Research Article

International Journal of Advanced Technology and Engineering Exploration, Vol 12(125)
ISSN (Print): 2394-5443 ISSN (Online): 2394-7454
http://dx.doi.org/10.19101/IJATEE.2023.10102496

Enhanced forest fire detection using a hybrid VGG-16 and YOLOv5 model

Indira K[1], Abinaya S[2*] and Aishwarya Shaji[2]
Department of Computer Science and Engineering, Thiagarajar College of Engineering, Madurai, Tamil Nadu, India[1]
School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, Tamil Nadu, India[2]

Received: 03-November-2023; Revised: 28-March-2025; Accepted: 08-April-2025

Abstract

Forests are an essential natural resource for humans, offering a plethora of direct and indirect advantages. The existence of life on Earth and the increasing effects of global warming are significantly impacted by forest fires and other natural disasters. Therefore, automatic forest fire detection is a crucial area of research for disaster prevention. Early detection plays a vital role in strategizing fire suppression efforts and mitigating potential damage. Wildfires exhibit diverse shapes, textures, and colors, making them challenging to detect using standard feature extraction techniques. This study examined the application of computer vision algorithms based on AI for smoke and fire detection in images. Convolutional neural networks (CNNs) demonstrated superior performance over traditional computer vision techniques, such as image classification. However, despite their effectiveness, CNNs required extensive training time, posing computational challenges in real-time fire detection scenarios. To solve this issue, a new hybrid visual geometry group 16 (VGG-16) layer network and the You Only Look Once Version 5 (YOLOv5) model are used for video-based fire detection. The image classifier VGG-16 is utilized in this instance to identify and categorize whether fire is present in the input data. From the categorized output of the VGG classifier, the object detector YOLOv5, is then applied to detect fire. A special effort has also been made to compile and preprocess the dataset, which includes images of fires and non-combustible objects, with as many real-world representative situations as possible. The test results demonstrated that the YOLOv5 detection algorithm has an accuracy of 70% and the VGG image classifier has an accuracy of more than 90%. The accuracy of YOLOv5 was greatly improved to 84% with reduced training time after the hybrid VGG-16-YOLOv5 model was developed. False positives (FP) were also reduced because background mistakes were removed even in videos containing smoke. According to experimental results, the suggested models perform better than the state-of-the-art techniques currently in use.

Keywords

Forest fire detection, Computer vision, Convolutional neural networks (CNNs), VGG-16-YOLOv5 hybrid model, Real-Time fire detection.

1.Introduction

Forest fires are among the most destructive natural disasters, characterized by their sudden onset and complexity, making rescue operations challenging. They not only threaten forest resources but also endanger human lives and property. Globally, over 100,000 forest fires are recorded annually, affecting more than one million hectares of forested land. In China alone, over 730,000 forest fires have occurred in the past five decades, with the affected area surpassing 30 million hectares [1]. From a global perspective, forest fires are becoming more frequent in tropical regions while becoming less common in polar areas beyond 70 degrees latitude [2, 3].

Climate change is significantly altering global forest fire patterns, with rising temperatures, recurrent droughts, and reduced air humidity leading to a 20% increase in the duration of the fire season [2, 3]. The intensity and frequency of forest fires are also rising, expanding their spatial impact worldwide [4]. Given the constant risk of forest fires, the development of accurate forest fire forecasting technologies is crucial for early warning systems and disaster prevention.

As urbanization expands and human activities increase in forested regions, the risk of human-induced forest fires has risen significantly. Activities such as agricultural burns, campfires, and the use of machinery that can inadvertently ignite fires contribute to the majority of forest fire incidents worldwide. The complexity of forest fires presents

*Author for correspondence

challenges in accurate modeling and prediction. However, advancements in remote sensing technology have significantly improved the observation and monitoring of forest fires, enhancing early detection capabilities [5]. Remote sensing equipment utilizes various technologies and methodologies to monitor, identify, and assess fire-related phenomena from a distance, eliminating the need for physical contact.

Onboard fire detection sensors monitor the distribution and changes of vegetation on a regular basis on several NASA satellites, including Terra (Earth Observing System AM-1), advanced very high-resolution radiometer (AVHRR), visible infrared imaging radiometer suite (VIIRS), and moderate resolution imaging spectroradiometer (MODIS). Furthermore, longer lead forecast timeframes and lower geographical resolutions are provided by advances in numerical weather prediction and climate models, which may improve the forecasting of severe fire weather events [6]. These advancements have impacted the modelling of forest fires and related research concerns, such as the availability of adequate data. As a result, machine learning has been employed more frequently in scientific research and forest fire management in recent years.

One of the major challenges in forest fire detection is the need for real-time monitoring and early warning systems. These systems rely on advanced technologies such as the internet of things (IoT) and smart sensors to effectively detect and alert authorities about potential fire outbreaks in remote forest areas [7]. However, these methods often struggle to cover vast, inaccessible regions and are prone to human error and fatigue [8]. This limitation has driven the development of more sophisticated, technology-driven fire monitoring and detection systems. While satellite surveillance provides valuable insights, it also has drawbacks, including delays, high costs, and limited coverage [9]. Recent advancements in machine learning and remote sensing technologies have significantly improved forest fire detection accuracy and efficiency. These approaches integrate data from aerial photography, satellite imagery, and ground-based sensors to enable rapid fire ignition detection and predict fire risk with greater precision.

Traditional forest fire prediction methods primarily rely on machine learning and empirical statistical models [10]. Statistical analysis, correlation methods, and fire prediction models fall under statistics-based approaches, which involve collecting historical meteorological data and analyzing weather conditions, temporal patterns, terrain features, and fire frequency. These methods help determine cause-and-effect relationships, aiding in the identification of fire occurrence patterns.

Several researchers have contributed to forest fire prediction advancements. Barmpoutis et al. developed an early fire warning model using optical remote sensing and computational models, successfully predicting fire likelihood in developing countries based on weather data from multiple sources [11,12]. Pradeep et al. used Geographic Information Systems (GIS) to classify forest fire hazard zones in Eravikulam National Park, India [13]. Similarly, river flow was identified as a credible predictor of forest fire risks in Australia's Tasmania Wilderness World Heritage Area [14]. Gülçin and Deniz created a forest fire risk map for the Manisa region in Turkey using GIS-based analysis [15]. Maffei et al. combined thermal and multispectral remote sensing to forecast forest fire characteristics [16].

In machine learning-based fire classification, Jin and Lee conducted a comparative study on feature extraction and classification techniques, concluding that AdaBoost and Naïve Bayes were the most effective algorithms for low-dimensional and high-dimensional classifiers, respectively. Their approach achieved an impressive 97.33% accuracy on evaluation data [17].

However, classifying smoke and fire images remains a challenge due to the large parameter space required by deep learning architectures like visual geometry group 16(VGG-16), DenseNet, Inception, and Xception. Transfer learning offers a promising solution for handling high-dimensional data by leveraging knowledge from pre-trained models in related domains with larger datasets [18]. This approach enables deep architectures to be trained effectively, even with limited labeled images. Nonetheless, deep learning models perform best when trained on large datasets, as insufficient training samples can lead to overfitting and suboptimal performance due to local optima convergence.

Real-time fire detection and classification is an emerging field in wildfire monitoring and prevention. This domain utilizes machine learning for real-time

fire tracking, classification, and alerting, enabling faster and more accurate fire detection and response.
This study evaluates the effectiveness of three deep learning architectures for real-time forest fire detection: YOLOv5, single-shot detector (SSD), and region-based convolutional neural networks (R-CNN). To train these models for fire and smoke identification and segmentation, images from diverse urban and forest environments were used. However, experimental findings revealed that the YOLOv5 algorithm struggled to detect small, low-intensity fires. To address this issue, Tian et al. [19] modified the YOLOv5 structure by incorporating additional convolutional and max-pooling layers, significantly enhancing detection performance. Following these adjustments, SSD and Faster R-CNN exhibited good results for fire detection, achieving impressive accuracies of 99.88% and 99.7%, respectively [19].

As part of an improved fire detection strategy, Shi et al. employed real-time processing techniques to detect fire movement. Their approach integrated real-time data tracking, feature extraction, and classification to improve fire detection accuracy. Additionally, their study identified topographic parameters, vegetation characteristics, meteorological conditions, and transmission line parameters as key factors influencing the likelihood of wildfires and power line failures. The experimental results confirmed that the developed model accurately predicts the probability of wildfires and power line tripping in high-voltage power line corridors, demonstrating its practical applicability in fire prevention and mitigation [20].

