Article

An Explainable AI-based Modified YOLOv8 Model for Efficient Fire Detection

Md. Waliul Hasan , Shahria Shanto , Jannatun Nayeema , Rashik Rahman , Tanjina Helaly * , Ziaur Rahman , Sk. Tanzir Mehedi *

Posted Date: 4 September 2024

doi: 10.20944/preprints202409.0372.v1

Keywords: fire detection; modified YOLOv8; deep learning; computer vision; explainable artificial intelligence; EigenCAM

Article

An Explainable AI-based Modified YOLOv8 Model for Efficient Fire Detection

Md. Waliul Hasan [1], Shahria Shanto [1], Jannatun Nayeema [1], Rashik Rahman [1,2], Tanjina Helaly [1,*], Ziaur Rahman[3], Sk. Tanzir Mehedi [1,3,*]

[1] Department of Computer Science and Engineering, University of Asia Pacific, Dhaka, Bangladesh
[2] Department of Computer Science, University of Calgary, Alberta, Canada
[3] School of Computer Science, Queensland University of Technology, Brisbane, Australia
* Correspondence: tanjina@uap-bd.edu (T.H.); s.tanzir@qut.edu.au (S.T.M.).

Abstract: Fire detection is crucial for ensuring human safety and minimizing property damage. By utilizing advanced technologies, we can identify fires early, before they escalate. Autonomous fire detection systems are particularly vital in high-risk areas with minimal human presence. To address this need, we propose an automated fire-detection model based on YOLOv8. We modified the architecture of YOLOv8 and achieved impressive performances. Through comprehensive analysis of different YOLO models and their predecessors, we identified and addressed substantial gaps in this field. Also, by augmenting the framework with additional capabilities, we surpassed previous models, enabling real-time fire detection. Our proposed model achieves impressive performances, with 98% accuracy in fire detection and 97.8% in smoke detection. Moreover, recall rates for fire and smoke are 97.1% and 97.4%, respectively, with a mean average precision (mAP) accuracy of 99.1% for both cases. Finally, we applied an explainable artificial intelligence technique to interpret our proposed model and its results. This study lays the foundation for future research aimed at enhancing fire detection efficiency. The reliability of our modified model presents promising opportunities for future advancements, ensuring increased effectiveness and precision in efficient fire detection.

Keywords: fire detection; modified YOLOv8; deep learning; computer vision; explainable artificial intelligence; EigenCAM

MSC: 68T07

1. Introduction

Fire incidents pose a substantial global threat, capable of causing widespread devastation once they ignite. These destructive events endanger lives and property worldwide. When a fire begins, it can quickly escalate, leading to severe destruction. Beyond the immediate threats to human life and property, fires also have long-term environmental consequences, including deforestation, air pollution, and the release of toxic chemicals. The economic impact is equally significant, with billions of dollars lost each year due to fire-related damages [1]. Fires can also disrupt communities, leading to prolonged displacement of residents, loss of livelihoods, and significant psychological trauma [2]. The cumulative effects of fire incidents are not limited to the immediate aftermath but can have repercussions that persist for decades, affecting future generations and the stability of entire regions [1,3]. Given the unpredictable nature and rapid escalation of fires, early detection is vital. The ability to identify and respond to fires in their earliest stages can make the difference between a minor incident and a major disaster. Addressing this challenge requires the development of advanced, highly accurate monitoring systems capable of early fire detection to minimize damage. Traditional sensor-based methods, such as smoke detectors and heat sensors, often prove inadequate, as they typically respond too late or fail to detect fires in their initial stages, especially in large or open areas [4]. Additionally, these methods are limited by their inability to cover vast spaces or detect fires in complex environments like forests or industrial settings, where early detection is most critical.

Although image-based systems hold promise, existing models often lack the necessary precision in critical situations [5]. These systems, despite their potential, are hindered by limitations such as low detection accuracy under varying lighting conditions, difficulty distinguishing between fire and non-fire objects, and challenges in efficiently processing real-time data [6]. In high-stakes scenarios where every second is crucial, these shortcomings can have devastating consequences. The risk of false alarms is particularly problematic, as they can desensitize response teams, leading to delays in addressing real emergencies [7]. Moreover, in areas where fire detection is crucial, such as densely populated urban centers or remote forest regions, the failure of these systems can result in catastrophic outcomes. The increasing threat posed by fires, combined with the limitations of current detection technologies, highlights the need for a more advanced solution. As climate change intensifies and urbanization continues to expand, the frequency and severity of fire incidents are expected to rise. This reality underscores the urgency of developing a fire detection system that is not only accurate but also robust enough to operate effectively in diverse environments. In response to this critical need, we have developed a state-of-the-art deep learning approach tailored to meet these demands. By utilizing computer vision, our solution offers a more precise and reliable method of fire detection, ensuring that fires are identified and managed before they cause significant harm.

