# Automatic Fire Detection Algorithm In Roads And Forests Using Convolutional Neural Network

**Amal Alshahrani, Lina Alrefi, Bayan Bajaber, Elaf Haider, Nadia Akber and Ola Halawani**
amshahrani@uqu.edu.sa, s442016671@st.uqu.edu.sa, S442017253@st.uqu.edu.sa, s442016728@st.uqu.edu.sa, s441017872@st.uqu.edu.sa, s442002801@st.uqu.edu.sa

**Abstract**
In our research paper, we propose using image processing techniques alongside convolutional neural networks to detect fires at night on roads and in forests. Fires, which generate thermal and light energy through oxidation, pose significant environmental and personal safety risks. The visual characteristics of flames, such as shape and color, vary based on the combustibles involved. Our study used a Kaggle dataset and additional night-time fire images from the internet. We enhanced these images through preprocessing methods like brightness adjustment and noise reduction to aid our neural networks in recognizing fire features under low-light conditions. By employing transfer learning, we utilized pre-trained models to improve detection accuracy and model generalization across different fire scenarios. Our validation tests confirmed the effectiveness of this approach, demonstrating its potential in early fire detection systems to mitigate risks associated with nocturnal fires in varied environments.

***Keywords:***
*Artificial neural networks, Deep Learning, Machine Learning, Détection*

## 1. Introduction

Fires that occur at night, whether on roads or in forests, present unique challenges for detection and response. The cover of darkness can obscure the early signs of fire, delaying detection until the fire has grown large enough to be visible through the glow of flames. On roads, nighttime fires can result from accidents or mechanical failures in vehicles, posing immediate risks to drivers and potentially leading to larger wildfires if the fire spreads to nearby vegetation. In forests, nocturnal fires can be especially devastating, as they may go unnoticed longer without the routine presence of people who might report them during daylight hours. The remote nature of many forest areas further complicates firefighting efforts, making early detection critical to preventing widespread damage to these vital ecosystems and the wildlife they support [1].

Artificial neural networks, particularly those employing deep learning techniques, are increasingly being utilized in the detection of fires, leveraging the broader field of machine learning for enhanced predictive capabilities. These sophisticated models are trained on vast datasets of fire images, enabling them to learn and recognize the distinct characteristics of fires, such as size, shape, and intensity. Deep learning models, especially convolutional neural networks (CNNs), are adept at processing visual data and can effectively identify fires from video feeds and satellite images. This capability allows for real-time detection and monitoring, significantly reducing response times. Machine learning algorithms can also analyze historical data to predict fire-prone conditions, enabling preemptive measures. Together, these technologies provide a robust framework for early fire detection, potentially saving lives and preserving natural and urban environments by alerting authorities quickly and accurately [2].

This paper investigates the use and evaluation of deep learning models like YoloV8, YoloV9, and VGG16 for fire detection. We explore different architectures, training methods, and optimization tools to enhance detection performance. Our goal is to compare these models comprehensively and implement them in various settings to maximize effectiveness in identifying fires efficiently and accurately.

By utilizing publicly accessible datasets and established evaluation metrics, we set a benchmark for future studies and showcase the transformative potential of artificial intelligence in fire detection. Our research could inform the creation of more precise, accessible, and efficient detection tools, ultimately enhancing safety measures on a global scale.

## 2. Related Work

This literature review explores the evolution of fire detection methodologies, focusing on the integration of

Manuscript received May 5, 2025
Manuscript revised May 20, 2025
https://doi.org/10.22937/IJCSNS.2025.25.5.21

image processing methods and convolutional neural networks (CNNs). The reviewed studies present a diverse range of approaches, each offering unique insights and contributions to the field.

In 2019, Ren et al. [3] conducted a study to present a video-based, image-processing forest fire detection method. It consists of four stages, which are background subtraction, color-based segmentation, special wavelet analysis for color variations, and classification using SVM. The dataset consists of 800 images; 500 are used to train SVM, and 300 are used for testing. The results show an average positive rate of 93.46% and a false positive rate of 6.89%, which indicates the good performance of the used method.

Moreover, in 2019, Barmpoutis et al. [4] proposed a new approach to fire detection based on images by combining deep learning networks and multi-dimensional texture analysis based on linear dynamic systems (LDS). In this research, two datasets were used. The first one is images of forest fires annotated from the Corsican Fire Database (CFDB), which is the largest dataset ever released in this field of research, and the other is images of different objects and categories from the (PASCAL) visual object categories (VOC) dataset. In order to detect candidate fire areas, the researcher tested Faster R-CNN with three different basic networks: Resnet101, VGG16, and AlexNet. He used 500 images from the dataset (CFDB) and divided them into 410 images for training and 90 for testing. In addition, non-fire images from the PASCAL VOC dataset for objects with fiery colors were divided into 200 images for training and 350 images for testing. The first goal of the research was to improve fire detection using images. To achieve this, the researcher compared the recognition rates of the proposed methodology with those achieved by faster R-CNN and spatial texture analysis. (Grassmannian VLAD encoding) when applied separately to fire detection and presents the percentages of images correctly identified as non-fire (true negatives) and false alarms (false positives) among all non-fire images. The other goal was to demonstrate the superiority of the proposed approach compared to other state-of-the-art approches by applying the methodology based on Faster R-CNN with Resnet101, VGG16, and AlexNet architectures. The experimental results of this study show that the proposed approach maintains high positive rates while significantly reducing false positives due to fiery-colored objects.