The research gap in wildfire detection highlights the devastating and long-lasting impact of both natural and human-induced fires on the ecosystem. Timely and accurate detection of active wildfires is crucial for minimizing damage and preventing disasters. Deep learning-based approaches offer a promising solution for automated forest fire detection, enabling faster response times. However, individual deep learning models often suffer from lower accuracy and prolonged training times when used in isolation. Therefore, developing a hybrid model is essential to enhance accuracy, improve efficiency, and optimize real-time fire detection. The major contribution of the proposed approach using hybrid VGG-16-YOLOv5 model is as follows.

- To develop a high-performance deep learning model for early-stage fire prediction, enabling rapid detection and mitigation to minimize its impact.
- To develop an algorithm to classify the dataset using VGG-16.
- To implement the YOLOv5 algorithm and develop a computationally efficient deep learning model that processes the source image in a single pass through the neural network, segmenting it into grids for real-time object detection and fire localization.
- To develop a VGG-16 and YOLOv5 hybrid model for efficient feature extraction, and reduced training time.

The remainder of the paper is organized as follows: Section 2 presents the literature review, Section 3 details the methodology, Section 4 discusses the results, Section 5 provides a discussion based on the findings, and Section 6 concludes the study.

2.Literature review

Extensive research has been conducted on fire detection models, particularly for smoke and fire identification. With advancements in artificial intelligence (AI), numerous studies have explored the application of deep learning and machine learning for detecting smoke and fire in images. This study focuses on neural network-based models for fire and smoke detection, addressing the limitations of conventional point sensors, which are insufficient for covering large areas.

Forest fires release significant amounts of smoke into the atmosphere, making a functional smoke warning system crucial for minimizing casualties and damage. The increasing frequency of catastrophic wildfires, driven by climate change, poses severe risks to ecosystems, human populations, and economies if not detected and controlled promptly. Wildfire monitoring can be approached in two ways: detecting smoke or flames. Since smoke is the primary indicator of an impending wildfire, deep learning models and advanced early warning systems that detect smoke efficiently are essential for real-time fire prevention and mitigation.

Linear models were among the earliest mathematical approaches developed for machine learning-based wildfire prediction. The Poisson regression method was applied to predict future fire risks in the Sioux Lookout Forest in Northern Ontario [21]. Similarly, logistic regression was utilized to estimate line trip probability and assess wildfire risks along the Hubei power line corridor, establishing it as a reliable predictor of wildfires and line trips based on experimental data [22].

Several researchers have explored machine learning and sensor-based detection systems for fire monitoring. Dampage et al. implemented wireless sensors integrated with machine learning for forest fire detection [23]. Qiu et al. conducted a quantitative analysis of wildfire patterns in Northeast China's Daxinanling region using Landsat time-series data and machine learning models [24].

For enhanced fire detection accuracy, Xu et al. developed a hybrid approach combining YOLOv5 and EfficientDet for fire detection [25]. To further reduce false positives (FP), EfficientNet was incorporated as an additional learner to acquire global knowledge, improving detection performance by 2.5% to 10.9%, as demonstrated in Tien et al.'s study [26]. The final model was compared to Support Vector Machine (SVM), a commonly used benchmark model, with Kernel Logistic Regression achieving a prediction accuracy of 92.2% on both training and validation datasets.

Chang et al. developed a fire ignition likelihood map, a valuable tool for identifying high-risk zones for forest fuel treatments, firefighting resource allocation, and surveillance prioritization [27]. They also mapped tropical forest fire vulnerabilities in Vietnam's Geba National Park.

Logistic regression-based analysis has been widely applied in wildfire risk forecasting. Bisquert et al. and Oliveira et al. utilized logistic regression models to predict fire danger in Heilongjiang Province, China, and Mediterranean forests, respectively [28–30].

Recent research has introduced lightweight deep learning-based fire detection methods to enhance wildfire prediction accuracy [31]. Sathishkumar et al. conducted a deep learning-based study on forest fire detection [32]. Bui et al. applied a neuro-fuzzy GIS model combined with particle swarm optimization inference, improving the precision of fire risk assessments in tropical regions [33].

Kang et al. leveraged deep learning and geostationary satellite data to forecast forest fires, demonstrating the superiority of machine learning over traditional regression models for fire prediction [34]. Guo et al. used long short-term memory (LSTM) to build a back propagation neural network (BPNN) forecasting model, analyzing fire risk factors across seven Chinese provinces [35]. Fu et al. developed a recurrent thrifty attention network for remote sensing scene identification [36]. Mohajane et al. employed machine learning and remote sensing for wildfire prediction in the Mediterranean region, while Guo et al. used random forest to simulate forest fire effects in northern China [37, 38].

Advancements in deep learning have significantly improved autonomous feature extraction from large datasets, making it more effective than traditional machine learning for wildfire prediction, data recognition, regression, and classification. LSTM-based models have been increasingly applied for forest fire forecasting in India [39]. Additionally, research has explored LSTM, BPNN, and recurrent neural networks (RNNs) to develop deep convolutional-based smoke prediction techniques, further advancing fire detection accuracy and early warning systems [40].

These methods rely entirely on historical data, which, while effective in predicting forest fires, struggle with feature extraction, limiting classification precision, particularly when handling large datasets. Mountain fires are influenced by both temporal and spatial factors, with their likelihood depending on multiple contributing variables that persist over several days [41]. A key challenge in multivariate temporal prediction is efficiently capturing and utilizing correlations between these variables.

An update to YOLOv5 introduced adaptive anchor training using the K-means++ algorithm, enhancing detection performance and speed while reducing fire damage [42]. They applied alternative loss functions—complete intersection over union (CIoU) and generalized intersection over union (GIoU)—to three YOLOv5 models: small, medium, and large. They expanded their 4815-image dataset to 20,000 images using synthetic data augmentation. The improved model outperformed the baseline YOLOv5 by 4.4%, with the best results achieved using the CIoU loss function (mean accuracy: 86.0%, recall: 78.0%).

Additionally, an improved YOLOv3 model was proposed for day and night fire detection, featuring an expanded detection area and faster processing speed [43]. The study emphasized the importance of high-quality datasets for accurate fire identification, compiling 9,200 images from Google image archives and movie stills. Data augmentation techniques, such as image rotation, further increased the dataset size. Using a customized camera setup and YOLOv3, the model achieved a mean precision of 98.9% for real-time fire detection. However, distinguishing non-fire bright light sources (e.g., high-beam lamps) remains a challenge.

To address real-time fire risk analysis in surveillance videos, a novel anomaly detection architecture was developed, minimizing computational resource requirements [44]. Inspired by human behavior, the approach combines temporal and spatial analysis. Experimental results showed that combining these methods reduced false negatives (FNs), improving the detection of anomalous events in video streams. Binary classification models outperformed multi-class models for this task.

Object detection remains a complex challenge in computer vision, requiring precise segmentation and classification [45]. Recent deep learning advancements have significantly improved object identification and detection accuracy, with models evaluated based on inference time and detection precision. Two-stage detectors offer higher accuracy than single-stage detectors. YOLO, first introduced in 2015, has undergone multiple refinements, with YOLOv8 released in January 2023, delivering optimized inference time and high detection accuracy. Traditional autoregressive models often overlook dependencies among multidimensional features influencing forest fire risks. This study incorporates geographical factors (e.g., slope direction, elevation) and meteorological variables (e.g., atmospheric pressure, rainfall, wind speed, specific humidity, temperature, and normalized difference vegetation index (NDVI)) [46]. The temperature variable, a key meteorological factor, exhibits diurnal and seasonal variations, forming short-term and long-term recurrence patterns that are rarely explicitly modeled in machine learning.

Integrating multiple deep learning models enhances prediction accuracy and versatility compared to standalone approaches. A comprehensive solution for forest fire detection and prediction can be achieved by combining YOLO and VGG: YOLO contributes real-time object detection speed. VGG enhances feature representation for image classification. This hybrid model effectively detects fire incidents, processes large-scale images, and provides contextual information for assessing fire severity and spread.

