fire

Article

Multi-Classification Using YOLOv11 and Hybrid YOLO11n-MobileNet Models: A Fire Classes Case Study

Eman H. Alkhammash

Department of Computer Science, College of Computers and Information Technology, Taif University, P.O. Box 11099, Taif 21944, Saudi Arabia; eman.kms@tu.edu.sa

Abstract: Fires are classified into five types: A, B, C, D, and F/K, according to the components involved in combustion. Recognizing fire classes is critical, since each kind demands a unique suppression approach. Proper fire classification helps to decrease the risk to both life and property. The fuel type is used to determine the fire class, so that the appropriate extinguishing agent can be selected. This study takes advantage of recent advances in deep learning, employing YOLOv11 variants (YOLO11n, YOLO11s, YOLO11m, YOLO11l, and YOLO11x) to classify fires according to their class, assisting in the selection of the correct fire extinguishers for effective fire control. Moreover, a hybrid model that combines YOLO11n and MobileNetV2 is developed for multi-class classification. The dataset used in this study is a combination of five existing public datasets with additional manually annotated images, to create a new dataset covering the five fire classes, which was then validated by a firefighting specialist. The hybrid model exhibits good performance across all classes, achieving particularly high precision, recall, and F1 scores. Its superior performance is especially reflected in the macro average, where it surpasses both YOLO11n and YOLO11m, making it an effective model for datasets with imbalanced classes, such as fire classes. The YOLO11 variants achieved high performance across all classes. YOLO11s exhibited high precision and recall for Class A and Class F, achieving an F1 score of 0.98 for Class A. YOLO11m also performed well, demonstrating strong results in Class A and No Fire with an F1 score of 0.98%. YOLO11n achieved 97% accuracy and excelled in No Fire, while also delivering good recall for Class A. YOLO11l showed excellent recall in challenging classes like Class F, attaining an F1 score of 0.97. YOLO11x, although slightly lower in overall accuracy of 96%, still maintained strong performance in Class A and No Fire, with F1 scores of 0.97 and 0.98, respectively. A similar study employing MobileNetV2 is compared to the hybrid model, and the results show that the hybrid model achieves higher accuracy. Overall, the results demonstrate the high accuracy of the hybrid model, highlighting the potential of the hybrid models and YOLO11n, YOLO11m, YOLO11s, and YOLO11l models for better classification of fire classes. We also discussed the potential of deep learning models, along with their limitations and challenges, particularly with limited datasets in the context of the classification of fire classes.

Keywords: YOLOv11; classification; fire classes; YOLOv8; MobileNet; deep learning

Academic Editors: Giovanni Laneve and Darko Stipanicev

Received: 7 December 2024
Revised: 16 December 2024
Accepted: 24 December 2024
Published: 3 January 2025

Citation: Alkhammash, E.H. Multi-Classification Using YOLOv11 and Hybrid YOLO11n-MobileNet Models: A Fire Classes Case Study. *Fire* **2025**, *8*, 17. https://doi.org/10.3390/fire8010017

1. Introduction

Fires are chemical reactions generated by combustion, in which flammable elements interact with oxygen in the presence of heat to release energy in the form of heat, light, flames, and smoke. Fires vary in kind and cause; therefore, understanding fire classifications is critical for effective and fast response. The National Fire Protection Association (NFPA) categorizes fire into five classes according to the burning material: Class A, Class B, Class C, Class D, and Class F/K [1]. Class A fires involve solid combustible materials such as wood, paper, and cloth, which are prevalent in houses and public areas. Class B fires are caused by flammable liquids, such as flammable gas and oil, that can spread quickly and create explosive risks. Class C fires are electrical components fires that can spread quickly and be highly explosive. Class D fires include flammable metals such as magnesium and aluminum, and are known for their bright white flames and thick smoke, and are difficult to extinguish. Class F/K fires involve cooking oils and greases, which frequently produce heavy smoke and heat in kitchens and are difficult to control with water. To prevent or mitigate future disasters of this nature, it is essential to understand the factors that contribute to these catastrophic events [2]. Moreover, it is crucial to determine the fire class to select the appropriate extinguishing agents that allow for a faster firefighting process.

Fires are a serious danger to life because fires affect people, animals, plants, and the environment. They can be caused by human actions and natural events. Forest fires are caused by a variety of factors, and their distribution varies not only across countries, but also within parts of the same country [3]. Wildfires increased in 2020 due to dry weather, human activity, and inadequate environmental policies and surveillance [2].

The majority of fires in Europe, particularly in the Mediterranean basin, are caused by humans, most notably by arson. Socioeconomic factors such as unemployment rates and agricultural activity variables can also be some of the main reasons for both purposeful and unintentional fires [3]. Environmental elements such as weather, fuel availability, and topography are important in forest fires. For fires caused by both human activity and lightning, ignition patterns reveal a geographical gradient, which is heavily impacted by climate, fuel types, and population density. The occurrence of fires varies in timing as well [3]. In populated areas, fires caused by humans often occur most frequently in the afternoon, during peak human activity. In contrast, fires triggered by lightning in remote areas are more influenced by seasonal weather, and take place almost during the summer [3]. Wildfires can also be started by humans, often due to things like burning wood piles, leaving campfires unattended, throwing away cigarette butts carelessly, or even intentionally starting fires. The rise in fuel hazards has likely led to more land being burned by lightning-caused fires, and climate change may also have added to the increase in wildfires [4]. Global warming contributes to the increase in forest fires, posing a significant threat to the environment and human health due to the resulting pollution.

According to the NFPA's 2022 fire loss study, firefighters in the United States responded to an estimated 1,504,500 fires. Approximately 382,500 of these occurred in residential buildings, 140,000 in non-residential structures, 760,000 outside of structures, and 222,000 in vehicles. From 2018–2022, an annual average of 15,941 structure fires involved flammable gases as the first ignited material, causing 191 civilian deaths, 747 injuries, and $402 million in property damage [5]. There were around 10,774 home fires, the majority of which were caused by cooking, and 5166 in non-residential properties. Natural gas was the most often-ignited material, followed by LP-Gas, which was more common in home fires than non-home fires. The frequency and proportion of these fires have increased during the last decade [5]. The financial impact of fires has also grown, reaching an estimated total loss of $18.1 billion in 2022, marking a 28.9% increase since 2013 [6]. Developing more advanced techniques is critical to reducing the hazards caused by fires. Machine learning and deep learning are useful techniques for developing applications that allow for early fire detection, forecast fire patterns, and enable faster response times. These applications, based on AI techniques, can handle massive amounts of data in real time, identify fire risks, classify fire type, monitor fire spread, and ultimately assist in minimizing harm to people and the environment. There are many efforts towards fire detection and classification using deep learning models that are applied to different types of data, such as images and videos [7–18]. Kim, B. and Lee, J. developed a video-based deep learning method for fire detection, using Faster Region-based Convolutional Neural Network (Faster R-CNN) and Long Short-Term Memory (LSTM). Faster R-CNN identifies suspected fire regions, whereas LSTM accumulates non-fire regions in successive frames. The decisions from successive short-term periods combined in majority voting for a final decision. Flame and smoke areas are also analyzed, and their temporal changes interpret the dynamic behavior of fire. Results show improved accuracy and reduced false detections compared to image-based or short-term video-based methods [19]. Seydi, S.T. et al. [20] have developed a deep learning framework called Fire-Net that is trained on Landsat-8 imagery to detect active fires and burning biomass. Fire-Net combines optical (RGB) and thermal data. Fire-Net achieves a more accurate representation. It uses residual and separable convolution blocks to capture deeper features from complex datasets. Tests show a high accuracy of 97.35%, with reliable detection of even small fires. The images cover several regions, including forests in Australia, North America, the Amazon, Central Africa, and Chernobyl [20]. Abdusalomov, A.B. et al. [21] have developed an enhanced forest fire detection method using an updated Detectron2 platform. A custom-labeled dataset of 5200 images was created. The results of the detection method reach 99.3%. The model detects small fires over long distances, day and night, demonstrating the effectiveness of the Detectron2 algorithm for precise fire detection [21]. Avazov, K. et al. [18] created a fast fire detector that triggers an alarm within 8 s of detecting even minor sparks. The method adopts YOLOv4, which operates on the Banana Pi M3 board based on three layers. Testing shows that the original YOLOv4 approach did not yield the desired results. Therefore, the training data were extended using data augmentation. The improved model detects fires quickly and accurately across various conditions: sunny, cloudy, day, or night, which makes it ideal for urban fire monitoring and the protection of smart cities [18].