Furthermore, in 2019, the paper by Alves et al. [5], titled “Automatic Forest fire detection based on a machine learning and image analysis pipeline”, presented a comprehensive approach to forest fire detection using machine learning and image analysis techniques. The paper discusses the challenges associated with forest fire detection, including dynamic environmental phenomena and the diverse landscape of forested areas. The proposed fire detection system consists of several modules, including image acquisition, feature extraction using a Deep Convolutional Neural Network (DCNN), classification using Logistic Regression (LR), and flame area estimation. The use of DCNN, particularly the Inception-V3 model, for feature extraction is highlighted, along with the choice of LR classification due to its superior performance. The authors conducted extensive evaluations to assess the performance of the proposed system. They trained and evaluated the classification models separately for daytime and nighttime scenarios, achieving high accuracy rates of 94.1% and 94.8%, respectively. Additionally, the influence of various image characteristics on the classification process was analyzed using metadata, revealing factors such as fog and artificial lighting to be significant contributors to false positives. Furthermore, the paper discusses the estimation of flame areas using the CAFE approach, which integrates the FFDI color index with preprocessing techniques in the lab color space. The evaluation of flame area estimation showed promising results, with a reduction in false positives compared to using FFDI alone.

In Addition, in 2020, Valikhujaev et al. [6], proposed a new method based on a deep learning approach, utilizing a convolutional neural network that incorporates dilated convolutions, which include the following four main features: the utilization of dilation filters, a limited number of layers, small kernel sizes, and a custom-built dataset. They curated a diverse dataset by extracting frames from fire and smoke videos and gathering images from internet sources, amounting to 16,860 images. The utilization of a dilation operator and a limited number of layers can enhance the method's performance by extracting valuable features. However, the method predominantly encounters issues, especially in cloudy weather conditions.

The study [7] presents a novel automatic early warning system designed to detect forest fires, utilizing a combination of sensors, the Raspberry Pi 3, a neural wand, APM 2.5, GPS, Wi-Fi, and advanced deep learning algorithms, including YOLO. With drones equipped, the neural wand performs real-time image processing. The system architecture primarily focuses on identifying potential flame areas, employing an optical flow strategy to classify moving objects within these regions, and analyzing movement vectors to confirm fire presence. The Yolov3 model integrated into the neural stick aids in this process. Upon capturing an image, it undergoes processing and is fed into a CNN for fire verification. The research concludes that this YOLO-based forest fire detection system operates with 90% accuracy, with sensitivity and specificity rates of 92% and 90%, respectively.

Likewise In 2021, Ya’acob et al. [8], presented an application of image processing techniques to analyze the captured images and distinguish fire pixels from the background. The RGB and YCbCr color models are used for pixel isolation. The application of MATLAB for image processing underscores the potential of combining infrared imaging with advanced computational methods for accurate fire detection. The results showed the method used is comprehensive and effective for forest fire detection.

Furthermore, in 2021, Taspinar et al. [9], employed advanced image processing algorithms to accurately identify fires. Additionally, CNN methods were utilized for fire detection in images, with models trained via transfer learning using Inception V3, SqueezeNet, VGG16, and VGG19. The dataset used in this study was comprehensive, consisting of 3041 images sourced from search engines and amalgamating datasets from prior research efforts. It comprised 1900 natural images devoid of fires, with the remainder depicting various fire scenarios. The study introduced a three-stage fire framework. In the initial stage, the researcher applied several image processing techniques for flame detection, including brightness reduction, HSL (Hue, Saturation, Luminance), YCbCr (Y: Luminance, Cb: Chroma (Blue Minus Luma), Cr: Chroma (Red) Minus Light), as well as median, grass, and edge detection filters. The second stage involved leveraging flame movement features, and detecting pixel movements by comparing consecutive frames. Finally, in the third stage, the presence of fire across the entire image was determined using CNN algorithms. Model tests yielded classification success rates of 97.3%, 97.0%, 98.8%, and 96.8%.

As well In 2021, a study written by Ryu et al. [10] discussed a way to reduce the incidence of false detections by using HSV color conversion and Harris Corner detection in the image pre-processing step. Furthermore, from the identified corners, the area surrounding the corner point directed upwards was isolated as a Region of Interest (ROI). Fire presence was then discerned utilizing a Convolutional Neural Network (CNN). These methodologies were tailored to recognize flame occurrences by leveraging their upward-pointing attributes, yielding superior accuracy and precision compared to traditional object detection algorithms reliant solely on static image inputs. Consequently, this approach significantly mitigated false detections of non-fire elements, thereby enhancing the precision of fire detection, and the accuracy increased compared to Faster R-CNN from 89% to 97.5%.

In the same vein, in 2022, the study conducted by K. Mohammed. Titled, "A real-time forest fire and smoke detection system using deep learning" [11], addressed the critical need for early detection systems to combat the increasing threat of forest fires globally. The paper highlights the challenges associated with traditional detection methods and proposes a novel approach using deep learning, specifically transfer learning with the Inception-ResNet-v2 model, for feature extraction and classification. The system's architecture involves real-time image processing from surveillance cameras or drones, where the trained model predicts the probability of fire or smoke. Through rigorous experimentation, the proposed system achieved impressive performance metrics, including an accuracy of 99.09%, precision of 100%, and sensitivity of 98.08%.

Additionally, Ahmed A. Alsheikhy [12], proposed a method in 2022 based on image processing techniques and a Convolutional Neural Network (CNN) to detect fires early using a deep learning approach. It used the AlexNet tool which is a type of convolutional neural network that utilizes the deep learning method, and used MATLAB as a simulation tool for the proposed method.
The deep learning approach filters an image into pixels based on thresholds according to features such as colors, immobility source, and flame texture with its reflection. the dataset was sourced from Kaggle and categorized into three distinct classes. The results obtained from this method showcased an impressive accuracy rate exceeding 97.73% when trained and tested on a dataset consisting of over 700 images.