3.Proposed method

The proposed system, as depicted in *Figure 1*, outlines the forest fire detection process. The overall workflow consists of the following stages:

1. Dataset collection
2. Preprocessing
3. Classification using VGG models
4. Object detection using YOLOv5
5. Model testing

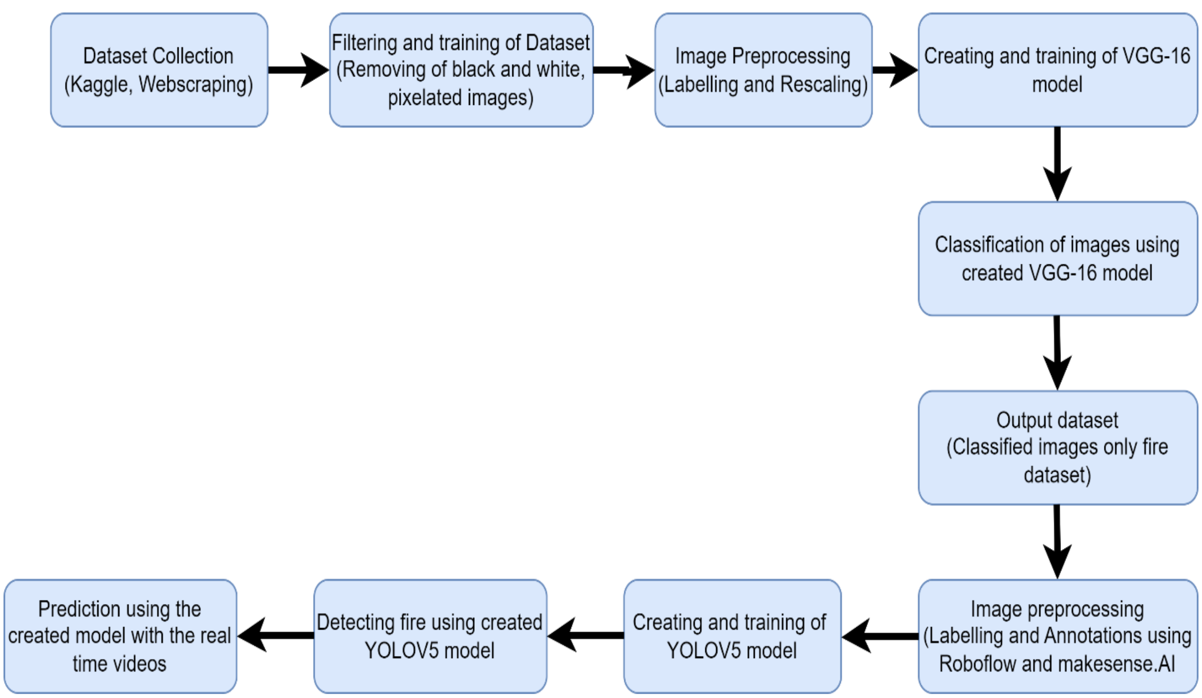

Figure 1 Forest fire detection system workflow

3.1Dataset collection

3.1.1Fire image dataset

The wildfire dataset used in this study was divided into three main partitions, all sourced from Kaggle [47]. The first partition initially contained 1,405 images, primarily featuring urban fire scenes. However, a significant portion of these images were irrelevant to the study's objective, including depictions of firefighters, fire equipment, maps, and cartoons. To improve the dataset quality, a thorough data cleansing process was carried out, which involved removing black-and-white images and those lacking visible smoke or fire. Following this process, 710 relevant images were retained, while 695 were discarded. The remaining images were then categorized into two main classes to support model training under diverse environmental conditions. The non-forest fire class included: forest, forest + clouds, and forest drought, while the forest fire class consisted of: nighttime forest fire, forest fire, and forest fire with smoke. The second partition comprised 999 images, of which 755 were fire images and 244 were non-fire images. Since most of the fire images were captured at night and the non-fire images during the day, this created a significant class imbalance. As a result, the non-fire images were excluded, and only the fire images were retained. Additionally, all fire images were converted from portable network graphics (PNG) to joint photographic experts group (JPG) format for uniformity. The third partition contained 110 fire images. After filtering out duplicates and non-relevant indoor fire scenes, only outdoor fire images—mostly captured during daylight—were kept. Upon merging all three cleaned partitions and removing redundancies, a final dataset of 1,665 fire images was compiled. Although this consolidated dataset enhances the model's robustness across varied fire scenarios, it still lacks complete representation of all possible wildfire conditions, indicating a need for further expansion. Overall, the three partitions were combined to form a comprehensive and diverse dataset optimized for training wildfire detection models.

3.1.2Non-fire images dataset

The non-fire dataset consists of an extensive collection of photos from the 8-Scene Categories Dataset, specifically selected to represent scenes without fire. It includes 2,688 images across various categories such as streets, tall buildings, landscapes, mountains, highways, coastal areas, and their respective subcategories. However, nearly all of these images were taken during the daytime, which introduces the risk that nighttime photos captured under artificial urban lighting could be mistakenly classified as fire-related. To mitigate this issue, an additional dataset containing 95 nighttime images, all depicting cityscapes without fire, was incorporated. Despite this effort, limitations remain, particularly due to the insufficient number of nighttime images. To address this shortfall and improve the balance between fire and non-fire image categories—especially under varied lighting conditions—web scraping was employed to collect supplementary images for both fire and non-fire classes. *Table 1* shows the categorization of the dataset.

Table 1 Categorization of dataset

Image group	Image dimensions	Number of samples	Use
High Resolution	1024 × 1024 pixels	300 images	High-detail images for deep learning models requiring fine-grained input
Medium Resolution	512 × 512 pixels	400 images	Balanced resolution for quality and computational efficiency
Low Resolution	256 × 256 pixels	300 images	Optimized for lightweight models focused on speed and memory efficiency

3.2Preprocessing

Every deep learning model requires input images to be of a specific size. Therefore, resizing operations are performed during the preprocessing stage to adjust each image to the required dimensions. Resizing modifies the actual width and height of an image to ensure uniformity across all inputs. The target image dimensions vary depending on the model architecture being used, and these requirements are detailed in *Table 2.*

By resizing all input images to a consistent shape—for instance, 150 × 150 pixels for certain CNN models—the input data becomes standardized, which improves processing efficiency and model performance. Additionally, normalization is applied to scale the pixel values to a fixed range, typically [0, 1]. This process helps stabilize and accelerate model convergence during training by ensuring that no single feature disproportionately influences the learning process.

To further enhance the training dataset and improve model generalization, data augmentation techniques are employed. These include random transformations such as flipping, translation, rotation, zooming, and brightness adjustment. Such transformations expose the model to a broader range of data variations, thereby making it more robust in real-world scenarios.

One specific augmentation technique is Color Jitter, which randomly alters the hue, saturation, and brightness of training images. This method increases the model's tolerance to color variations that may occur due to different lighting conditions. Additionally, maintaining a neutral color balance during preprocessing helps improve model performance under varying illumination environments.

Table 2 Target size of the images

Model name	Target size
VGG-16	(224, 224)

The dataset was divided into two segments: 80% was used for training the classifier, while the remaining 20% was reserved for testing and validation. To further evaluate model performance, K-fold cross-validation was implemented with k = 10 folds. In this technique, the dataset is randomly divided into 10 approximately equal-sized subsets. During each iteration, one-fold is used as the validation set, while the remaining nine folds are used for training. This process ensures a more reliable and generalized assessment of the model’s performance.

3.3Classification using VGG models

The VGG-19 model was initially trained on a dataset of 1,405 fire images for classification. To adapt the architecture for this task, the final fully connected layers were modified based on the output received from the earlier convolutional layers. These layers are located directly after the final max pooling layer, which generates 512 feature maps of spatial dimension 7 × 7.

For optimization, the Adam optimizer was used to improve convergence and training stability. A Dropout layer was incorporated to reduce overfitting by randomly deactivating neurons during training, thereby improving model generalization.

The custom classification head added on top of the pre-trained VGG-19 model included the following layers:

- Flatten: Converts the 3D output from the convolutional layers into a 1D vector.
- Dropout: Helps prevent overfitting during training.
- Dense (SoftMax): A fully connected layer using the SoftMax activation function to perform binary classification (fire vs. non-fire).

VGG-16 model for fire classification

In the main experiment, the VGG-16 model was used for fire image classification. This pre-trained CNN consists of 16 layers, including 13 convolutional layers and 3 fully connected layers. It uses small 3×3 convolution filters and max-pooling to extract hierarchical features. A total of 4,000 labeled images were used—2,000 fire images and 2,000 non-fire images—divided into 3,000 for training and 1,000 for testing, with balanced class distribution.

Due to the relatively small size of the dataset, the model was not trained from scratch. Instead, a pre-trained version of VGG-16, trained on the ImageNet dataset, was used for transfer learning. This approach utilizes previously learned feature representations and fine-tunes them on the fire dataset, allowing for faster convergence and better performance. Input images were resized to the VGG-16 input requirement of 224 × 224 pixels. The model was implemented using the Keras library. The classification head added to the base model included as shown in *Table 3*.