The contributions of this paper are as follows:

1. We created a new dataset for fire classification that covers Classes A, B, C, and F/K, which was validated by firefighting experts. This dataset combined different images from five benchmark existing datasets. Moreover, additional images are added manually to the dataset;
2. We developed fire classification models based on YOLOv11, using five different YOLO11 variants applied to our dataset. All YOLO11 variants show excellent performance. YOLO11n, YOLO11m, YOLO11s, and YOLO11l models showed the highest performance, especially in classifying challenging categories like Class F/K;
3. We developed a hybrid model based on YOLO11n and MobileNetV2 that outperformed other YOLO11 models in terms of macro metrics;
4. A comparison with a similar study and other models such as the YOLOv8n model and YOLOv8s model was also performed to show the performance of the proposed hybrid model;
5. We discussed the potential of YOLO11 for fire classification and highlighted challenges and limitations related to the scarcity of comprehensive fire datasets.

The rest of this paper is organized as follows: Section 2 covers related works, Section 3 presents the methodology, Section 4 discusses the results, followed by the discussion in Section 5, and the conclusion in Section 6.

2. Related Studies

There are several studies that utilize deep learning methods for both fire detection and classification. This section highlights key research focused on advancements in fire classification techniques using deep learning.

Yar, Hikmat, et al. [22] developed the Dual Fire Attention Network (DFAN), which enhances fire detection by using attention mechanisms that yield significantly emphasized feature maps and capture spatial details. The DFAN is optimized for real-world use by cutting extra parameters to increase speed by 50% FPS values. Additionally, the authors created a fire dataset with diverse fire/non-fire images and multiple classes, including indoor and outdoor fire types. The testing of DFAN on four datasets shows that DFAN outperformed 21 state-of-the-art methods and provides accurate and fast fire detection on edge devices.

Xue, Q., Lin, H., and Wang, F. developed the YOLOv5-based model for forest fire classification and detection. The developed model performs well in identifying forest fires, but struggles to distinguish surface fires from canopy fires over time, especially with low-resolution images. Therefore, the model was improved by introducing SIoU Loss and directionality in the loss function cost for better training and detection. The CBAM attention module is introduced for higher classification accuracy. Moreover, the PANet layer is improved as a weighted BiFPN, for better inspection of different fire types. Experimental results show that the model outperforms YOLOv5 in detecting forest, surface, and canopy fires [23].

Khan, S., and Khan, A. [24] introduced FFireNet, a deep learning model using the MobileNetV2, and added fully connected layers for classifying forest fire images. The model's performance was assessed using various metrics and compared to other CNN models. Results show that the proposed approach achieved 98.42% accuracy, 1.58% error rate, 99.47% recall, and 97.42% precision, which demonstrates its potential for forest fire classification.

Akagic, A., and Buza, E. [25] developed a wildfire image classification model based on convolutional neural networks called LW-FIRE. They explored different ways to use the existing dataset to train the model, and introduced a new dataset transformation method to expand sample size, improving accuracy and generalization. Experimental results show that LW-FIRE outperforms state-of-the-art methods and is well-suited for real-time wildfire image classification.

Lee, Jiwon et al. [26] introduced a fire classification system based on federated learning (FL) and image clustering for industries. The system accurately classifies fire, smoke, and normal conditions. The server in the proposed system uses a pre-trained vision transformer model using bisecting K-means to obtain clustering weights. The clients use these weights to cluster local data with the K-means algorithm. The system achieves nearly 99% accuracy, and strong clustering quality. The normalized mutual information (NMI) is above 0.6 and the silhouette score is 0.9, demonstrating improved clustering quality for effective real-time fire detection.

Refaee, E. A. et al. [27] introduced a publicly available benchmark dataset with 1353 manually labeled images in four categories capturing different fire origins and no fire class. It presents a system that uses eight deep learning models for detecting fires and classifying fire types: solid material, chemical, electrical, and oil-based fires. In a single-level, five-way classification, the system achieves 94.48% accuracy, while in a two-level classification, it achieves 98.16% for fire detection and 97.55% for fire-type classification, using DenseNet121 and EfficientNet-b0. The results show that electrical and oil-based fires are the hardest to detect.

Park, Minsoo et al. [28] developed a multilabel classification (MLC) model for wildland fires, using transfer learning and data augmentation to output multiple details from a single image. VGG-16, ResNet-50, and DenseNet-121 were applied as pre-trained models, trained on a custom dataset, and compared using performance metrics. Tests showed that transfer learning and data augmentation enhance the MLC model's performance. The heatmap processed from gradient-weighted class activation mapping visualizes prediction reliability. Table 1 provides an overview of the studies discussed in the related work.

Table 1. Summary of the relevant studies.

Study	Approach	Key Contributions	Applications
Yar, Hikmat et al. [22]	DFAN with attention mechanisms	Enhanced fire detection speed (50% FPS), and outperformed 21 state-of-the-art methods	Fire classification and detection
Xue, Q., Lin, H., and Wang, F. [23]	YOLOv5-based model with CBAM attention and SIoU Loss	Improved forest fire detection accuracy and outperformed YOLOv5	Forest fire classification and detection
Khan, S. and Khan, A. [24]	FFireNet using MobileNetV2	98.42% accuracy, 99.47% recall, and 97.42% precision	Forest fire classification
Akagic, A. and Buza, E. [25]	LW-FIRE with CNN architecture	High accuracy via dataset transformation method	Wildfire image classification
Lee, Jiwon et al. [26]	Federated learning and clustering	99% accuracy, effective clustering, NMI is above 0.6, and silhouette is 0.9	Fire classification systems
Refaee, E. A. et al. [27]	Multi-model classification system	Benchmark dataset with 1353 images. A 98.16% accuracy for fire detection and 97.55 accuracy for fire-type classification (two-level classification)	Fire type classification and detection
Park, Minsoo et al. [28]	MLC model with transfer learning	Enhanced multilabel classification with reliable heatmaps and robust performance with pretrained models	Wildfire image classification

Most studies were primarily focused on classifying smoke and fire, or fire and not fire, and were often associated with limited class categories. In contrast, this study emphasizes fire classification based on fire classes, creating a new dataset from existing datasets and integrating advanced YOLO models for classification. Additionally, a hybrid model combining YOLO11n and MobileNetV2 was developed, demonstrating promising results in terms of macro metrics.

3. Methodology

3.1. Dataset

We have created a dataset using images from five existing datasets, including dataset [29], which includes a variety of images from wildfires and bushfires, comprising 1900 images focused on forest fires. The second dataset [30] includes images of forest fires captured during daytime, dusk, and nighttime. The third dataset [22] includes images of indoor fire classes and outdoor fire classes. The fourth dataset [27,31] involves classes of solid material-based fire, electrical-based fire, chemical-based fire, and oil-based fire. The fifth dataset is the Bowfire dataset [32], which consists of images with various resolutions divided into two categories: fire and no fire. The fire images capture emergency situations from diverse incidents, including buildings fires, industrial fires, and car accidents. We have also manually added images from different sources [33–36]. Figure 1 represents the created fire dataset that contains data collected from various sources and datasets and organized into fire classes (A, B, C, D, and K/F). A firefighting expert was involved in approving the categories to ensure accurate classification of fire classes.

Figure 2 illustrates the distribution of images across different fire classes in the dataset. Class A has the highest number of images (2446), followed by No Fire (1054), while smaller counts are observed for Class B (186), Class F/K (164), Class D (169), and Class C (462). This highlights a significant class imbalance, with Class A being the largest class in the dataset. The total number of images is 4481.

Figure 3 displays a selection of sample images representing each fire class, which provides visual examples of the distinct characteristics associated with classes A, B, C, D, and K/F in the dataset.

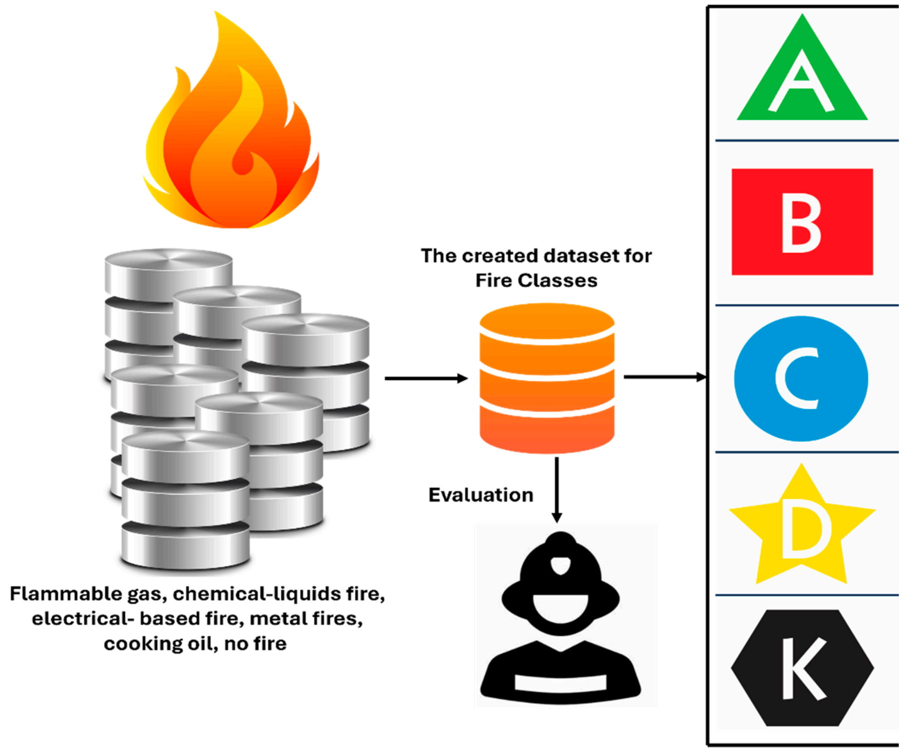


Figure 1. Fire dataset for classification of fire classes. Data was collected from multiple sources and organized into fire classes (A, B, C, D, and K/F). A firefighting expert was involved to validate and ensure accurate classification.