Lastly, in 2023, Abdusalomov et al. [13] improved the forest fire detection method to classify fires based on a new version of the Detectron2 platform using deep learning approaches. They collect a large custom dataset with two classes, fire and non-fire, with different scenarios (day and night) of fire and flame, light, and shadows, totaling 348,600 images. Mask_rccn_50_FPN_3x had a high training accuracy of 98.3% and a testing accuracy of 97.8%, followed by Keypoint_rcnn_R_50_FPN_3x, with 96.1% training accuracy and 95.3% testing accuracy, with a difference of less than 2%. Panoptic_fpn_R_101_3x also improved the training and testing accuracy by 88.3% and 85.1%, respectively. The proposed model has some limitations; for example, electric light or the sun was considered fire in some cases when we tested the model in different environments. They did not create any classes for smoke in the custom dataset.

The study [14] introduces an enhanced method for detecting fires in smart cities, termed the Intelligent Fire Detection System, built upon the YOLOv8 algorithm. This approach overcomes previous limitations by offering heightened accuracy, real-time detection, adaptability, reduced false alarms, and cost-effectiveness. The SFDS methodology adopts the YOLOv8 detection model, optimized for efficient object detection without a regional proposal network, and involves processes such as gathering data for fire and non-fire images. Utilizing a dataset of 26,520 images featuring diverse fire and smoke scenarios, indoor and outdoor settings, various lighting conditions, and normal scenes, the research demonstrates superior performance compared to existing fire detection systems. Achieving an accuracy rate of 95.7% for fire detection, 99.3% for smoke detection, and an overall average accuracy of 97.5%, the proposed approach excels in recall, accuracy, and F1 score metrics.

In this research [15], the author presented the DCGC-YOLO fire detection algorithm, a modification of the YOLOv5 model. This model introduces a novel layer structure and linking algorithm to enhance the original YOLOv5. Initially, a cross-stage partial architecture (CSP) is introduced, incorporating large convolutional kernels in the bottleneck layer to expand the network's receptive field and improve feature extraction capabilities. Moreover, an intersection-over-union (IoU)-based anchor generation algorithm is employed to adjust anchor sizes in a custom fire dataset, augmenting model robustness and detection accuracy. Experimental results on the fire dataset demonstrate that the proposed DCGC-YOLO algorithm achieves effective target detection with a mean average precision (mAP) of 41.1%, surpassing YOLOv5s by 2.9%, while also reducing network parameters and computational complexity. Additionally, experiments conducted on the COCO2017 dataset validate the algorithm's effectiveness, showcasing a mAP of 38.9%, indicating strong generalization and competitive performance compared to contemporary detectors.

Insufficient data presents a notable obstacle to fire detection at night research, affecting the progress and efficacy of fire detection systems. Enhanced accuracy in fire detection has been observed with the utilization of extensive datasets compared to smaller ones. Rectifying the issue of data scarcity not only boosts the precision and dependability of fire detection systems but also advances fire safety and disaster prevention efforts. To tackle this challenge, we opted to employ an extensive database for model training.

## 3. Methodology

In the methodology section, we outline our approach to evaluating various deep learning models for fire detection. We describe the selection and preparation of datasets, the choice of model architectures, and the evaluation metrics used. This section details the systematic process we followed to assess and compare the effectiveness of each model, ensuring our results are both reliable and replicable.

### 3.1 Dataset

Our study leveraged two distinct datasets, each containing 8,000 images, for our research. The first dataset, obtained from Roboflow, was used to train the Yolov8 and Yolov9 models. The second dataset, sourced from Kaggle, facilitated the training of the VGG-16 model. We enhanced these datasets with additional nighttime fire images to tailor the training specifically for fire detection scenarios under low-light conditions. We structured the dataset division with 70% allocated for training to help the models learn and identify patterns effectively, while the remaining 30% was used for testing and validation. This division enabled thorough analysis and meaningful evaluation of the models' performance.

### 3.2 Detection Using Yolov8

We incorporate the use of YoloV8, a state-of-the-art object detection model, to identify fires in various environments. YoloV8 is an evolution of the YOLO (You Only Look Once) series, renowned for its real-time object detection capabilities. This model is particularly effective due to its deep convolutional neural networks that analyze images in a single pass, making it highly efficient for scenarios where quick detection is critical, such as in fire surveillance and safety systems.

YoloV8 operates on the principle of dividing the image into a grid and predicting bounding boxes and probabilities for each grid cell. The model applies a series of convolutional layers to extract features from the image, followed by prediction layers that determine the presence of objects within the grid cells. The bounding boxes are defined by coordinates relative to each cell, and each box is associated with a confidence score that indicates the likelihood of detecting a specific object, in this case, fire.

The primary equation that governs YoloV8's detection process involves calculating the Intersection over Union (IoU) as part of the loss function during training. The IoU is calculated as follows:

$$IoU = \frac{Area\ of\ Overlap\ between\ the\ predicted\ box\ and\ ground\ truth\ box}{Area\ of\ Overlap\ between\ the\ predicted\ box\ and\ ground\ truth\ box}$$

[16]

### 3.3 Detection Using Yolov9

We also employ YoloV9, a successor to YoloV8, incorporating advanced features to enhance fire detection capabilities. YoloV9 introduces several architectural improvements that optimize both computational efficiency and detection accuracy. These include deeper layers and an enhanced feature extraction network, which are crucial for capturing detailed aspects of complex images under varied conditions.