Table 3 VGG-16 model

Layer (Type)	Output Shape	Parameters
VGG-16 (Functional)	(None, 4, 4, 512)	14,714,688
Flatten	(None, 8192)	0
Dense (ReLU)	(None, 256)	2,097,408
Dense (Sigmoid/Softmax)	(None, 1)	257
Total Parameters	—	**16,812,353**
Trainable Params	—	2,097,665
Non-trainable Params	—	14,714,688

In this setup, the lower convolutional layers of the VGG-16 model were frozen, and only the upper layers were fine-tuned to adapt to the fire classification task. This strategy leverages the hierarchical feature extraction capabilities of CNNs, where lower layers capture basic features (edges, textures) and upper layers identify task-specific patterns.

Training was conducted on a system equipped with an Intel(R) Core™ i7-10750H central processing unit (CPU) @ 2.60 GHz, 64 GB random access memory (RAM), and an NVIDIA GeForce RTX 2070 general processing unit (GPU), ensuring optimal performance for training and inference.

To efficiently manage memory, an image data generator was used to load data in batches from disk to memory [48]. This was applied to both training and test datasets. During initialization, data augmentation was also applied to increase training data variability.

The augmentation techniques employed included rotation, translation, shearing, zooming, and horizontal flipping. Vertical flipping was deliberately excluded, as it is not relevant to wildfire imagery [49]. Data augmentation was applied exclusively to the training set, while the test and validation sets remained unaltered. The model was trained for 30 epochs, with validation data generated concurrently during training. Data augmentation enhances model robustness by exposing it to a variety of lighting and orientation conditions. Additionally, it serves as a form of regularization, helping to reduce overfitting and improve the model's ability to generalize to unseen data. The YOLOv5 object detection model also benefits from data augmentation, particularly in varying color, brightness, contrast, and lighting conditions. This enhances the model's adaptability to real-world environments and improves its accuracy in detecting fire across diverse scenarios. Together, this comprehensive training strategy strengthens the overall fire detection pipeline, improving both reliability and response time—critical for enhancing public safety and minimizing fire-related risks [50].

3.4Object detection using YOLOv5

YOLOv5 is a deep learning model designed for real-time object detection. It has demonstrated superior performance, particularly in detecting small objects, which is critical for industrial applications that require lightweight and stable models [51]. YOLOv5 is capable of detecting multiple object classes simultaneously and localizing their positions within an image.

3.5Testing of created model

The learning rate (LR) refers to the step size the model takes to minimize the loss function during training. It plays a crucial role in regulating how quickly the model learns and affects the overall training dynamics—a rate too high can cause the model to overshoot optimal solutions, while a rate too low can slow convergence or lead to local minima. In YOLOv5, the input layer of the network is designed to match the size of the input image, and pre-trained models generally require fixed image dimensions. If the input image exceeds this size, it is cropped or resized accordingly to maintain compatibility.

To train the YOLOv5 object detector, various optimization techniques were applied. The model was trained on a dataset of 3,500 forest fire images and evaluated on 500 unseen images to assess its performance. The implementation was carried out using the PyTorch framework.

Throughout training, the model effectively learned meaningful patterns while avoiding both overfitting and underfitting, achieving a balance between learning speed and accuracy. Maintaining consistency in input preprocessing, particularly image resizing, ensures the model can reliably extract key visual features across different datasets. Furthermore, rigorous evaluation on unseen data is essential for testing the model’s generalization and real-world applicability. These practices enhance the model's robustness and reliability, making it well-suited for deployment in dynamic environments such as wildfire detection systems.

3.6Hyperparameter optimization

To fully leverage the potential of deep learning models, selecting the appropriate hyperparameters is essential. An objective and effective approach involves evaluating multiple combinations of hyperparameter values and selecting the set that delivers the best performance on a given dataset. This process is known as hyperparameter tuning or hyperparameter optimization. The first step in any optimization process is to define the search space—the range of values over which each hyperparameter will be explored. Common search strategies include grid search, random search, and Bayesian optimization. In this study, Bayesian optimization was chosen to identify the optimal hyperparameters. Unlike traditional methods, Bayesian optimization leverages knowledge from previous evaluations to guide the selection of future candidates, significantly improving search efficiency. It is particularly effective in finding high-performing models with fewer evaluations compared to other methods [48].

As a result, Bayesian optimization was employed in the proposed work to fine-tune the selected hyperparameters. The tuned hyperparameters and their corresponding search spaces are presented in *Table 4.*

Table 4 Hyperparameter search space for bayesian optimization

Parameter	Search Space
Optimizer	Adam
LR	0.001
Batch Size	16
Number of epochs	10,15,20,25,30

4.Results and discussion

The experiments were conducted on Google Colaboratory, a cloud-based platform with GPU support, which served as the implementation environment. Google Colab seamlessly integrates with popular deep learning libraries such as PyTorch, Keras, and TensorFlow, as well as tools like Roboflow and makesense.AI for dataset preprocessing and annotation. For authentication, Google Authentication version 2.6.5 was used during the trials.

4.1VGG performance

The performance of the fine-tuned VGG-16 model is illustrated in *Figures 2* and *3*. In *Figure 2,* the training and testing accuracies are plotted against the number of epochs. The training accuracy (red) exhibits a steady increase, while the testing accuracy (blue) fluctuates, indicating variations in the model's generalization across epochs. *Figure 3* shows the training and validation loss(val_loss) curves as a function of epochs. The training loss (red) steadily decreases, whereas the val_loss (blue) fluctuates, suggesting variations in model performance and potential signs of overfitting.

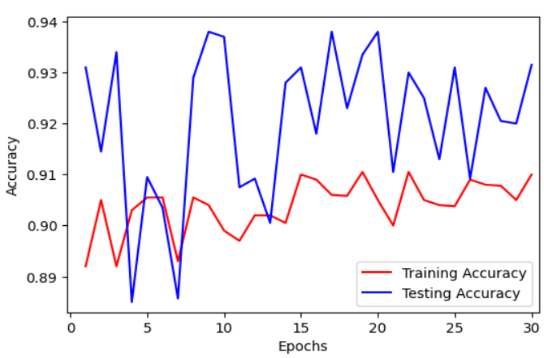

Figure 2 Performance of the fine-tuned model: training and validation accuracy over epochs

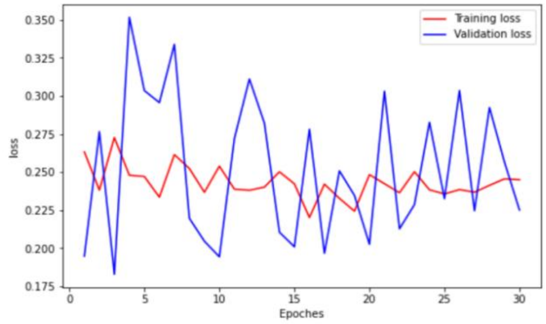

Figure 3 Performance of the fine-tuned model: training and validation loss over epochs

A concerted effort was made to prepare a dataset comprising fire and non-fire images, covering a wide range of real-life scenarios. Secondary data sources were utilized wherever available, and additional data was collected through web scraping to ensure comprehensive coverage. In addition to accuracy, the VGG-16 model was evaluated using several performance metrics: loss = 0.2450, accuracy = 0.9110, val_loss = 0.2251, and validation accuracy (val_acc) = 0.9320.

The predicted classes are represented in the columns of the confusion matrix shown in *Figure 4*: VGG-16, where each column corresponds to the categories assigned by the model to the input data. The confusion matrix provides a count of instances where the model's predictions match or differ from the actual labels, supporting the evaluation of key metrics such as F1-score, recall, precision, and overall accuracy. The confusion matrix results indicate: True Positives (TP) = 981, True Negatives (TN) = 963, False Positives (FP) = 20, and FNs = 38. These values offer valuable insights into the model's classification performance, enabling a more comprehensive assessment of its effectiveness.