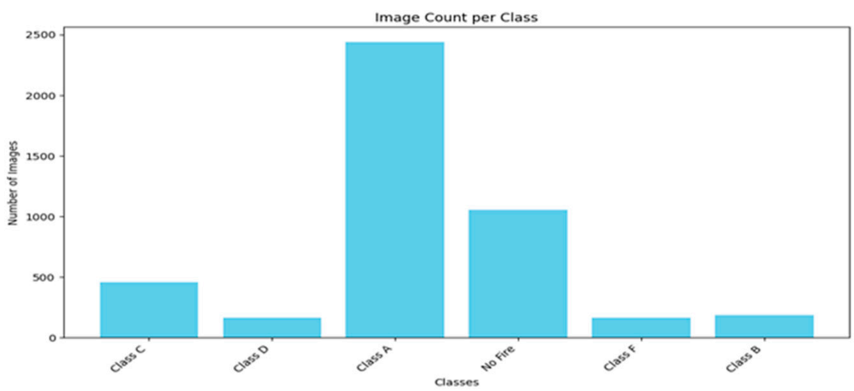


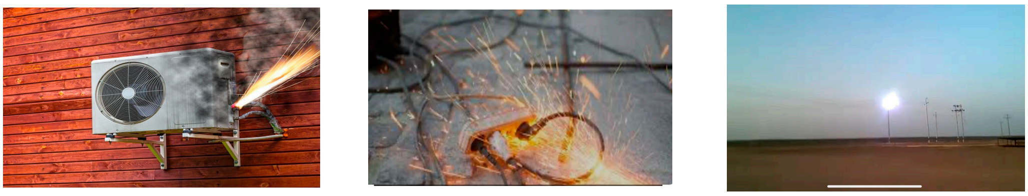
Figure 2. The image distribution across fire classes (A, B, C, D, and F/K).

3.2. YOLO11

YOLO11 (https://docs.ultralytics.com/models/yolo11/) is the latest version in the Ultralytics YOLO series, designed for a wide array of computer vision applications, including object detection, instance segmentation, image classification, pose estimation, and oriented object detection (OBB). YOLO11 enhances feature extraction through an upgraded backbone and neck architecture, achieving a higher mean Average Precision (mAP) on the COCO dataset, with 22% fewer parameters than YOLOv8m. The YOLO11 model comes in several variants: YOLO11n, YOLO11s, YOLO11m, YOLO11l, and YOLO11x with YOLO11n and YOLO11s having fewer parameters than the others. In this study, we use YOLO11 for classification purposes. YOLO 11 classifies images into one of the sets of predefined classes: Class A Fire, Class B Fire, Class C Fire, Class D Fire, Class F/K Fire, or No Fire.

(**a**) Class A Fire (solid combustible materials fires)

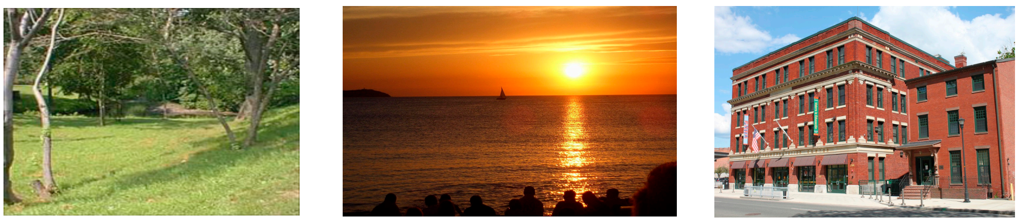
(**b**) Class B Fire (flammable liquids fires)



(**c**) Class C Fire (electrical components fires)

(**d**) Class D Fire (flammable metals fires)

(**e**) Class F/K Fire (cooking oils and grease fires)



(**f**) No Fire

Figure 3. Sample of images for each class.

The output of an image classifier includes one class label and a confidence score. Figure 4 shows the YOLO11 architecture, which includes new modules like the C2PSA (Cross-Stage Partial with Self-Attention) module. This module allows YOLO11 to capture context more effectively across layers, improving accuracy, especially for detecting small and colluded objects. Additionally, the C2f block has been replaced by C3k2, a customized CSP Bottleneck with two convolutions, which uses a smaller filter size to maintain accuracy while increasing speed and efficiency.

The equation representing C3k2 block is as follows [37]:

$$\text{C3k2} \times (\text{X}) = \text{Conv} \times (\text{Split} \times (\text{X})) + \text{Conv} \times (\text{Merge} \times (\text{Split} \times (\text{X}))) \tag{1}$$

In Split × (X), the feature map is divided into two parts, the first part processed through the bottleneck. Merge then combines the resulting outputs. In this study, we have adopted all variants of YOLO11 for the classification of fire classes.

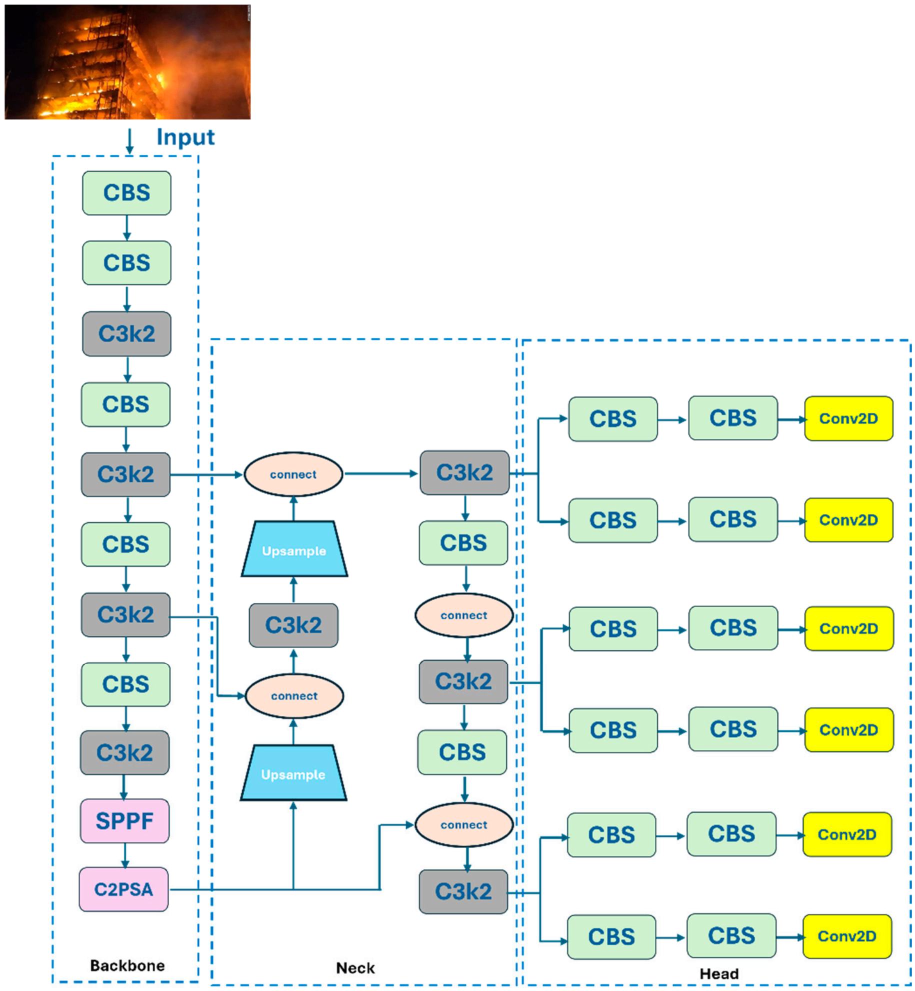


Figure 4. The architecture of YOLO 11 for classification of fire classes [38,39].

3.3. MobileNetV2

MobileNetV2 is a deep learning architecture built on MobileNetV1. It introduces inverted residuals and linear bottlenecks [40]. MobileNetV2 uses depth-wise separable convolutions. The inverted residuals module takes a low-dimensional compressed representation as an input, which is first expanded to high dimension and filtered with a lightweight depth-wise convolution [40]. Features are subsequently projected back to a low-dimensional representation with a linear convolution. This design balances speed and accuracy, making it ideal for applications in image classification, object detection, and semantic segmentation [40].