YoloV9 also benefits from improved training techniques that accelerate the learning process without compromising accuracy. The model uses advanced data augmentation methods, like mosaic augmentation and self-adversarial training, to increase robustness against diverse fire scenarios, especially those challenged by low-light conditions.

Furthermore, YoloV9 has refined its approach to bounding box regression and class prediction, achieving higher precision in detecting and classifying objects. This improvement is vital for accurately identifying fires and distinguishing them from other luminous sources in nighttime images. Figure 1 below shows the architecture diagram for YOLO [17].

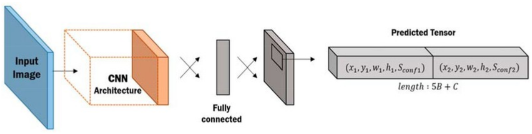


Fig.1 YOLO Network Architecture Diagram[18]

## 3.4 Detection Using VGG-16

We employ the VGG-16 model, which is widely recognized for its robust performance in image recognition tasks. Developed by the Visual Graphics Group at Oxford, VGG-16 features a deep architecture with 16 layers that have learnable weights, including 13 convolutional layers and 3 fully connected layers. This model is known for its use of small (3x3) convolutional filters, which are particularly effective in capturing detailed image features. Such granularity is crucial for identifying subtle and critical characteristics in fire images, such as edges and textures.

VGG-16's application in fire detection leverages its strength in handling complex image classification tasks. We adapted a pre-trained VGG-16 model, initially trained on the extensive ImageNet dataset, to the specific requirements of fire detection. This approach allows us to utilize the model's already rich feature representations, refining them to recognize various fire types and their developmental stages.

For training VGG-16 in our fire detection context, we employ the categorical cross-entropy loss function, defined mathematically as:

$$L = \sum_{c=1}^{M} y_{o,c} \log(P_{o,c})$$

*Here:*

- *$L$ is the loss for one observation.*
- *$M$ represents the number of classes.*
- *$y_{o,c}$ is a binary indicator (0 or 1) if class label $c$ is the correct classification for observation $o$.*
- *$P_{o,c}$ is the predicted probability that observation $o$ belongs to class $c$.*

This loss function calculates the loss by summing the negative log of the predicted probabilities assigned to the true class labels across all classes. It effectively penalizes deviations from the actual labels, enhancing the model's accuracy in classifying and detecting fires.

Through this methodology, using VGG-16 enriches our detection strategy with deep learning insights, providing a reliable tool for accurate and efficient fire detection [19] [20].

Figure 2 below shows the architecture diagram for the VGG-16 model.

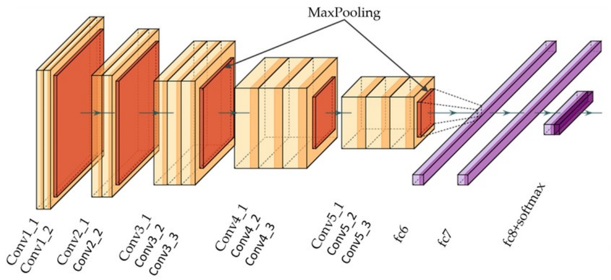


Fig.2 Architecture Diagram for VGG-16 model.[21]

After introducing YoloV8, YoloV9, and VGG-16, our study will compare these models to determine which is most effective for nighttime fire detection in forests and on roads. We will evaluate their performance based on accuracy under low-light conditions. This comparison will help us identify the optimal model for reliable and efficient fire detection, guiding future enhancements in this technology.

## 4. Implementation

In the implementation phase of our study, we utilized Google Colab, a cloud-based platform that provides a conducive environment for running high-performance models without requiring local computational resources. For each model YoloV8, YoloV9, and VGG-16, we imported the necessary libraries and frameworks essential for deploying and testing these advanced deep learning models. Additionally, we conducted training over three different epochs for each model to evaluate their learning curves and stability across iterations. Using Google Colab allowed us to take advantage of its powerful GPUs, facilitating efficient training and evaluation processes. This setup not only streamlined our experimental procedures but also ensured that we could rigorously test each model under consistent conditions to accurately assess their performance in detecting nighttime fires.

## 4.1 Training Parameters

Hyperparameters are crucial settings of an algorithm, established before the learning process begins. They define the model's structure and influence the learning dynamics. Unlike model parameters, which are learned during training, hyperparameters such as the learning rate, number of epochs, and batch size must be manually set to optimize performance. The learning rate adjusts the model in response to error, the number of epochs determines the training duration, and the batch size affects the update frequency and overall learning efficiency.

In Tables 1 and 2, the hyperparameters for YOLOv8 and YOLOv9 include a large input image size of 4135, which emphasizes the need for robust computational capabilities. The training process uses 40 epochs and a batch size of 16, balancing efficiency and memory use. The SGDP optimizer, combined with an initial learning rate of 0.01 and a final rate of 0.001, supports rapid yet stable convergence. A high momentum value of 0.994 ensures quick adjustments in the learning direction, while a moderate weight decay of 0.0063 helps prevent overfitting, maintaining the generalization of the model.