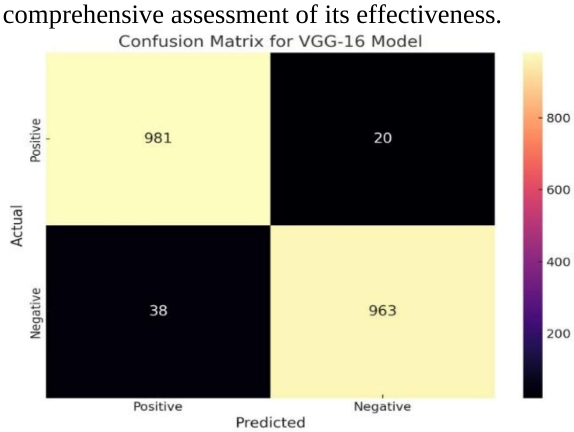

Figure 4 Confusion matrix of the fine-tuned VGG-16 model

The performance metrics considered are as follows:
Precision: Precision quantifies the accuracy of the model's positive predictions. It is defined as the ratio of correctly predicted positive instances to the total number of instances predicted as positive. This is shown in Equation 1:

$$\text{Precision} = \frac{TP}{TP+FP} \quad (1)$$

Sensitivity (Recall): Recall measures the model's ability to identify all actual positive instances in the dataset. It is calculated as the ratio of correctly predicted positive cases to the total actual positive cases (Equation 2).

$$\text{Recall} = \frac{TP}{TP+FN} \quad (2)$$

F1-score: The F1-score is the harmonic mean of precision and recall, providing a balanced measure of both metrics (Equation 3).

$$\text{F1} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (3)$$

4.2YOLO performance

Every model requires input images to be of a specific size. Therefore, image resizing was performed as part of the preprocessing step. Resizing is a technique that alters the original dimensions of an image to match the required input specifications. The target sizes for different models are listed in *Table 5*. The classified images used for training the neural network were labeled using online annotation tools such as Roboflow and makesense AI.

Table 5 Target size of the images

Model Name	Target Size
YOLO v5	(640, 640)

4.3Performance evaluation

Experiments were conducted using two strategies: a standalone YOLOv5 model and a hybrid model that incorporates VGG-16 for initial classification. As shown in *Table 6*, the results indicate that combining YOLOv5 with VGG-16 yields better performance compared to using YOLOv5 alone. The hybrid model demonstrated higher accuracy, reduced training time, and lower loss values, highlighting its improved efficiency and effectiveness.

Table 6 Comparison of YOLOv5 and hybrid model

Metrics	Only YOLOv5	Hybrid model
Accuracy	0.70	0.84
Epochs	10	10
train/box_loss	0.041567	0.055368
train/obj_loss	0.023187	0.050079
train/cls_loss	0	0
metrics/precision	0.40653	0.51518
metrics/recall	0.40695	0.46094
metrics/ mean Average Precision (mAP)_0.5	0.32919	0.45746
metrics/mAP_0.5:0.95	0.092572	0.19912
val/box_loss	0.066455	0.054104
val/obj_loss	0.039918	0.0175
val/cls_loss	0	0
x/lr0	0.00208	0.00208
x/lr1	0.00208	0.00208
x/lr2	0.00208	0.00208

4.4Model only with YOLOv5

The confusion matrix corresponding to the model developed using the YOLOv5 method is shown in *Figure 5*. The values from this confusion matrix—TP = 1.00, TN = 0.46, FP = 0, and FN = 0.52—provide insights into the model's performance. These metrics help evaluate the model's ability to correctly identify fire and non-fire instances.

Figure 6 presents the performance metrics of the model created using only the YOLOv5 algorithm. It serves as a visual representation of various evaluation metrics, offering a comprehensive overview of the model's effectiveness.

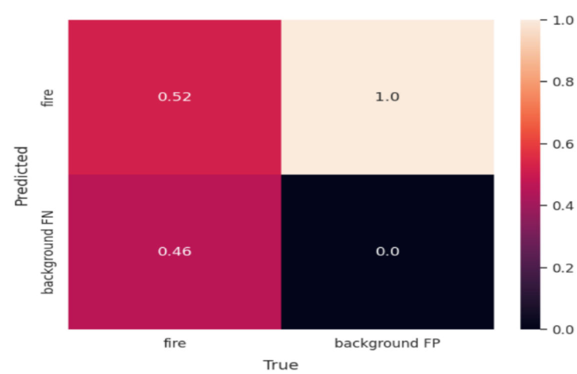

Figure 5 Confusion matrix of model created with only YOLOv5

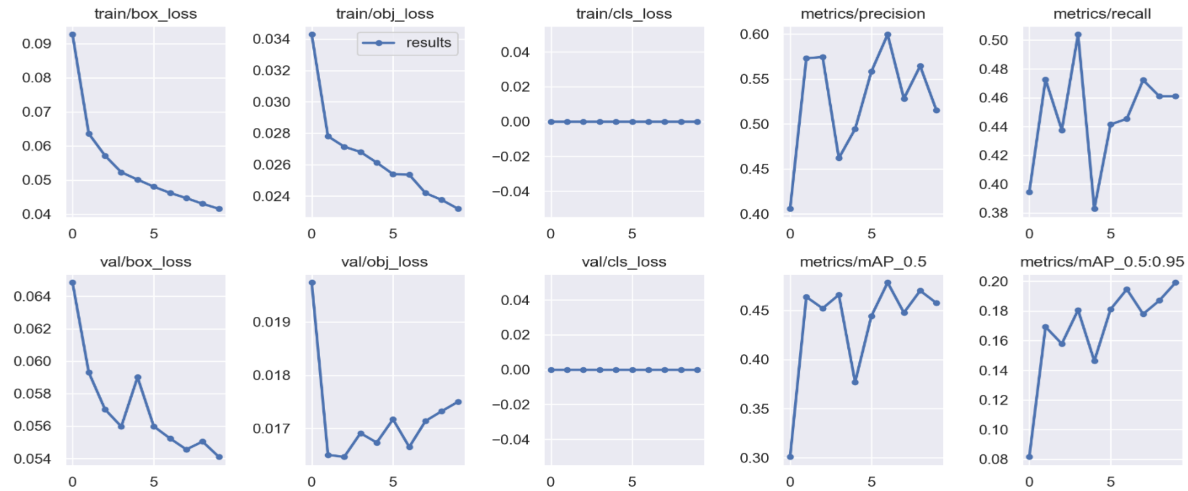

Figure 6 Training and validation metrics of the YOLOv5 model

Table 7 compares the performance of YOLOv5, VGG-16, and the hybrid model based on key metrics: precision, recall, F1-score, and confidence. The results highlight the Hybrid Model's superior accuracy and its balanced performance in terms of detection capability and reliability.

Table 7 Comparison of the performance of YOLOv5, VGG-16, and the hybrid model based on precision, recall, f1-score, and confidence

Model	Precision	Recall	F1-score	Confidence
YOLOv5	0.733	0.320	0.42	0.217
VGG-16	0.678	0.412	0.51	0.300
Hybrid	0.812	0.528	0.61	0.350

4.5Hybrid model with VGG-16

Figure 7 represents a confusion matrix associated with the hybrid model that incorporates the VGG-16 architecture. The matrix provides key performance metrics, showing TP = 1.00, TN = 0.61, FP = 0, and FN = 0.39. This visualization helps in evaluating the classification accuracy and effectiveness of the hybrid model. While the model demonstrates high precision, the confusion matrix suggests that improvements in recall are needed to reduce FN and enhance fire detection reliability. *Figure 8* shows various training and validation metrics for the hybrid model incorporating VGG-16. The training metrics in the top row show a decreasing trend in train/box_loss and train/obj_loss, indicating that the model improves in detecting object locations and identifying objects accurately over training epochs. The train/cls_loss remains constant, suggesting minimal classification errors during training. Additionally, the precision and recall metrics show an upward trend, highlighting the model's increasing ability to correctly classify positive instances while minimizing FNs.

The validation metrics in the bottom row follow a similar pattern, where val/box_loss and val/obj_loss decrease, and reflecting better generalization of the model to unseen data. The val/cls_loss remains stable, indicating consistency in classification performance. Furthermore, the mAP at different IoU thresholds (mAP_0.5 and mAP_0.5:0.95) shows a progressive increase, demonstrating that the hybrid model is effectively learning to make accurate predictions across different confidence levels. Overall, the figure provides an insight into the model's improving performance throughout training.