The custom YOLOv11 for multi-class classification is outlined in Table 2.

We have used FocalLoss for multi-class classification and AdamW as the optimizer, with a learning rate of 5×10^{-5} and a weight decay of 0.001. The custom YOLO backbone (feature extractor) is taken from the YOLO11n-cls model and MobileNetV2 backbone. The first 9 layers of YOLO11n-cls are used to extract initial features from the input image.

The MobileNetV2 backbone (excluding the classification layers) extracts complementary features. The attention mechanism refines the feature maps, while pooling ensures consistent spatial output to enable effective feature fusion. The features from both backbones are concatenated to form a comprehensive feature representation. The classification head generates the classification output for the given number of classes. Head does not predict bounding boxes or objectness scores. It produces class scores for the entire image, ultimately classifying the image into a single category, with no information about object locations within the image.

Table 2. The custom YOLOn11 and MobileNetV2 for multi-class classification.

Backbone	YOLO11n-cls 1. Conv2d(3, 16, kernel_size = 3 × 3, stride = 2 × 2, padding = 1 × 1, bias = False) 2. Conv Layer 1: Conv2d(16, 32, kernel_size = 3 × 3, stride = 2 × 2, padding = 1 × 1, bias = False) 3. C3k2 Block 2: cv1: Conv2d(32, 32, kernel_size = 1 × 1, bias = False), cv2: Conv2d(48, 64, kernel_size = 1 × 1, bias = False) 4. Conv Layer 3: Conv2d(64, 64, kernel_size = 3 × 3, stride = 2 × 2, padding = 1 × 1, bias = False) 5. C3k2 Block 4: cv1: Conv2d(64, 64, kernel_size = 1 × 1, bias = False), cv2: Conv2d(96, 128, kernel_size = 1 × 1, bias = False) 6. Conv Layer 5: Conv2d(128, 128, kernel_size = 3 × 3, stride = 2 × 2, padding = 1 × 1, bias = False) 7. C3k2 Block 6: cv1: Conv2d(128, 128, kernel_size = 1 × 1, bias = False), cv2: Conv2d(192, 128, kernel_size = 1 × 1, bias = False) 8. Conv Layer 7: Conv2d(128, 256, kernel_size = 3 × 3, stride = 2 × 2, padding = 1 × 1, bias = False) 9. C3k2 Block 8: cv1: Conv2d(256, 256, kernel_size = 1 × 1, bias = False), cv2: Conv2d(384, 256, kernel_size = 1 × 1, bias = False) MobileNetV2 backbone
Attention mechanism	Attention applied to YOLO11n-cls and MobileNet features using Conv2D, BatchNorm2D, and Sigmoid activation
	Adaptive Average Pooling
	Feature Concatenation: Merges YOLO11n-cls and MobileNet feature vectors
Head	Fully connected (linear) layer, batch normalization, and final linear layer for classification

3.4. Performance Metrics

In this study, we adopt four performance metrics, which are accuracy, precision, recall, and F1 score, and they provide insights into how well a model is performing.

3.4.1. Accuracy

Accuracy reflects the proportion of correct predictions (both positive and negative) among all predictions. A high accuracy indicates the model correctly identifies most classes. It is computed as:

$$Accuracy = \frac{True\ Positives + True\ Negatives}{True\ Positives + True\ Negatives + False\ Positives + False\ Negatives} \tag{2}$$

3.4.2. Precision

Precision measures the accuracy of the positive predictions made by the YOLOv11 model. High precision means fewer false positives, which is essential in applications where false alarms can be disruptive or costly, such as fire safety applications.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \tag{3}$$

3.4.3. Recall

This metric indicates the model's ability to find all relevant instances within a dataset. High recall means the model catches most actual fire instances; even if it occasionally mislabels non-fire images as types of fires, false negatives are minimized.

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} \tag{4}$$

3.4.4. F1 Score

This is the harmonic mean of precision and recall, which provides a single metric that balances both. A high F1 score indicates that the model is both accurate and comprehensive in identifying fire instances.

$$F1\ score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{5}$$

4. Results

Tables 3 and 4 provide the configuration parameters for the study setup and the training parameters for the YOLO11 deep learning models, respectively.

Table 3. Setup parameters of the study.

Parameter	Value
Epochs	50
Batch size	16
Optimal algorithm	Adam
Model weight	YOLO11n, YOLO11s, YOLO11m, YOLO11l, and YOLO11x

Table 4. Training parameters of the deep learning models.

Name	Type
CPU	Intel Xeon CPU @ 2.20 GHz
GPU	NVIDIA Tesla T4 GPU
RAM	32 GB
Framework	PyTorch 2.5
Accelerator	Tesla T4

The test results of YOLO11 models are shown in Table 5. Table 5 shows the performance comparison of YOLO model variants (a) YOLO11n, (b) YOLO11s, (c) YOLO11m, (d) YOLO11l, and (e) YOLO11x based on precision, recall, F1 score, and accuracy metrics. All YOLO11 models performed exceptionally well, showing strong accuracy and reliability across the board. YOLO11m has strong performance, particularly in Class B (0.97 F1 score) and Class F/K (0.97 F1 score), along with 0.98 F1 score for No Fire. YOLO11n delivers good results, especially in No Fire (1.00 F1 score), and performs well in most other classes, particularly Class A and Class D, although its performance in Class B and Class C is slightly lower. YOLO11s excels in Class A (0.98 F1 score) and achieves strong precision and F1 score for No Fire (0.99). However, its performance in Class B is somewhat lower compared to the other models.

The evaluation metrics of the hybrid model are presented in Table 6. Overall, the model performs exceptionally well across all classes, achieving high precision, recall, and F1 score metrics. Class A achieves a precision, recall, and F1 score of 0.98 each, demonstrating the model's ability to accurately identify this class with minimal errors. Class B maintains a perfect precision of 1.00, a recall of 0.95, and an F1 score of 0.97, indicating flawless predictions when identifying Class B and successfully capturing 95% of all actual instances. Class C shows a precision of 0.90, a recall of 0.96, and an F1 score of 0.93, demonstrating effective classification in most instances, with a few false positives. Class D exhibits outstanding performance, with a perfect precision of 1.00, a recall of 0.94, and an F1 score of 0.97, highlighting its ability to accurately predict Class D and correctly identify 94% of actual instances. Class F achieves a precision, recall, and F1 score of 1.00 each, signifying perfect identification without any misclassifications. No Fire achieves a precision, recall, and F1 score of 0.98 each, indicating the model's high accuracy in distinguishing non-fire instances with minimal errors.

Table 5. Evaluation metrics (precision, recall, F1 score, and accuracy) for YOLO11 model variants (a) YOLO11n, (b) YOLO11s, (c) YOLO11m, (d) YOLO11l, and (e) YOLO11x.

	YOLO11n			**YOLO11s**			**YOLO11m**			**YOLO11l**			**YOLO11x**		
	Precision	**Recall**	**F1 Score**	**Precision**	**Recall**	**F1 Score**	**Precision**	**Recall**	**F1 Score**	**Precision**	**Recall**	**F1 Score**	**Precision**	**Recall**	**F1 Score**
Class A	0.96	0.98	0.97	0.96	0.99	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.97	0.97	0.97
Class B	0.95	0.95	0.95	0.94	0.84	0.89	1.00	0.95	0.97	0.94	0.89	0.92	0.81	0.89	0.85
Class C	0.93	0.85	0.89	0.98	0.89	0.93	0.96	0.94	0.95	0.92	0.96	0.94	0.98	0.89	0.93
Class D	0.94	1.00	0.97	0.94	0.94	0.94	0.89	1.00	0.94	0.94	1.00	0.97	0.89	1.00	0.94
Class F	0.94	1.00	0.97	0.94	1.00	0.97	0.94	1.00	0.97	1.00	0.94	0.97	0.94	0.88	0.91
No Fire	1.00	0.99	1.00	0.99	0.98	0.99	0.98	0.98	0.98	0.99	0.97	0.98	0.98	0.98	0.98
Accuracy		0.97			0.97			0.97			0.97			0.96	
Macro avg	0.96	0.96	0.96	0.96	0.94	0.95	0.96	0.97	0.97	0.96	0.96	0.96	0.93	0.94	0.93
Weighted avg	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.96	0.96	0.96

Table 6. Evaluation metrics (precision, recall, F1 score, and accuracy) for the hybrid model.