**Table 1**: Hyperparameters for YOLOv8

| Hyperparameter | Value for YOLOv8 |
|---|---|
| **Input of image size** | 4135 |
| **Epochs** | 40 |
| **Batch size** | 16 |
| **Optimizer** | SGDP |
| **Initial learning rate** | 0.01 |
| **Final learning rate** | 0.001 |
| **Momentum** | 0.994 |
| **Weight decay** | 0.0063 |

**Table 2**: Hyperparameters for YOLOv9

| Hyperparameter | Value for YOLOv8 |
|---|---|
| **Input of image size** | 4135 |
| **Epochs** | 40 |
| **Batch size** | 16 |
| **Optimizer** | SGDP |
| **Initial learning rate** | 0.01 |
| **Final learning rate** | 0.001 |
| **Momentum** | 0.996 |
| **Weight decay** | 0.0063 |

In comparing the hyperparameters between YOLOv8 and YOLOv9, the main difference lies in the momentum value used during training. While YOLOv8 uses a momentum of 0.994, YOLOv9 slightly increases this to 0.996. All other parameters, including the input image size of 4135, the number of epochs at 40, a batch size of 16, the use of the SGDP optimizer, an initial learning rate of 0.01, a final learning rate of 0.001, and a weight decay of 0.0063, remain consistent across both versions. This subtle change in momentum could suggest an attempt to enhance convergence speed or stability in the newer YOLOv9 model.

# 5. Results

In the results phase of our study, we meticulously analyzed and documented the outcomes of the comparative evaluation between YoloV8, YoloV9, and VGG-16. This section details the performance metrics of each model, such as accuracy, precision, recall, and detection speed, derived from the tests conducted over three different epochs. The results are presented to highlight the capabilities and limitations of each model in the context of nighttime fire detection in both forested areas and roads. By interpreting these findings, we aim to provide a clear understanding of which model demonstrates superior effectiveness under the specified conditions, offering insights that could be pivotal for future developments in fire detection technology.

## 5.1 Comparison Experiments between YOLO-8 and YOLO-9

It can be observed from Table 3. The experimental results demonstrate that YOLO-8 achieved a validation accuracy of 99.4% and a test accuracy of 98% in fire detection tasks. In comparison, YOLO-9 attained a validation accuracy of 98.3% and a test accuracy of 90%. These results indicate that YOLO-8 outperforms YOLO-9 in terms of overall accuracy, particularly in the test phase.

The higher accuracy of YOLO-8 in both validation and test sets suggests its robustness in detecting fires in road and forest environments. Conversely, YOLO-9 exhibits slightly lower accuracy, especially in the test phase, indicating potential challenges in real-world deployment, particularly in forest settings.

**Table 3**: Result of Yolov8 and Yolov9

| Model | Validation Accuracy (%) | Test Accuracy (%) |
|---|---|---|
| YOLO-8 | 99.4% | 98% |
| YOLO-9 | 98.3% | 90% |

Figure 3 below displays the results of **Yolov8** in different metrics using **25** epochs.

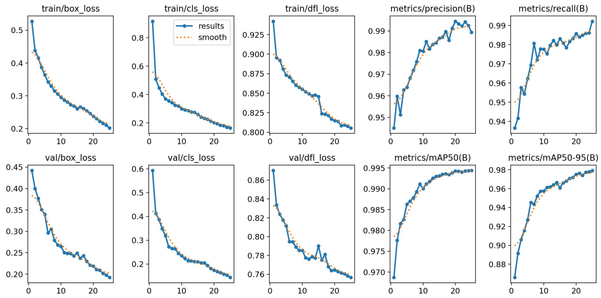


**Fig 3.** Results of Yolov8 in different metrics (25 epochs)

Figure 4 presents the results of YOLOv8's fire detection on a testing image after 25 epochs of training.


**Fig 4.** The detection of fire on a testing image utilizing YOLOv8 after 25 epochs.

Figure 5 below displays the results of **Yolov8** in different metrics using **40** epochs.

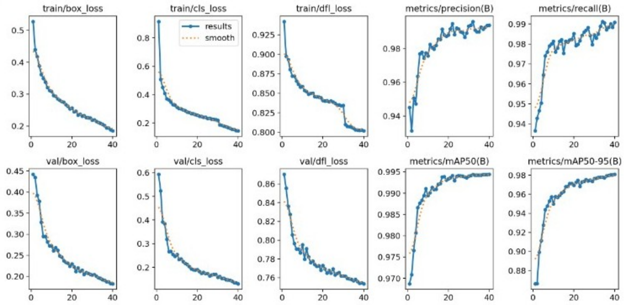


**Fig 5.** results of Yolov8 in different metrics (40 epochs)

Figure 6 presents the results of YOLOv8's fire detection on a testing image after 40 epochs of training.


**Fig 6.** The detection of fire on a testing image utilizing YOLOv8 after 40 epochs.

Figure 7 Below displays the results of **Yolov8** in different metrics using **60** epochs.

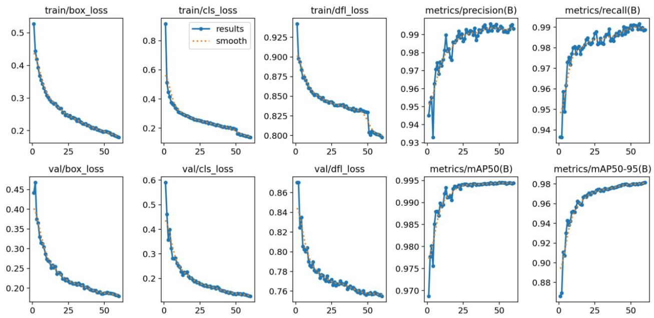


Fig 7. Results of Yolov8 in different metrics (60 epochs)

Figure 8 presents the results of YOLOv8's fire detection on a testing image after 60 epochs of training.