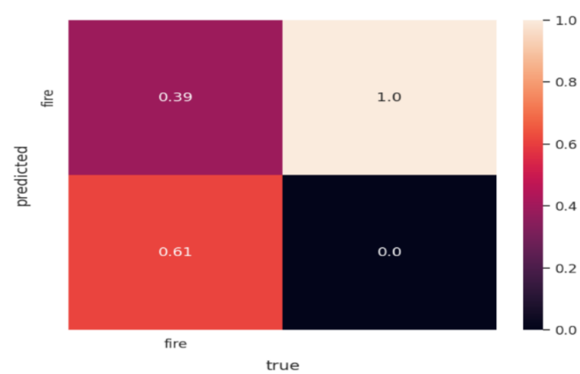

Figure 7 Confusion matrix of the hybrid model incorporatingVGG-16

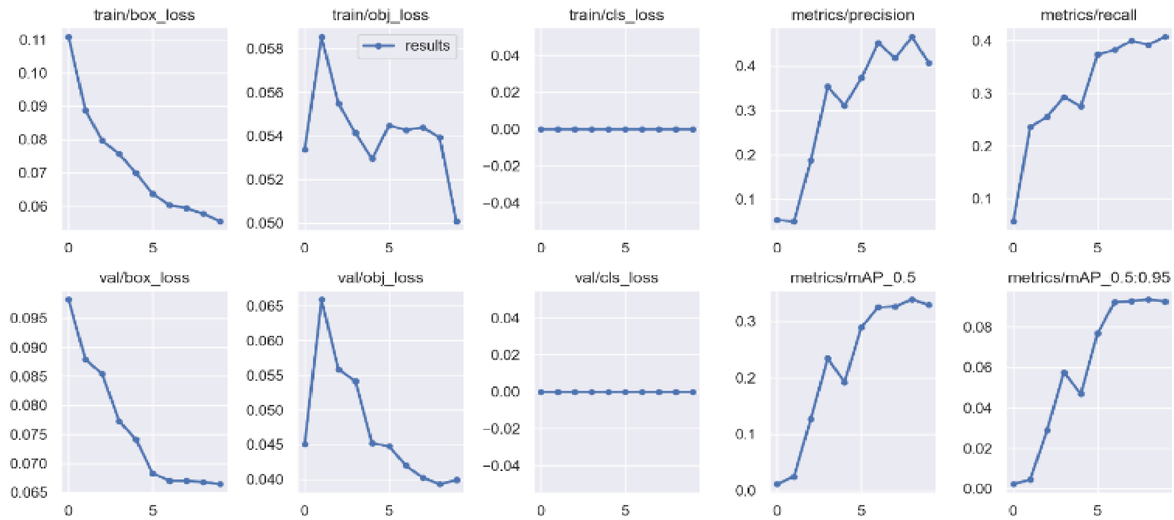

Figure 8 Training and validation metrics of the hybrid model incorporating VGG-16

The VGG-16 model was used to classify the dataset into fire and non-fire images. *Figure 9* illustrates the classification results produced by the proposed VGG-16 model. These results represent the outcome of a classification task performed by a machine learning model based on the VGG-16 architecture.

Figure 9 Fire and non-fire image classification using the VGG-16 model

4.6Error classification

Figure 10 highlights the misclassified images generated by the trained VGG neural network. It depicts instances where the model incorrectly classified fire images as non-fire, providing a visual representation of the model's classification errors.

Figure 10 Misclassification of fire images as non-fire by the VGG-16 model

Figure 11 illustrates the misclassification of non-fire images that were incorrectly labeled as fire by the machine learning model. This visual representation highlights the classification errors where the model falsely identified non-fire instances as fire.

This research explores the complexities of misclassifications in fire detection systems, emphasizing the need for a critical analysis of their underlying causes and potential mitigation strategies. Through visual error analysis using a trained VGG neural network, the study identifies instances where fire images are misclassified as non-fire and vice versa. However, contributing factors such as lighting conditions and image quality remain underexplored. To address these challenges, the study proposes mitigation techniques including advanced data augmentation, refined feature engineering, and adaptive transfer learning methods. The overarching objective is to improve the performance and reliability of fire detection systems across diverse real-world applications by analyzing misclassification dynamics and implementing tailored solutions.

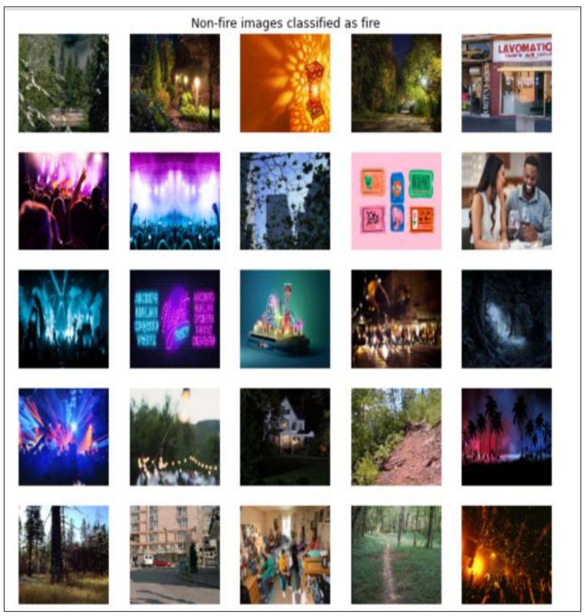

Figure 11 Misclassification of non-fire images as fire by the VGG-16 model

4.7YOLOv5 object detection

Classified images are currently employed to train the YOLOv5 model, aiming to enhance detection accuracy and reduce loss. The YOLOv5 training process was conducted over 10 epochs, with performance metrics such as Precision (P), Recall (R), and mAP tracked throughout to monitor improvements in model accuracy over time.

Figure 12 displays one of the training batches used for training the YOLOv5 model. The bounding boxes shown in the figure were annotated using labeling tools such as Roboflow and Makesense.AI. *Figure 13* presents a sample validation batch detected by the trained YOLOv5 model. The bounding boxes are generated by the model, each accompanied by a label indicating the predicted class. *Figure 14* further illustrates a validation batch detected and predicted by the trained YOLOv5 model. The figure includes bounding boxes with prediction labels and confidence values, offering insight into the model’s confidence in its classifications.

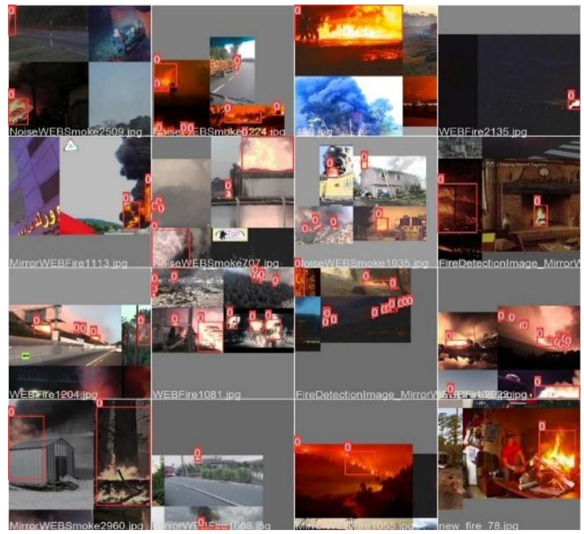

Figure 12 Training batches used for YOLOv5 model training

Figure 15 demonstrates fire detection results on a real-time video using the YOLOv5 model. The model identifies fire in various frames with prediction confidence scores of 0.22, 0.36, and 0.27, respectively. This real-time fire detection example highlights the YOLOv5 model's capability to perform inference efficiently after being trained for more than 10 epochs. During inference, the model typically requires less time compared to training, as it involves only a forward pass to generate predictions on new, unseen data. The actual inference time can vary depending on several factors, including model complexity, input image resolution, batch size, and hardware specifications. In terms of resource requirements, the YOLOv5 model demands computing resources such as CPU, GPU, memory, and storage. These requirements differ based on the model architecture and the extent of optimization applied during training. Larger models with more parameters generally require increased computational power and memory during inference.

Figure 13 Validation batches detected by the trained YOLOv5 model

Figure 14 Predicted validation batches by the trained YOLOv5 model

Figure 15 Fire detection results on real-time video using the YOLOv5 model

5.Discussion

This research focuses on addressing the limitations of current fire detection systems by exploring strategies that enhance model performance and reliability. One key approach involves improving the quality of training data by curating datasets that reflect the unpredictable nature of real-world forest fires. Another effective technique is multi-sensor fusion, which combines data from thermal and visible light cameras to minimize false alarms and offer a more comprehensive understanding of fire incidents. Additionally, the development of lightweight model architectures is essential to ensure efficiency on devices with limited computational resources. While YOLO and VGG architectures have shown promise in forest fire detection, their effectiveness is often constrained by the diversity and quality of available datasets. Recognizing these challenges, this study proposes a hybrid model that leverages the strengths of both VGG-16 and YOLOv5 to improve detection accuracy and reduce errors.

