, O., Azizi, M., ... & Elmiad, A. K. (2023). Investigation of combining deep learning object recognition with drones for forest fire detection and monitoring. *Int. J. Adv. Comput. Sci. Appl*, *14*(3), 377-384.
23. Jiao, Z., Zhang, Y., Xin, J., Mu, L., Yi, Y., Liu, H., & Liu, D. (2019, July). A deep learning based forest fire detection approach using UAV and YOLOv3. In *2019 1st International conference on industrial artificial intelligence (IAI)* (pp. 1-5). IEEE.
24. Mohamed, M., Bakheet, S., Madhi, N., Mesfer, F., Rashed, L., Rashed, M., & Ahmed, I. (2024, June). AI-MEFDES: AI-based Multi-class Early Fire Detection and Extinguishing System. In *2024 Advances in Science and Engineering Technology International Conferences (ASET)* (pp. 1-6). IEEE.

25. Goyal, S., Shagill, M., Kaur, A., Vohra, H., & Singh, A. (2020). A yolo based technique for early forest fire detection. *Int. J. Innov. Technol. Explor. Eng*, *9*, 1357-1362.
26. Zhao, L., Zhi, L., Zhao, C., & Zheng, W. (2022). Fire-YOLO: a small target object detection method for fire inspection. *Sustainability*, *14*(9), 4930.Tr
27. Titu, M. F. S., Pavel, M. A., Michael, G. K. O., Babar, H., Aman, U., & Khan, R. (2024). Real-Time Fire Detection: Integrating Lightweight Deep Learning Models on Drones with Edge Computing. *Drones*, *8*(9), 483.
28. Bahhar, C., Ksibi, A., Ayadi, M., Jamjoom, M. M., Ullah, Z., Soufiene, B. O., & Sakli, H. (2023). Wildfire and smoke detection using staged YOLO model and ensemble CNN. *Electronics*, *12*(1), 228.
29. Sarkar, M., Yan, X., Erol, B. A., Raptis, I., & Homaifar, A. (2021). A novel search and survey technique for unmanned aerial systems in detecting and estimating the area for wildfires. *Robotics and Autonomous Systems*, *145*, 103848.
30. Afghah, F., Razi, A., Chakareski, J., & Ashdown, J. (2019, April). Wildfire monitoring in remote areas using autonomous unmanned aerial vehicles. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)* (pp. 835-840). IEEE.
31. Bushnaq, O. M., Chaaban, A., & Al-Naffouri, T. Y. (2021). The role of UAV-IoT networks in future wildfire detection. *IEEE Internet of Things Journal*, *8*(23), 16984-16999.
32. Jahan, N., Niloy, T. B. M., Silvi, J. F., Hasan, M., Nashia, I. J., & Khan, R. (2024). Development of an IoT-based firefighting drone for enhanced safety and efficiency in fire suppression. *Measurement and Control*, *57*(10), 1464-1479.
33. Mahaveerakannan, R., Anitha, C., Thomas, A. K., Rajan, S., Muthukumar, T., & Rajulu, G. G. (2023). An IoT based forest fire detection system using integration of cat swarm with LSTM model. *Computer Communications*, *211*, 37-45.
34. Hristov, G., Raychev, J., Kinaneva, D., & Zahariev, P. (2018, September). Emerging methods for early detection of forest fires using unmanned aerial vehicles and lorawan sensor networks. In *2018 28th EAEEIE annual conference (EAEEIE)* (pp. 1-9). IEEE.
35. J. Brokking, “YMFC-AL – Your DIY Arduino-based flight controller,” [Online]. Available: http://www.brokking.net/ymfc-al_main.html. [Accessed: May 28, 2025].
36. Roboflow, “Roboflow documentation,” [Online]. Available: https://docs.roboflow.com/. [Accessed: May 28, 2025].
37. Ultralytics, “YOLOv11 models,” [Online]. Available: https://docs.ultralytics.com/models/yolo11/. [Accessed: May 28, 2025].