	Precision	**Recall**	**F1 Score**
Class A	0.98	0.98	0.98
Class B	1.00	0.95	0.97
Class C	0.90	0.96	0.93
Class D	1.00	0.94	0.97
Class F	1.00	1.00	1.00
No Fire	0.98	0.98	0.98
Accuracy		0.98	
Macro Avg	0.98	0.97	0.97
Weighted Avg	0.98	0.98	0.98

Although YOLO11n, YOLO11m, and the hybrid model all perform strongly, the hybrid model achieves the highest macro-average performance. The macro average is particularly useful in imbalanced classes such as fire class datasets. It is used when the dataset contains imbalanced classes, meaning one or more of the classes have significantly more samples than others. In such cases, metrics like accuracy or weighted averages can be biased toward the majority class, giving a false sense of model performance. The hybrid model is very promising in macro average compared to other YOLOv11 models.

The macro average of the hybrid model (0.98 for precision, 0.97 recall, and 0.97 F1 score) reflects the model's balanced performance across all classes, with no significant bias toward one class over others.

The training and validation loss curves of the hybrid model are shown in Figure 5. The training loss decreased consistently across the epochs, showing effective learning. The training loss starts at 0.5 and continues to decrease steadily, reaching less than 0.1 by epoch 50, and validation loss also decreases. This suggests that the model is generalizing well to the validation set. Overall, both losses show a steady decline, suggesting effective model training.

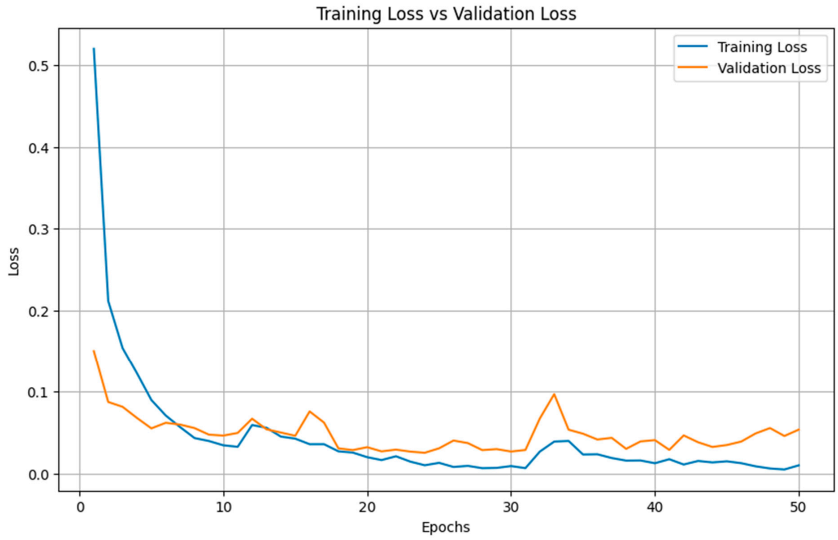


Figure 5. The training and validation loss curves of the hybrid model.

The precision–recall curve of the hybrid model in Figure 6 demonstrates how well the model performs across different classes. Minor variations in precision for certain classes like Class B and Class D might indicate some challenging instances or overlaps, but overall, the performance is very good.

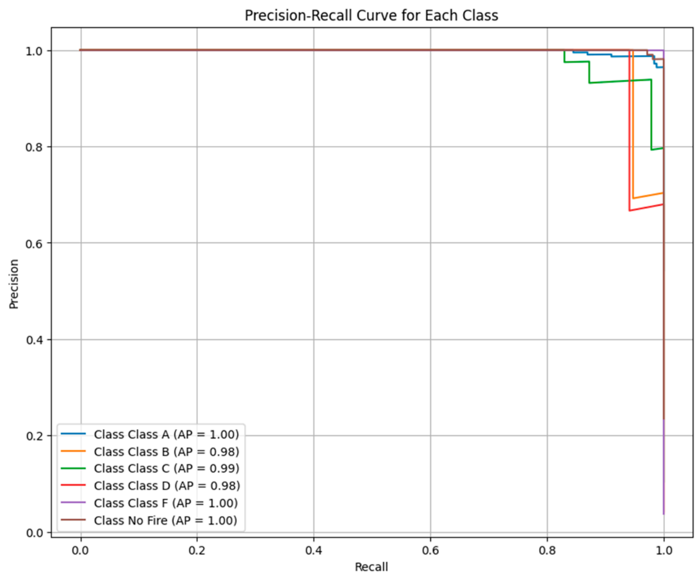
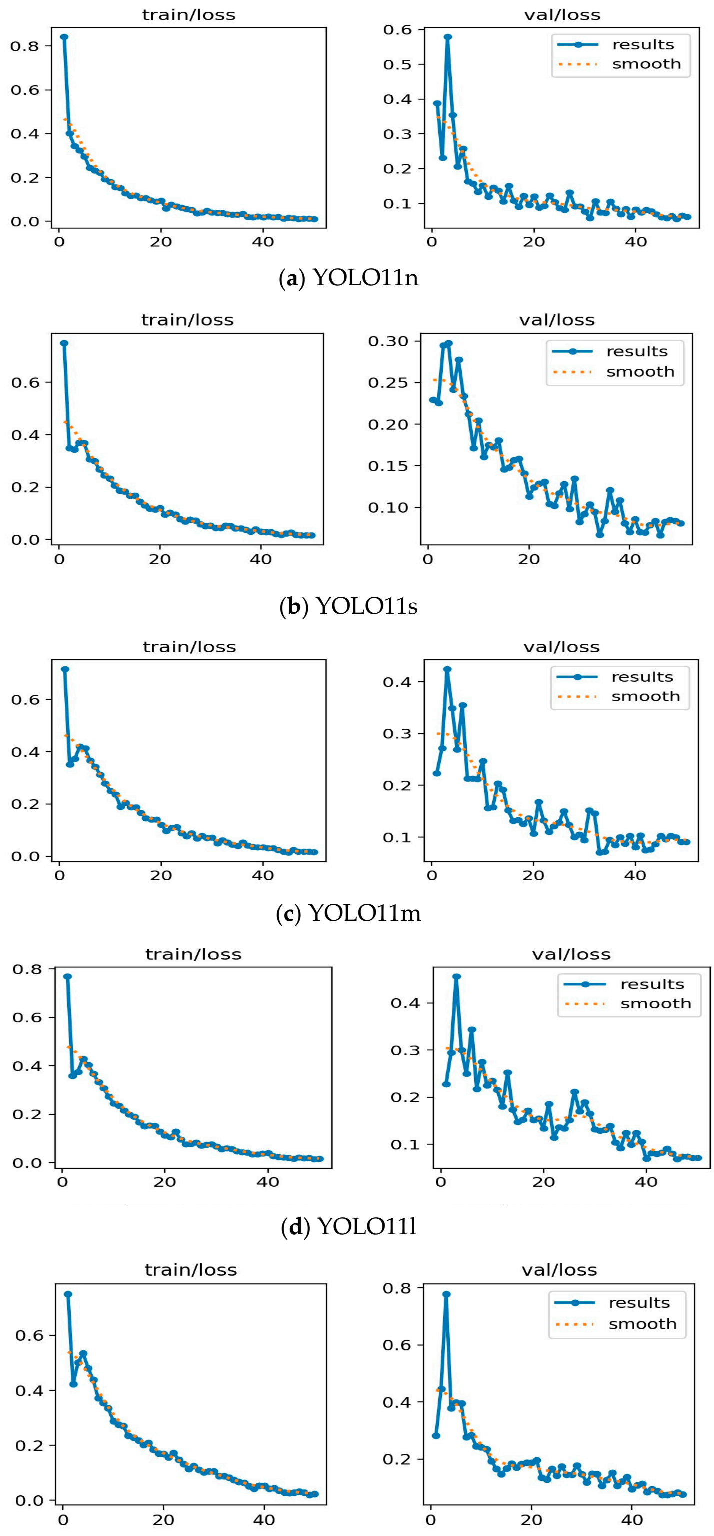


Figure 6. The precision–recall curve of the hybrid model.

In general, regarding the training and validation loss curves of YOLO11 models, over 50 epochs, as seen in Figure 7, show a steady decline, indicating effective learning. Both losses flatten and decrease towards the end. The training loss of YOLO11n started at around 0.8 and rapidly decreased to a value below 1. The validation loss was initially at 0.4 and progressively decreased to a value near 0.1. However, the trend showed consistent improvement, ending near 0.1. In YOLO11s, the loss also started relatively high but decreased quickly, it eventually remained between 0.15 and 0, though it did not stabilize and continued to fluctuate throughout training. Nevertheless, this fluctuation had no significant effect on the final accuracy which was 97% similar to other models. The validation of YOLO11m started at around 0.2 and decreased to around 0.1. The validation loss of YOLO11l started at around 0.2 and gradually decreased to around 0.1. Similarly, the validation loss of YOLO11x started at 0.2 but eventually converged to a value slightly above 0.1. All versions of YOLO11 have minimum losses.



(**a**) YOLO11n

(**b**) YOLO11s

(**c**) YOLO11m

(**d**) YOLO11l

(**e**) YOLO11x

Figure 7. Training and validation loss curves of the YOLO11 models (**a**) YOLO11n, (**b**) YOLO11s, (**c**) YOLO11m, (**d**) YOLO11l, and (**e**) YOLO11x.