**Fig 8**.The detection of fire on a testing image utilizing YOLOv8 after 60 epochs.

Figure (9) below displays the results of **Yolov9** in different metrics using 40 epochs.

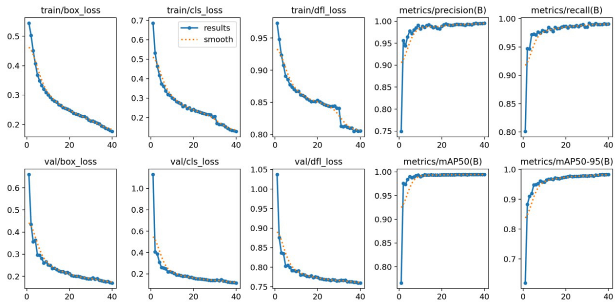


Fig 9. Results of Yolov8 in different metrics.

Figure 10 showcases YOLOv9's ability to accurately detect fire across a series of testing images.

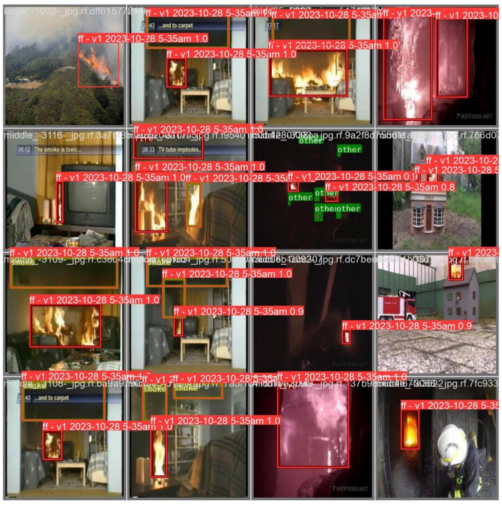

**Fig 10**.the detection of fire on a testing image utilizing YOLOv9.

## 5.2 Comparison Experiments between YOLO and VGG-16

The experimental findings indicate that YOLOv8 achieved an impressive average accuracy of 99% in fire detection tasks. YOLOv9 demonstrated slightly lower but still commendable accuracy, securing 98.3%. In comparison, VGG-16 achieved an accuracy of 91.3%. These results underscore the effectiveness of YOLO in accurately detecting fires in both road and forest environments.

The superior performance of YOLOv8 and YOLOv9 compared to VGG-16 highlights the benefits of object detection frameworks for fire detection applications. The robustness of YOLO, particularly in complex scenarios such as forest environments, can be attributed to its ability to efficiently localize and classify objects. While VGG-16 achieves respectable accuracy, its performance falls short of YOLO's. Figure 11 below displays the results of the accuracy rate for the detection algorithms.

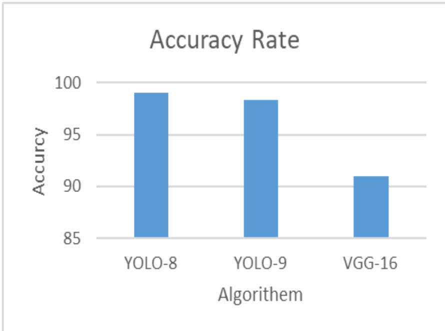


**Fig 11**. results of accuracy rate for the algorithms.

**Table 4**: Result of Yolov8 and Yolov9 and VGG17

| | Number of epochs | Accuracy |
|---|---|---|
| YOLOv8 | 25 | 90% |
| | 40 | 98% |
| | 60 | 94% |
| | Average | 94% |
| YOLOv9 | 40 | 98.3% |
| VGG16 | 5 | 91.3% |

As shown in Table 4, the comparison of the YOLOv8, YOLOv9, and VGG16 models reveals distinct performance trends. YOLOv9 stands out with the highest accuracy of 98.3% achieved after 40 epochs, surpassing both YOLOv8 and VGG16. While YOLOv8 also exhibits strong performance, reaching 98% accuracy with 40 epochs and averaging at 94%, VGG16 trails behind with an accuracy of 91.3% after only 5 epochs. These results underline the advancements in object detection models, with YOLO architectures generally outperforming the traditional VGG16 in accuracy, albeit potentially requiring more computational resources.

### 5.3 Comparison Experiments between YOLO and VGG-16 using confusion matrix

The confusion matrices for YOLOv8, YOLOv9, and VGG16 that are displayed in Figures 12, 13 and 14 reveal varied performances across these models. YOLOv9 demonstrates the highest accuracy with minimal misclassification, as indicated by almost perfect scores in identifying true positives and negatives. YOLOv8, while also showing high true positive rates, has slightly higher false positives and negatives compared to YOLOv9, suggesting areas for improvement in specificity and sensitivity. VGG16, on the other hand, has higher rates of both false positives and false negatives, indicating it may require more substantial adjustments to match the precision seen in the YOLO models. Overall, YOLOv9 stands out for its robust accuracy in object detection, making it a superior choice for applications requiring high reliability.