The hybrid system begins by using the VGG-16 image classifier to detect the presence of fire in the input data. The classified output is then passed to the YOLOv5 object detection model, which localizes and labels the fire within the image or video frame. Extensive effort has been made to preprocess and compile a dataset that includes a wide range of fire and non-fire scenarios, capturing the complexity of real-world environments. Experimental results show that the standalone YOLOv5 model achieves 70% accuracy, while the VGG-16 classifier performs with over 90% accuracy. When integrated, the hybrid VGG-16 and YOLOv5 model improves YOLOv5's accuracy to 84%, shortens the training time, and reduces FP by filtering out background noise—even in cases involving smoke. This hybrid approach not only outperforms state-of-the-art methods but also offers practical benefits for applications beyond fire detection, such as surveillance and industrial safety.

Limitation

This study primarily focuses on object recognition and prediction using VGG and YOLO architectures. While YOLO offers high-speed performance and strong detection capabilities, it also presents several limitations in the context of forest fire detection. One major challenge is data dependency—YOLO learns to detect fire and smoke based on the characteristics present in the training data. However, the visual appearance of forest fires can vary widely depending on factors such as fuel type, weather conditions, and the stage of the fire. If the training data lacks sufficient diversity, the model's performance may decline when exposed to real-world scenarios.

Another issue is false alarms. Certain elements in forest environments, such as sunlight reflecting off leaves or controlled burns, can visually resemble smoke or flames. YOLO may misinterpret these patterns as fire, leading to unnecessary alerts. Additionally, dense smoke from large fires can obscure flames, reducing the model's ability to detect fire-related features accurately. In extreme cases, YOLO may fail to detect the fire altogether.

Lastly, while YOLO is optimized for speed compared to many other object detection models, it still incurs a significant computational cost, particularly when deployed on devices with limited processing power. These constraints highlight the importance of careful model training, diverse datasets, and complementary techniques—such as hybrid architectures—to improve robustness and reliability in fire detection systems. A complete list of abbreviations is listed in *Appendix I.*

6.Conclusion and future work

This study presents a comprehensive fire detection system that integrates classification and object detection using a hybrid deep learning model combining VGG-16 and YOLOv5. The system was developed using a meticulously curated dataset of fire and non-fire images collected from multiple sources, ensuring broad representation across real-world scenarios. Extensive preprocessing, including image resizing, normalization, and data augmentation, improved model robustness and generalization. VGG-16 was employed for binary classification, while YOLOv5 performed object localization, enabling real-time detection in complex visual environments. Experimental results demonstrated that the standalone YOLOv5 model achieved an accuracy of 70%, while the VGG-16 model alone reached over 90%. When integrated, the hybrid model improved YOLOv5's accuracy to 84%, reduced FP, and delivered enhanced detection even in smoke-obscured frames. Performance metrics such as precision, recall, F1-score, and mAP confirmed the hybrid model's superiority over individual implementations. Additionally, the study addressed misclassification challenges and proposed targeted mitigation strategies including advanced data augmentation and transfer learning. The system's real-time inference capabilities and adaptability to various lighting conditions make it suitable for deployment in high-risk environments. Overall, the proposed hybrid approach significantly advances the effectiveness and reliability of forest fire detection, with potential applications in surveillance, disaster response, and public safety.

Acknowledgment
None.

Conflicts of interest
The authors have no conflicts of interest to declare.

Data availability
The dataset used in this study, known as the Wildfire Detection Dataset, is publicly accessible and available at https://www.kaggle.com/datasets/brsdincer/wildfire-detection-image-data.

Author's contribution statement
Indira K and Abinaya S: Conceptualization, investigation, writing–original draft, analysis and interpretation, and study. **Aishwarya Shaji:** Editing, analysis and interpretation, study, and supervision.

References

[1] Wu Z, Li M, Wang B, Quan Y, Liu J. Using artificial intelligence to estimate the probability of forest fires in Heilongjiang, northeast China. Remote Sensing. 2021; 13(9):1-18.

[2] Lin X, Li Z, Chen W, Sun X, Gao D. Forest fire prediction based on long-and short-term time-series network. Forests. 2023; 14(4):1-18.

[3] Pechony O, Shindell DT. Driving forces of global wildfires over the past millennium and the forthcoming century. Proceedings of the National Academy of Sciences. 2010; 107(45):19167-70.

[4] Fried JS, Gilless JK, Riley WJ, Moody TJ, Simon DBC, Hayhoe K, et al. Predicting the effect of climate change on wildfire behavior and initial attack success. Climatic Change. 2008; 87:251-64.

[5] Yang S, Huang Q, Yu M. Advancements in remote sensing for active fire detection: a review of datasets and methods. Science of the Total Environment. 2024:173273.

[6] Fried JS, Torn MS, Mills E. The impact of climate change on wildfire severity: a regional forecast for northern California. Climatic Change. 2004; 64(1):169-91.

[7] Nagolu C, Cheekula C, Thota DS, Padmanaban K, Bhattacharyya D. Real-time forest fire detection using IoT and smart sensors. In international conference on inventive computation technologies 2023 (pp. 1441-7). IEEE.

[8] Allison RS, Johnston JM, Craig G, Jennings S. Airborne optical and thermal remote sensing for wildfire detection and monitoring. Sensors. 2016; 16(8):1-29.

[9] Preeti T, Kanakaraddi S, Beelagi A, Malagi S, Sudi A. Forest fire prediction using machine learning techniques. In international conference on intelligent technologies 2021 (pp. 1-6). IEEE.

[10] Abinaya S, Kumar KU, Alphonse AS. Cascading autoencoder with attention residual U-Net for multi-class plant leaf disease segmentation and classification. IEEE Access. 2023; 11:98153-70.

[11] Barmpoutis P, Papaioannou P, Dimitropoulos K, Grammalidis N. A review on early forest fire detection systems using optical remote sensing. Sensors. 2020; 20(22):1-25.

[12] Sakr GE, Elhajj IH, Mitri G. Efficient forest fire occurrence prediction for developing countries using two weather parameters. Engineering Applications of Artificial Intelligence. 2011; 24(5):888-94.

[13] Pradeep GS, Danumah JH, Nikhil S, Prasad MK, Patel N, Mammen PC, et al. Forest fire risk zone mapping of Eravikulam National Park in India: a comparison between frequency ratio and analytic hierarchy process methods. Croatian Journal of Forest Engineering: Journal for Theory and Application of Forestry Engineering. 2022; 43(1):199-217.

[14] Bowman DM, Williamson GJ. River flows are a reliable index of forest fire risk in the temperate tasmanian wilderness world heritage area, Australia. Fire. 2021; 4(2):1-12.

[15] Gülçin D, Deniz B. Remote sensing and GIS-based forest fire risk zone mapping: the case of Manisa, Turkey. Turkish Journal of Forestry. 2020; 21(1):15-24.

[16] Maffei C, Lindenbergh R, Menenti M. Combining multi-spectral and thermal remote sensing to predict forest fire characteristics. ISPRS Journal of Photogrammetry and Remote Sensing. 2021; 181:400-12.

[17] Jin R, Lee KS. Investigation of forest fire characteristics in North Korea using remote sensing data and GIS. Remote Sensing. 2022; 14(22):1-19.

[18] Cunningham AA, Martell DL. A stochastic model for the occurrence of man-caused forest fires. Canadian Journal of Forest Research. 1973; 3(2):282-7.

[19] Tian Y, Wu Z, Li M, Wang B, Zhang X. Forest fire spread monitoring and vegetation dynamics detection based on multi-source remote sensing images. Remote Sensing. 2022; 14(18):1-20.