We have also evaluated the YOLOv8n and YOLOv8s models (smallest versions of YOLOv8) with our dataset and compared their performance with the hybrid model. The comparison highlights the hybrid model's superior macro average metrics when compared to YOLOv8n and YOLOv8s, as shown in Table 7.

Table 7. Performance comparison of the hybrid model and YOLOv8n and YOLOv8s models.

	YOLOv8n			YOLOv8s			Hybrid Model		
	Precision	Recall	F1 Score	Precision	Recall	F1 Score	Precision	Recall	F1 Score
Class A	0.98	1.00	0.99	0.97	0.99	0.98	0.98	0.98	0.98
Class B	0.94	0.84	0.89	0.94	0.84	0.89	1.00	0.95	0.97
Class C	0.95	0.89	0.92	0.93	0.87	0.90	0.90	0.96	0.93
Class D	0.85	1.00	0.92	0.94	1.00	0.97	1.00	0.94	0.97
Class F	0.83	0.88	0.86	0.94	1.00	0.97	1.00	1.00	1.00
No Fire	0.99	0.96	0.98	0.99	0.97	0.98	0.98	0.98	0.98
Accuracy		0.97			0.97			0.98	
Macro avg	0.92	0.93	0.93	0.95	0.95	0.95	0.98	0.97	0.97
Weighted avg	0.97	0.97	0.97	0.97	0.97	0.97	0.98	0.98	0.98

Table 8 provides a comparison between the proposed hybrid model and a similar study [27].

Table 8. Comparison of the proposed hybrid model performance with results from study [27].

Model	Epoch	Dataset	Accuracy	Task	F1 Score
MobileNetV2	300	The dataset includes 1353 instances, with 541 labeled as no fire, 308 as wood and solid materials, 163 as flammable gas and chemical liquid fire, 187 electrical based fire, and 154 as oil-based fires.	94.48	Fire classification of fire types	0.944
The proposed hybrid model	50	The proposed dataset	0.98	Fire classification of fire classes	0.98

The hybrid model demonstrated higher accuracy (98%) compared to MobileNetV2 (94.48% accuracy), requiring only 50 epochs on a more extensive dataset of 4481 instances, versus MobileNetV2's 300 epochs on 1353 instances across fewer categories.

The hybrid model is also used to train the dataset [27,31] that consists of classes solid combustible materials, chemical liquid fire, electrical-based fire, oil-based fire, and no fire. We select this dataset to train it first to predict for other datasets, because it has many classes compared to other datasets. We organized these classes as Class A, Class B, Class C, Class F, and No Fire, to correspond to the solid combustible materials class, flammable gas class, electrical-based fire, oil-based fire class, and no fire class, respectively. Then we use the best model generated to predict for other datasets. The hybrid model shows good capability in predicting other classes. Details can be seen in Appendix A.

5. Discussion

The classification of fire classes is a challenging task due to many factors, including the limited availability of comprehensive datasets that cover a wide range of fire types. Fire classification is an essential task for advancing fire detection systems and related real-world applications. Current datasets are limited in scope, often covering only a few specific fire types, which restricts the model's ability to generalize across diverse fire scenarios. Additionally, unbalanced datasets can cause some models to be biased towards more common fire types, potentially overlooking less frequent but critical cases. Creating a dataset for fire classification is challenging because not all fires are visually identifiable, and classification should consider the types of burned materials involved. Some fires can belong to multiple classes, which highlights the need for more flexible classification approaches. This section explores these points in more detail.

- To the best of our knowledge, there is only one dataset specifically for the classification of fire classes that includes four fire classes and no fire class [27]. The limited datasets make it challenging to train models that can accurately identify classes of diverse fire types across real-world scenarios. This highlights the need for more comprehensive datasets to improve fire classification accuracy of models;
- An unbalanced dataset can lead the model to favor majority classes, like Class A, and perform poorly on minority classes. Despite this, YOLO11 successfully classified most fire instances with high accuracy. However, it struggled with a few images in smaller classes, such as Class F, where the limited data made accurate classification more challenging.
- Building a dataset for fire classification is challenging because not all fires can be identified visually. Some require additional characteristics, such as recognition by sound. Fire classes are not directly determined by the type of object. For instance, building fires can be classified as Class A or Class B depending on the materials involved in the fire. The classification should be based on the materials causing the fire. This complexity makes accurate classification more difficult;
- Some fires can belong to multiple classes, as they can share characteristics from more than one category. To address this, it might be beneficial to introduce additional fire classes that represent these combinations, effectively capturing complex fire types. This approach could improve model accuracy by allowing classification for cases where fires do not fit into a single predefined category;
- The YOLOv11 versions are highly advanced models, with YOLO11n, YOLO11m, YOLO11s and YOLO11l delivering particularly promising results. A comparison study [39] highlights that YOLO11m, YOLO11n, and YOLO11s achieve high accuracy, low processing time, efficient power consumption, and minimal disk usage. This performance is attributed to the use of C3k2 and C2PSA blocks, which enhance feature extraction and preserve contextual information, resulting in improved convergence and overall performance [39]. The proposed hybrid model, which integrates YOLO11n with MobileNetV2, demonstrates superior performance across macro metrics.

6. Conclusions

In this study, we first introduce a fire dataset developed to improve the classification of different fire classes, addressing the crucial need for more reliable fire detection systems. Our dataset covers a broad range of fire categories, including no fire, wood and solid materials, flammable gas and chemical liquid fires, oil-based fires, cooking oil fires, and electrical fires. We developed a hybrid model that combines YOLO11n and MobileNetV2, which shows good results for macro metrics. We also developed several deep learning models based on various YOLO11 variants, including YOLO11n, YOLO11s, YOLO11m, YOLO11l, YOLO11x, YOLOv8n, and YOLOv8s, and evaluated their performance in accurately classifying different fire types. The YOLOv11 models demonstrate a strong performance across all fire classes. YOLO11m achieves a strong macro average for precision, recall, and F1 score of 0.96, 0.97, and 0.97, respectively. YOLO11n and YOLO11s have an accuracy of 0.97, but show slightly lower performance for specific classes, particularly Class B and Class C. YOLO11l demonstrates high performance with a macro-average precision, recall, and F1 score of 0.97. YOLO11x exhibits minor drops in precision and recall, especially in Class B and Class F, resulting in macro averages of 0.93, 0.94, and 0.93, for precision, recall, and F1 score respectively.

Future efforts will focus on expanding the fire dataset to improve model accuracy across a wider range of fire classes. To address dataset imbalance, we plan to include more samples from classes that have fewer images, such as chemical and oil-based fires, which will help other models classify these types more accurately. Moreover, developing more advanced deep-learning models for complex fire scenarios will be essential for fire classification and detection. To enhance the interpretability of the models, future work could explore the integration of explainability frameworks. These frameworks help to analyze and identify the characteristics of observations that were more challenging to classify, offering insights into potential areas for the hybrid model's improvement and refinement.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available in [Kaggle] at [https://www.kaggle.com/datasets/imankhammash/classesoffire] (1 December 2024). These data were derived from the resources available in the following: [27–36].

Acknowledgments: The researcher would like to acknowledge the Deanship of Graduate Studies and Scientific Research, Taif University, Saudi Arabia, for funding this work. She also acknowledges the use of ChatGPT and Grammarly for improving the manuscript's syntax and grammar, thereby enhancing its readability.

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

Appendix A

The hybrid model trained on the fire dataset [27,31] involves classes of solid combustible materials, electrical-based fire, chemical fire, oil-based fire, and no-fire. The results are shown in Table A1.

Table A1. Evaluation of the hybrid model on the fire dataset [27,31].

Test Results of Fire Dataset [25,29]			
	Precision	**Recall**	**F1 score**
Class A	0.98	0.95	0.96
Class B	1.00	1.00	1.00
Class C	0.90	0.95	0.92
Class F	0.94	0.94	0.94
No Fire	0.95	0.96	0.95
accuracy	0.96		
Macro avg	0.95	0.96	0.96
Weighted avg	0.96	0.96	0.96

The confusion matrix for this dataset is shown in Figure A1.

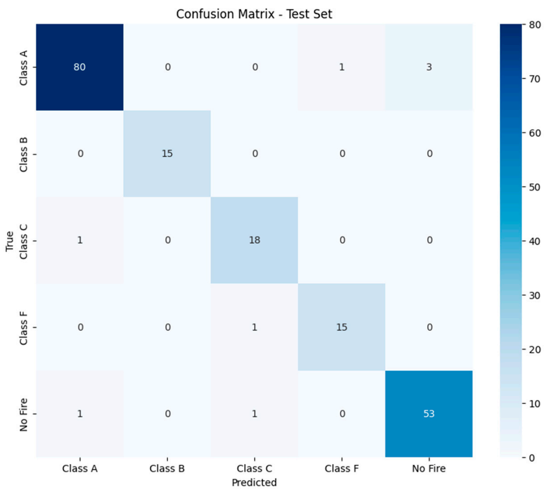


Figure A1. The confusion matrix for the dataset [27,31], generated using the hybrid model.