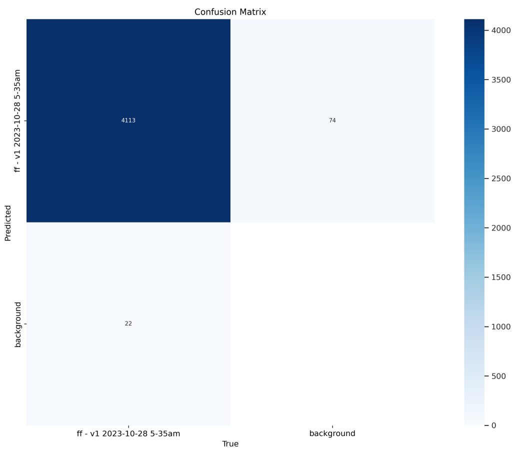


**Fig12:** Confusion matrix for YOLOv8

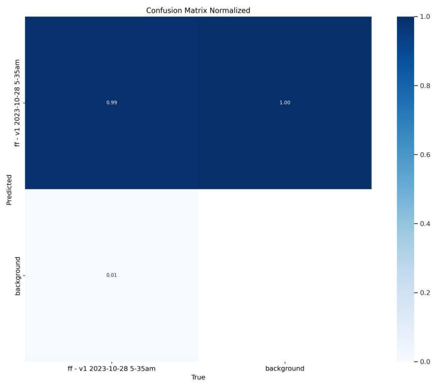


**Fig13**: Confusion matrix for YOLOv9

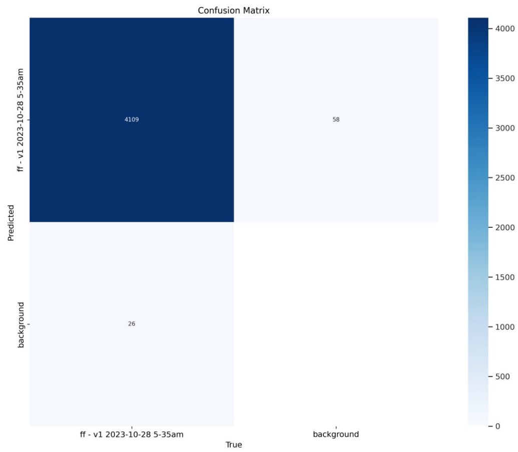


Fig14: Confusion matrix for VGG16

## 6. Findings

The findings highlight the advantages of utilizing advanced object detection frameworks like YOLO for critical applications such as fire detection. The ability of YOLO models to efficiently localize and classify objects in complex and dynamic backgrounds makes them particularly useful in emergency response scenarios where accuracy and speed are paramount. Despite VGG-16's respectable performance, it falls short of the high standards set by YOLO models in our testing scenarios.

## 7. Limitations

Despite the promising outcomes of this study, several limitations must be acknowledged. The performance of the models, particularly in complex and variable environments like forests at night, may face challenges not captured during controlled testing. Additionally, the computational demand of the YOLO models, while efficient, requires significant processing power that might not be available in all practical applications. Furthermore, the study did not explore the impact of adverse weather conditions on detection accuracy, which is a crucial factor for real-world deployment. Addressing these limitations in future research will be essential to advancing the reliability and applicability of fire detection systems.

## 8. Future Work

Looking ahead, several avenues for future work can build on the findings of this study to enhance fire detection technologies. One critical area is the integration of models with real-time monitoring systems, such as drones or CCTV networks, which could provide dynamic data to improve the models' adaptability and accuracy in varied environments. Additionally, further research could explore the implementation of lighter, more efficient versions of the YOLO models to reduce computational demands and enable deployment on less powerful devices.

Expanding the dataset to include a broader range of fire scenarios, including those affected by different weather conditions, would also be beneficial. This expansion could help to develop models that are more robust against environmental variations. Moreover, incorporating multi-spectral imaging data, such as infrared or thermal imagery, might improve detection capabilities, particularly for hidden or smoldering fires that are difficult to detect with standard visual inputs.

Finally, collaborative efforts between AI researchers, fire safety experts, and emergency response teams could lead to innovations in model training and real-world application. Such collaborations could ensure that the developed models not only meet technical specifications but also align with practical firefighting needs and strategies, ultimately leading to more effective and reliable fire detection systems.

## 9. Conclusion

In conclusion, this study thoroughly assessed the performance of YOLO-8, YOLO-9, and VGG-16 in detecting fires under nighttime conditions in road and forest settings. Our results clearly show that YOLO-8 outperforms the others, demonstrating remarkable accuracy with 99.4% in validation and 98% in testing, thus confirming its robustness and reliability for practical applications. YOLO-9, while slightly less accurate in testing, especially in forest environments, still shows commendable performance. In contrast, VGG-16, though it achieves respectable accuracy, does not match the superior performance of the YOLO models.

This study highlights the advantages of using advanced object detection frameworks like YOLO for critical tasks such as fire detection, where the ability to accurately and quickly localize and classify objects can be lifesaving. The insights gained from this comparative analysis reinforce the potential of YOLO models in enhancing emergency response strategies and suggest further research avenues for optimizing these models for even greater effectiveness in diverse and challenging environments.

## Acknowledgment

We would like to express our sincere gratitude to everyone who contributed to the completion of this research paper. First and foremost, we extend our deep appreciation to Amal Al-Shahrani for her invaluable guidance, unwavering support, and insightful feedback throughout the entire process.

We also thank the participants in this study for their time, cooperation, and willingness to provide every effort so that we could complete this paper, without which this research would not have been possible.

Additionally, our heartfelt thanks go to Umm al-Qura University for their generous support and the resources they provided, which were instrumental in facilitating our research. The university's commitment to fostering an environment conducive to scientific inquiry and innovation has been crucial to the success of our work.

## 10. References

[1] A. A. A. Alkhatib, ‘A Review on Forest Fire Detection Techniques’, *International Journal of Distributed Sensor Networks*, vol. 10, no. 3, p. 597368, Mar. 2014, doi: 10.1155/2014/597368.

[2] X. Wang, S. Lo, and H. Zhang, ‘Influence of feature extraction duration and step size on ANN based multisensor fire detection performance’, *Procedia Engineering*, vol. 52, pp. 413–421, 2013.