[20] Shi S, Yao C, Wang S, Han W. A model design for risk assessment of line tripping caused by wildfires. Sensors. 2018; 18(6):1-15.
[21] Kalantar B, Ueda N, Idrees MO, Janizadeh S, Ahmadi K, Shabani F. Forest fire susceptibility prediction based on machine learning models with resampling algorithms on remote sensing data. Remote Sensing. 2020; 12(22):1-24.
[22] Ye Q, Huang P, Zhang Z, Zheng Y, Fu L, Yang W. Multiview learning with robust double-sided twin SVM. IEEE Transactions on Cybernetics. 2021; 52(12):12745-58.
[23] Dampage U, Bandaranayake L, Wanasinghe R, Kottahachchi K, Jayasanka B. Forest fire detection system using wireless sensor networks and machine learning. Scientific Reports. 2022; 12(1):1-11.
[24] Qiu J, Wang H, Shen W, Zhang Y, Su H, Li M. Quantifying forest fire and post-fire vegetation recovery in the Daxin'anling area of northeastern China using landsat time-series data and machine learning. Remote Sensing. 2021; 13(4):1-17.
[25] Xu R, Lin H, Lu K, Cao L, Liu Y. A forest fire detection system based on ensemble learning. Forests. 2021; 12(2):1-16.
[26] Tien BD, Le KT, Nguyen VC, Le HD, Revhaug I. Tropical forest fire susceptibility mapping at the Cat Ba National Park Area, Hai Phong City, Vietnam, using GIS-based kernel logistic regression. Remote Sensing. 2016; 8(4):1-15.
[27] Chang Y, Zhu Z, Bu R, Chen H, Feng Y, Li Y, et al. Predicting fire occurrence patterns with logistic regression in Heilongjiang Province, China. Landscape Ecology. 2013; 28:1989-2004.
[28] Bisquert M, Caselles E, Sánchez JM, Caselles V. Application of artificial neural networks and logistic regression to the prediction of forest fire danger in Galicia using MODIS data. International Journal of Wildland Fire. 2012; 21(8):1025-9.
[29] Oliveira S, Oehler F, San-miguel-ayanz J, Camia A, Pereira JM. Modeling spatial patterns of fire occurrence in mediterranean Europe using multiple regression and random forest. Forest Ecology and Management. 2012; 275:117-29.
[30] Satir O, Berberoglu S, Donmez C. Mapping regional forest fire probability using artificial neural network model in a Mediterranean forest ecosystem. Geomatics, Natural Hazards and Risk. 2016; 7(5):1645-58.
[31] Fan R, Pei M. Lightweight forest fire detection based on deep learning. In 31st international workshop on machine learning for signal processing 2021 (pp. 1-6). IEEE.
[32] Sathishkumar VE, Cho J, Subramanian M, Naren OS. Forest fire and smoke detection using deep learning-based learning without forgetting. Fire Ecology. 2023; 19(1):1-17.
[33] Bui DT, Bui QT, Nguyen QP, Pradhan B, Nampak H, Trinh PT. A hybrid artificial intelligence approach using GIS-based neural-fuzzy inference system and particle swarm optimization for forest fire susceptibility modeling at a tropical area. Agricultural and Forest Meteorology. 2017; 233:32-44.
[34] Kang Y, Jang E, Im J, Kwon C. A deep learning model using geostationary satellite data for forest fire detection with reduced detection latency. GIScience & Remote Sensing. 2022; 59(1):2019-35.
[35] Guo FT, Hu HQ, Ma ZH, Zhang Y. Applicability of different models in simulating the relationships between forest fire occurrence and weather factors in Daxing'an Mountains. The Journal of Applied Ecology. 2010; 21(1):159-64.
[36] Fu L, Zhang D, Ye Q. Recurrent thrifty attention network for remote sensing scene recognition. IEEE Transactions on Geoscience and Remote Sensing. 2020; 59(10):8257-68.
[37] Mohajane M, Costache R, Karimi F, Pham QB, Essahlaoui A, Nguyen H, et al. Application of remote sensing and machine learning algorithms for forest fire mapping in a Mediterranean area. Ecological Indicators. 2021; 129:107869.
[38] Guo F, Zhang L, Jin S, Tigabu M, Su Z, Wang W. Modeling anthropogenic fire occurrence in the boreal forest of China using logistic regression and random forests. Forests. 2016; 7(11):1-14.
[39] Natekar S, Patil S, Nair A, Roychowdhury S. Forest fire prediction using LSTM. In 2nd international conference for emerging technology 2021 (pp. 1-5). IEEE.
[40] Mohan KV, Satish AR, Rao KM, Yarava RK, Babu GC. Leveraging machine learning to predict wild fires. In 2nd international conference on smart electronics and communication 2021 (pp. 1393-400). IEEE.
[41] Jiang K, Chen L, Wang X, An F, Zhang H, Yun T. Simulation on different patterns of mobile laser scanning with extended application on solar beam illumination for forest plot. Forests. 2022; 13(12):1-18.
[42] Fu JJ, Wu ZW, Yan SJ, Zhang YJ, Gu XL, Du LH. Effects of climate, vegetation, and topography on spatial patterns of burn severity in the Great Xing'an Mountains. *Acta Ecologica Sinica*. 2020; 40:1672-82.
[43] Poirier F, Jaziri R, Srour C, Bernard G. Enhancing anomaly detection in videos using a combined YOLO and a VGG GRU approach. In 20th ACS/IEEE international conference on computer systems and applications 2023 (pp. 1-6). IEEE.
[44] Wang Z, Wu L, Li T, Shi P. A smoke detection model based on improved YOLOv5. Mathematics. 2022; 10(7):1-13.
[45] Vijayakumar A, Vairavasundaram S. Yolo-based object detection models: a review and its applications. Multimedia Tools and Applications. 2024; 83(35):83535-74.
[46] Abinaya S, Alphonse AS, Indira K, Karthiga S. Enhanced detection of brain tumour cells using visual geometry group. International Journal of Intelligent Systems and Applications in Engineering. 2022; 10(4):296-300.
[47] https://www.kaggle.com/datasets/brsdincer/wildfire-detection-image-data. Accessed 25 November 2024.

[48] Abinaya S, Rajasenbagam T. Enhanced visual analytics technique for content-based medical image retrieval. International Journal of Electrical and Electronics Research. 2022; 10(2):93-9.
[49] Alphonse AS, Abinaya S, Abirami S. Alibaba and forty thieves algorithm and novel prioritized Prewitt pattern (PPP)-based convolutional neural network (CNN) using hyperspherically compressed weights for facial emotion recognition. Journal of Visual Communication and Image Representation. 2023; 97:103948.
[50] Abinaya S, Indira K, Karthiga S, Rajasenbagam T. Time cluster personalized ranking recommender system in multi-cloud. Mathematics. 2023; 11(6):1-17.
[51] Chen H, Liu H, Sun T, Lou H, Duan X, Bi L, et al. MC-YOLOv5: a multi-class small object detection algorithm. Biomimetics. 2023; 8(4):1-15.

Indira K is currently working as an Assistant Professor at Thiagarajar College of Engineering, with over 13 years of teaching experience. Her research interests include Recommender Systems and Cloud Computing, with several publications in reputed journals and conferences. She is also keenly interested in Mobile Application Development and Cloud-Native Application Development.
Email: kiit@tce.edu

Abinaya S received her B.E., M.E., and Ph.D. degrees in Artificial Intelligence from Anna University. She is currently serving as an Assistant Professor (Senior Grade 2) at Vellore Institute of Technology, Chennai. She has developed several novel information filtering algorithms for recommender systems and pattern recognition techniques in computer vision. She has authored numerous articles in reputed international journals and contributed to books and book chapters focusing on the application of machine learning in agriculture and the Internet of Things (IoT). Her research interests include Artificial Intelligence, Computer Vision, Recommender Systems, Fuzzy Logic, and Optimization Techniques. She is a Lifetime Member of The Institution of Engineers (India) (IEI).
Email: s.abinaya@vit.ac.in

Aishwarya is currently pursuing a B.Tech in Computer Science and Engineering with a specialization in Artificial Intelligence and Machine Learning at Vellore Institute of Technology, Chennai, and is expected to graduate in 2026. Her research interests focus on integrating Machine Learning into real-life applications to enhance efficiency and optimization.
Email: aishwarya.shaji2022@vitstudent.ac.in

Appendix I

S. No.	Abbreviation	Description
1	AI	Artificial Intelligence
2	AVHRR	Advanced Very High-Resolution Radiometer
3	BPNN	Back Propagation Neural Network
4	CPU	Central Processing Unit
5	CIoU	Complete Intersection Over Union
6	CNN	Convolutional Neural Network
7	FN	False Negative
8	FP	False Positive
9	GIoU	Generalized Intersection Over Union
10	GIS	Geographic Information System
11	GPU	General Processing Unit
12	IoT	Internet of Things
13	JPG	Joint Photographic Experts Group
14	LR	Learning Rate
15	LSTM	Long Short-Term Memory
16	mAP	Mean Average Precision
17	MODIS	Moderate Resolution Imaging Spectroradiometer
18	NDVI	Normalized Difference Vegetation Index
19	P	Precision
20	PNG	Portable Network Graphics
21	R	Recall
22	RAM	Random Access Memory
23	R-CNN	Region-based Convolutional Neural Network
24	RNN	Recurrent Neural Networks
25	SSD	Single-Shot Detector
26	SVM	Support Vector Machine
27	TN	True Negative
28	TP	True Positive
29	val_acc	Validation Accuracy
30	val_loss	Validation Loss
31	VGG-16	Visual Geometry Group 16
32	VIIRS	Visible Infrared Imaging Radiometer Suite
33	YOLOv5	You Only Look Once Version 5
34	YOLO	You Only Look Once
35	YOLOv7	You Only Look Once Version 7