As can be seen in Figure A1, the hybrid model demonstrates high performance in Class A and No Fire, with minimal misclassifications. Class B achieves perfect classification without any errors. However, minor misclassifications are observed for Class C and Class F, indicating overlap with other classes.

The best model is used to evaluate the following four datasets: The DFAN dataset [22] includes images of indoor fire classes and outdoor fire classes. The forest fire dataset [29] includes a variety of images from wildfires and bushfires. The forest smoke and fire dataset [30] includes images of forest fires captured during daytime, dusk, and nighttime. The Bowfire dataset [32], which consists of images with various resolutions, is divided into two categories: fire and no fire. The results are shown in Table A2, including performance on the right side, which resulted from the direct use of the best model on these different datasets. The results are low because the weights of the best model were not used completely, especially for the classification layer. The output dimensions of the classification layer in the trained model (five classes) do not match the number of classes in the other datasets (two or three classes). Therefore, the model is fine-tuned for five epochs. The model is trained for five epochs while only the last layer is updated. Only the last layer is unfrozen and trained, while all other layers remain frozen, to retain their pre-trained weights. After fine-tuning, the performance is evaluated. The results after fine-tuning show significant improvement, as presented on the left side of Table A2.

Table A2. The best model to evaluate four datasets.

DFAN Dataset						
	Fine-Tuning Results			**Non-Fine-Tuning Results**		
	Precision	**Recall**	**F1 Score**	**Precision**	**Recall**	**F1 Score**
Class A	0.78	0.78	0.78	0.22	0.33	0.27
Class C	0.92	0.95	0.94	0.59	0.65	0.62
Class No Fire	1.00	0.93	0.96	0.76	0.35	0.48
Accuracy	0.92			0.50		
Macro avg	0.90	0.88	0.89	0.52	0.45	0.45
Weighted avg	0.92	0.92	0.92	0.56	0.50	0.51
Forest fire dataset						
Class A	1.00	0.94	0.97	0.64	0.96	0.77
Class No Fire	0.95	1.00	0.98	0.92	0.45	0.61
Accuracy	0.97			0.71		
Macro avg	0.98	0.97	0.97	0.78	0.71	0.69
Weighted avg	0.97	0.97	0.97	0.78	0.71	0.69
Forest smoke and fire dataset						
Class A	0.94	1.00	0.97	0.81	0.49	0.61
Class No Fire	1.00	0.90	0.95	0.60	0.87	0.71
Accuracy	0.96			0.67		
Macro avg	0.97	0.95	0.96	0.70	0.68	0.66
Weighted avg	0.97	0.96	0.96	0.71	0.67	0.66
Bowfire dataset						
Class A	0.83	1.00	0.91	0.95	0.72	0.82
Class No Fire	1.00	0.86	0.92	0.81	0.97	0.88
Accuracy	0.92			0.86		
Macro avg	0.92	0.93	0.92	0.88	0.84	0.85
Weighted avg	0.93	0.92	0.92	0.87	0.86	0.85

Figures A2 and A3 present the confusion matrices for the following datasets: (a) DFAN dataset, (b) Forest Fire dataset, (c) Forest Smoke and Fire dataset, and (d) Bowfire dataset. Figure A2 displays the confusion matrices generated using a hybrid model without fine-tuning, while Figure A3 illustrates the results from the same model after fine-tuning. These visualizations not only provide insights into the classification performance across the different datasets, but also highlight the significant improvements in accuracy achieved through fine-tuning. The comparison clearly demonstrates how fine-tuning enhances the model's ability to correctly classify instances, thereby improving overall performance. For instance, in the DFAN dataset after fine-tuning, the hybrid model shows significant improvement in performance. Class C accuracy increased to 36 correct predictions, with only 2 misclassifications, compared to 100 correct and 45 misclassified previously. Class A misclassifications dropped from 40 to 2, and No Fire improved, with 13 correct predictions and only 1 misclassification compared to 31 correct and 29 misclassified previously. These results demonstrate the model's enhanced ability to distinguish between classes after fine-tuning.

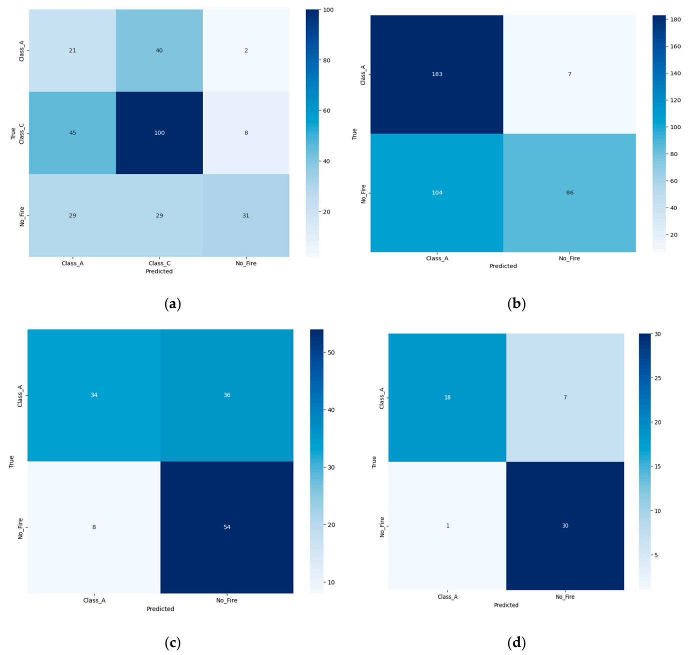


Figure A2. Confusion matrices for the datasets (**a**) DFAN dataset, (**b**) Forest Fire dataset, (**c**) Forest Smoke and Fire dataset, and (**d**) Bowfire dataset. These matrices were generated using a hybrid model without fine-tuning.

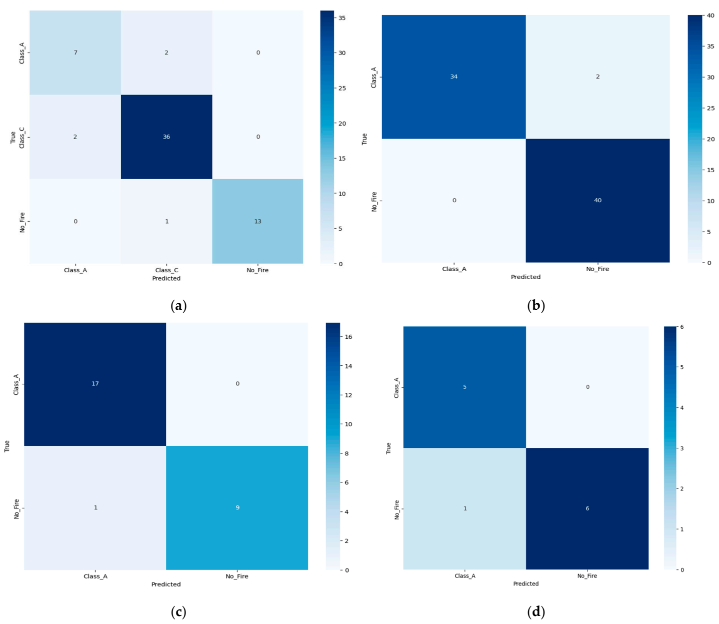


Figure A3. Confusion matrices for the datasets (**a**) DFAN dataset, (**b**) Forest Fire dataset, (**c**) Forest Smoke and Fire dataset, and (**d**) Bowfire dataset. These matrices were generated using a hybrid model with fine-tuning.