[3] Y. Ren, D. Liu, Q. Xiong, J. Fu, and L. Wang, ‘Spec-ResNet: A General Audio Steganalysis scheme based on Deep Residual Network of Spectrogram’. arXiv, Feb. 26, 2019. doi: 10.48550/arXiv.1901.06838.

[4] P. Barmpoutis, K. Dimitropoulos, K. Kaza, and N. Grammalidis, ‘Fire detection from images using faster R-CNN and multidimensional texture analysis’, in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2019, pp. 8301–8305. Accessed: Apr. 06, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8682647/

[5] J. Alves, C. Soares, J. M. Torres, P. Sobral, and R. S. Moreira, ‘Automatic forest fire detection based on a machine learning and image analysis pipeline’, in *New Knowledge in Information Systems and Technologies: Volume 2*, Springer, 2019, pp. 240–251. Accessed: Apr. 06, 2024. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-030-16184-2_24

[6] Y. Valikhujaev, A. Abdusalomov, and Y. I. Cho, ‘Automatic fire and smoke detection method for surveillance systems based on dilated CNNs’, *Atmosphere*, vol. 11, no. 11, p. 1241, 2020.

[7] S. Goyal, M. Shagill, A. Kaur, H. Vohra, and A. Singh, ‘A yolo based technique for early forest fire detection’, *Int. J. Innov. Technol. Explor. Eng*, vol. 9, pp. 1357–1362, 2020.

[8] N. Ya’acob, M. S. M. Najib, N. Tajudin, A. L. Yusof, and M. Kassim, ‘Image processing based forest fire detection using infrared camera’, in *Journal of Physics: Conference Series*, IOP Publishing, 2021, p. 012014. Accessed: Apr. 06, 2024. [Online]. Available: https://iopscience.iop.org/article/10.1088/1742-6596/1768/1/012014/meta

[9] Y. S. Taspinar, M. Koklu, and M. Altin, ‘Fire detection in images using framework based on image processing, motion detection and convolutional neural network’, *International Journal of Intelligent Systems and Applications in Engineering*, vol. 9, no. 4, pp. 171–177, 2021.

[10] J. Ryu and D. Kwak, ‘Flame Detection Using Appearance-Based Pre-Processing and Convolutional Neural Network’, *Applied Sciences*, vol. 11, no. 11, Art. no. 11, Jan. 2021, doi: 10.3390/app11115138.

[11] R. K Mohammed, ‘A real-time forest fire and smoke detection system using deep learning’, *International Journal of Nonlinear Analysis and Applications*, vol. 13, no. 1, pp. 2053–2063, 2022.

[12] A. A. Alsheikhy, ‘A fire detection algorithm using convolutional neural network’, *J. King Abdulaziz Univ. Eng. Sci.*, vol. 32, no. 2, 2022, Accessed: Mar. 26, 2024. [Online]. Available: https://www.academia.edu/download/105047522/75668_48841.pdf

[13] A. B. Abdusalomov, B. M. S. Islam, R. Nasimov, M. Mukhiddinov, and T. K. Whangbo, ‘An improved forest fire detection method based on the detectron2 model and a deep learning approach’, *Sensors*, vol. 23, no. 3, p. 1512, 2023.

[14] F. M. Talaat and H. ZainEldin, ‘An improved fire detection approach based on YOLO-v8 for smart cities’, *Neural Comput & Applic*, vol. 35, no. 28, pp. 20939–20954, Oct. 2023, doi: 10.1007/s00521-023-08809-1.

[15] Y. He, J. Hu, M. Zeng, Y. Qian, and R. Zhang, ‘DCGC-YOLO: The Efficient Dual-Channel Bottleneck Structure YOLO Detection Algorithm for Fire Detection’, *IEEE Access*, 2024, Accessed: May 14, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10494335/

[16] M. Hussain, ‘YOLO-v1 to YOLO-v8, the Rise of YOLO and Its Complementary Nature toward Digital Manufacturing and Industrial Defect Detection’, *Machines*, vol. 11, no. 7, Art. no. 7, Jul. 2023, doi: 10.3390/machines11070677.

[17] J. Redmon and A. Farhadi, ‘YOLO9000: better, faster, stronger’, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7263–7271. Accessed: May 14, 2024. [Online]. Available: http://openaccess.thecvf.com/content_cvpr_2017/html/Redmon_YOLO9000_Better_Faster_CVPR_2017_paper.html

[18] ‘YOLO network architecture diagram. | Download Scientific Diagram’. Accessed: May 14, 2024. [Online]. Available: https://www.researchgate.net/figure/YOLO-network-architecture-diagram_fig1_362639148

[19] ‘very deep convolutional networks for large-scale recognition - الباحث العلمي من Google’. Accessed: May 14, 2024. [Online]. Available: https://scholar.google.com/scholar?hl=ar&as_sdt=0%2C5&q=very+deep+convolutional+networks+for+large-scale+recognition&btnG=

[20] H. Qassim, A. Verma, and D. Feinzimer, ‘Compressed residual-VGG16 CNN model for big data places image recognition’, in *2018 IEEE 8th annual computing and communication workshop and conference (CCWC)*, IEEE, 2018, pp. 169–175. Accessed: May 14, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8301729/

[21] ‘The architecture of the VGG-16 model. | Download Scientific Diagram’. Accessed: May 14, 2024. [Online]. Available: https://www.researchgate.net/figure/The-architecture-of-the-VGG-16-model_fig3_335918494