References

1. *NFPA 10*; Standard for Portable Fire Extinguishers. NFPA: Quincy, MA, USA, 2018.
2. Pivello, V.R.; Vieira, I.; Christianini, A.V.; Ribeiro, D.B.; da Silva Menezes, L.; Berlinck, C.N.; Melo, F.P.; Marengo, J.A.; Tornquist, C.G.; Tomas, W.M.; et al. Understanding Brazil's catastrophic fires: Causes, consequences and policy needed to prevent future tragedies. *Perspect. Ecol. Conserv.* **2021**, *19*, 233–255. [CrossRef]
3. Ganteaume, A.; Camia, A.; Jappiot, M.; San-Miguel-Ayanz, J.; Long-Fournel, M.; Lampin, C. A review of the main driving factors of forest fire ignition over Europe. *Environ. Manag.* **2013**, *51*, 651–662. [CrossRef] [PubMed]
4. Stephens, S.L. Forest fire causes and extent on United States Forest Service lands. *Int. J. Wildland Fire* **2005**, *14*, 213–222. [CrossRef]
5. NFBA Research: Structure Fires Involving Flammable Gases. Available online: https://www.nfpa.org/education-and-research/research/nfpa-research/fire-statistical-reports/structure-fires-involving-flammable-gases (accessed on 1 November 2024).
6. National-Fire-Protection-Association. US Fire Adminstration Stats. 2024. Available online: https://www.usfa.fema.gov/statistics/ (accessed on 1 November 2024).
7. Gaur, A.; Singh, A.; Kumar, A.; Kumar, A.; Kapoor, K. Video flame and smoke based fire detection algorithms: A literature review. *Fire Technol.* **2020**, *56*, 1943–1980. [CrossRef]
8. Li, P.; Zhao, W. Image fire detection algorithms based on convolutional neural networks. *Case Stud. Therm. Eng.* **2020**, *19*, 100625. [CrossRef]
9. Muhammad, K.; Ahmad, J.; Lv, Z.; Bellavista, P.; Yang, P.; Baik, S.W. Efficient deep CNN-based fire detection and localization in video surveillance applications. *IEEE Trans. Syst. Man Cybern. Syst.* **2018**, *49*, 1419–1434. [CrossRef]
10. Muhammad, K.; Ahmad, J.; Mehmood, I.; Rho, S.; Baik, S.W. Convolutional neural networks based fire detection in surveillance videos. *IEEE Access* **2018**, *6*, 18174–18183. [CrossRef]
11. Yang, Y.; Pan, M.; Li, P.; Wang, X.; Tsai, Y.T. Development and optimization of image fire detection on deep learning algorithms. *J. Therm. Anal. Calorim.* **2023**, *148*, 5089–5095. [CrossRef]
12. Sharma, J.; Granmo, O.C.; Goodwin, M.; Fidje, J.T. Deep convolutional neural networks for fire detection in images. In Proceedings of the Engineering Applications of Neural Networks: 18th International Conference, EANN 2017, Athens, Greece, 25–27 August 2017; pp. 183–193.
13. de Almeida Pereira, G.H.; Fusioka, A.M.; Nassu, B.T.; Minetto, R. Active fire detection in Landsat-8 imagery: A large-scale dataset and a deep-learning study. *ISPRS J. Photogramm. Remote Sens.* **2021**, *178*, 171–186. [CrossRef]
14. Zhang, Q.; Xu, J.; Xu, L.; Guo, H. Deep convolutional neural networks for forest fire detection. In *2016 International Forum on Management, Education and Information Technology Application*; Atlantis Press: Amsterdam, The Netherlands, 2016; pp. 568–575.
15. Jiao, Z.; Zhang, Y.; Xin, J.; Mu, L.; Yi, Y.; Liu, H.; Liu, D. A deep learning based forest fire detection approach using UAV and YOLOv3. In Proceedings of the 2019 1st International conference on industrial artificial intelligence (IAI), Dublin, Ireland, 16–18 October 2019; pp. 1–5.
16. Jin, C.; Wang, T.; Alhusaini, N.; Zhao, S.; Liu, H.; Xu, K.; Zhang, J. Video Fire Detection Methods Based on Deep Learning: Datasets, Methods, and Future Directions. *Fire* **2023**, *6*, 315. [CrossRef]
17. Saydirasulovich, S.N.; Mukhiddinov, M.; Djuraev, O.; Abdusalomov, A.; Cho, Y.I. An improved wildfire smoke detection based on YOLOv8 and UAV images. *Sensors* **2023**, *23*, 8374. [CrossRef] [PubMed]
18. Avazov, K.; Mukhiddinov, M.; Makhmudov, F.; Cho, Y.I. Fire detection method in smart city environments using a deep-learning-based approach. *Electronics* **2021**, *11*, 73. [CrossRef]
19. Kim, B.; Lee, J. A video-based fire detection using deep learning models. *Appl. Sci.* **2019**, *9*, 2862. [CrossRef]
20. Seydi, S.T.; Saeidi, V.; Kalantar, B.; Ueda, N.; Halin, A.A. Fire-Net: A Deep Learning Framework for Active Forest Fire Detection. *J. Sens.* **2022**, *2022*, 8044390. [CrossRef]
21. Abdusalomov, A.B.; Islam, B.M.S.; Nasimov, R.; Mukhiddinov, M.; Whangbo, T.K. Abdusalomov. An improved forest fire detection method based on the detectron2 model and a deep learning approach. *Sensors* **2023**, *23*, 1512. [CrossRef]
22. Yar, H.; Hussain, T.; Agarwal, M.; Khan, Z.A.; Gupta, S.K.; Baik, S.W. Optimized dual fire attention network and medium-scale fire classification benchmark. *IEEE Trans. Image Process.* **2022**, *31*, 6331–6343. [CrossRef]
23. Xue, Q.; Lin, H.; Wang, F. Fcdm: An improved forest fire classification and detection model based on yolov5. *Forests* **2022**, *13*, 2129. [CrossRef]
24. Khan, S.; Khan, A. Ffirenet: Deep learning based forest fire classification and detection in smart cities. *Symmetry* **2022**, *14*, 2155. [CrossRef]
25. Akagic, A.; Buza, E. Lw-fire: A lightweight wildfire image classification with a deep convolutional neural network. *Appl. Sci.* **2022**, *12*, 2646. [CrossRef]
26. Lee, J.; Kang, J.; Park, C.S.; Jeong, J. Distributed Fire Classification and Localization Model Based on Federated Learning with Image Clustering. *Appl. Sci.* **2024**, *14*, 9162. [CrossRef]
27. Refaee, E.A.; Sheneamer, A.; Assiri, B. A Deep-Learning-Based Approach to the Classification of Fire Types. *Appl. Sci.* **2024**, *14*, 7862. [CrossRef]

28. Park, M.; Tran, D.Q.; Lee, S.; Park, S. Multilabel image classification with deep transfer learning for decision support on wildfire response. *Remote Sens.* **2021**, *13*, 3985. [CrossRef]
29. Khan, A.; Hassan, B.; Khan, S.; Ahmed, R.; Adnan, A. Available online: https://www.kaggle.com/datasets/alik05/forest-fire-dataset (accessed on 1 November 2024).
30. Yang, J.; Zhu, W.; Sun, T.; Ren, X.; Liu, F. Lightweight forest smoke and fire detection algorithm based on improved YOLOv5. *PLoS ONE* **2023**, *18*, e0291359. [CrossRef] [PubMed]
31. Refaee, E. Fire Type Classes Dataset. 2024. Available online: https://zenodo.org/records/13119922?preview=1&token=eyJhbGciOiJIUzUxMiJ9.eyJpZCI6ImI0NjA1OWRlLWM1ZjctNDM0NS04Mjk0LTA1OGViNzA1Zjg2MCIsImRhdGEiOnt9LCJyYW5kb20iOiIyZDE3NjBlMTQ4YzNmZDQ0MmQzMTZiODgzZThmMzk5OCJ9.q7irvvmF2Ecs6u1hSOlz8Ny2QgHp1FksziQCFCrgcawqkwrDi47vVkRmksXMuLwNpkz0pi4AtzCA489aAR76Gg (accessed on 1 November 2024).
32. Chino, D.Y.; Avalhais, L.P.; Rodrigues, J.F.; Traina, A.J. Bowfire: Detection of fire in still images by integrating pixel color and texture analysis. In Proceedings of the 2015 28th SIBGRAPI Conference on Graphics, Patterns and Images, Salvador, Brazil, 26–29 August 2015; pp. 95–102.
33. Hazard Control Technologies. Available online: https://www.youtube.com/@hazardcontroltechnologies (accessed on 1 November 2024).
34. Fire at Fast Food Resturant. Safety Video. Available online: https://www.youtube.com/@safetyvideos9453 (accessed on 1 November 2024).
35. Firemill Extinguisher. Kitchen Fire. Available online: https://www.youtube.com/watch?v=WpTGTwzQtOs (accessed on 1 November 2024).
36. Cofessco Fire Protection. Class K Fire Extinguishers vs. Other Agents: Cofessco Fire Protection. Available online: https://www.youtube.com/watch?v=X4GPxV_2vV0&t=461s (accessed on 1 November 2024).
37. Alif, M.A.R. YOLOv11 for Vehicle Detection: Advancements, Performance, and Applications in Intelligent Transportation Systems. *arXiv* **2014**, arXiv:2410.22898.
38. Paul, P.T. YOLO11: The Cutting-Edge Evolution in Object Detection—A Brief Review of the Latest in the Yolo Series. Available online: https://medium.com/@paultsoi1014/yolo11-the-cutting-edge-evolution-in-object-detection-a-brief-review-of-the-latest-in-the-yolo-19065729edfe (accessed on 1 November 2024).
39. Jegham, N.; Koh, C.Y.; Abdelatti, M.; Hendawi, A. Evaluating the Evolution of YOLO (You Only Look Once) Models: A Comprehensive Benchmark Study of YOLO11 and Its Predecessors. *arXiv* **2024**, arXiv:2411.00201.
40. Howard, A.G. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.