Previous research has explored various frameworks such as YOLOv3, YOLOv5, R-CNN, vanilla CNN, and dual CNN models for fire detection [2]. However, these models often faced challenges related to accuracy and speed, particularly in high-pressure situations [2,8]. Several studies have pointed out the limitations of these fire detection models. For instance, the YOLOv3-based model exhibited a tendency towards false positives, often misidentifying electrical lamps as fires, especially during nighttime conditions [2]. Similarly, the YOLOv2-based model also experienced issues with false positives, particularly in complex environments [1]. Additionally, the YOLOv4-based model was found to be less effective when applied to larger areas [4]. These models also struggled to adapt to new and unforeseen scenarios, which is crucial for real-world fire detection applications. Furthermore, some CNN-based models encountered difficulties in generalizing fire detection across new situations, often resulting in a high rate of false alarms [9,10]. False alarms not only waste valuable resources but can also lead to "alarm fatigue," where responders become desensitized to alarms, potentially overlooking genuine emergencies. Moreover, dual CNN-based detection systems exhibited inconsistent performance depending on the environment [11,12]. These inconsistencies made it challenging to rely on these models in critical situations where a high degree of precision and reliability is necessary. Furthermore, CNN-based technology with aerial 360-degree cameras mounted on UAVs to capture unlimited field of view images also had drawbacks. UAVs were affected by weather conditions, limiting their effectiveness and the system was not capable of detecting wildfires at night [3]. In other studies, some researchers utilized a multifunctional AI framework and the Direct-MQTT protocol to enhance fire detection accuracy and minimize data transfer delays. This approach also applied a CNN algorithm for visual intelligence and used the Fire Dynamics Simulator for testing. However, it did not consider sensor failures and used static thresholds [13]. In another study, researchers utilized the ELASTIC-YOLOv3 algorithm to quickly and accurately detect fire candidate areas and combined it with a random forest classifier to verify fire candidates. Additionally, they used a temporal fire-tube and bag-of-features histogram to reflect the dynamic characteristics of nighttime flames. However, the approach faced limitations in real-time processing due to the computational demands of combining CNN with RNN or LSTM and it struggled with distinguishing fire from fire-like objects in nighttime urban environments [14]. The Intermediate Fusion VGG16 model and the Enhanced Consumed Energy-Leach protocol were utilized in a study for early detection of forest fires. Drones were employed to capture RGB and IR images, which were then processed using the VGG16 model. However, the study faced limitations due to the lack of real-world testing and resource constraints that hindered comprehensive evaluation [5]. In another computer vision-based study, researchers utilized a YOLOv5 fire detection algorithm based on an attention-enhanced ghost model, mixed convolutional pyramids, and flame-center detection. It incorporated Ghost bottlenecks, SECSP attention modules, GSConv convolution, and the SIoU loss function to enhance accuracy. However, the limitations included potential challenges in real-time detection due to high computational complexities and the need for further validation in diverse environments [6]. In a different study that was based on CNN, the researchers modified the CNN for forest fire recognition, integrating transfer learning and a feature fusion algorithm to enhance detection accuracy. The researchers had utilized a diverse dataset of fire and non-fire images for training and testing. However, the study had faced limitations due to the small sample size and the need for further validation in real-world scenarios to ensure robustness and generalizability. In a different study to detect fire and smoke, the researchers have utilized a capacitive particle-analyzing smoke detector for very early fire detection, employing a multiscale smoke particle concentration detection algorithm. This method involved capacitive detection of cell structures and time-frequency domain analysis to calculate particle concentration. However, the study had faced limitations in distinguishing particle types and had struggled with false alarms in complex environments.

In our pursuit of a more reliable solution, we turned to the robust capabilities of YOLOv8, a model renowned for its superior object detection abilities. Our goal was to significantly enhance fire detection performance by optimizing the architecture of YOLOv8 to better identify fire-specific visual cues. Through extensive training on a comprehensive fire image dataset, the modified YOLOv8 model has demonstrated improved precision, recall, mAP@50, and F1 score. This advanced model excels at detecting not only flames but also smoke, which is often an early indicator of larger fires, thereby providing an early warning system that can prevent a small incident from escalating into a full-blown disaster.

To further enhance the interpretability of our YOLOv8 model, we utilized EigenCAM, a technique within explainable AI that generates class activation maps. EigenCAM allows us to visualize the regions in the input images that the model focuses on when making predictions [15,16]. By highlighting these areas, we gain insights into which features the model considers critical for detecting fire and smoke. This transparency is essential for building trust in AI-based systems, particularly in crucial applications like fire detection. With these technologies in place, the issue of false alarms triggered by orange hues or steam can be effectively mitigated. Our approach not only aims to reduce false positives but also ensures that fire detection remains accurate and reliable across a wide range of environments and scenarios. The development of this cutting-edge deep learning solution marks a significant leap forward in the field of fire detection. By capitalizing on the strengths of YOLOv8 and enhancing its capabilities with EigenCAM, we have created a system that is both highly accurate and interpretable. This innovative approach addresses the shortcomings of previous models and provides a robust tool for early fire detection, which is crucial for minimizing damage and safeguarding lives and property globally. This breakthrough paves the way for ultra-reliable fire monitoring in homes, industries, and beyond. The modified YOLOv8 fire detection technique is ushering in a new era of fire safety innovations. It is crucial to take a stand against fire devastation. Together, armed with smarter AI, we can make a real difference, saving lives and property while forging a safer future. The main contributions of this research are as follows:

- In this work, a modified YOLOv8 architecture is proposed for the development of an efficient automated fire and smoke detection system, achieving 98% accuracy in fire and 97.8% accuracy in smoke detection.
- A comprehensive fire dataset has been created, consisting of 4,301 labeled images. Each image has been carefully annotated to ensure accuracy, providing a valuable resource for improving the performance of models in identifying fire and smoke in various conditions.
- EigenCAM is employed to explain and visualize the results of the proposed model, highlighting the image areas that most influenced the model's decisions. This approach enhances the understanding of the model's behavior, thereby improving the interpretability and transparency of its predictions.

The rest of this research paper is organized as follows. First of all, Section 2 discusses the background and related work in this field. Additionally, we present the methodology and a detailed description of the dataset in Section 3. Experimental results analysis of these methods are discussed in Section 4. In Section 5, we introduce the explainable AI to interpret our proposed model and its results. Finally, Section 6 and 7 includes the discussion, summary and feasible future directions for this research.

2. Related Works

Deep learning-based methods for fire and smoke detection in smart cities have been receiving increasing attention recently. Hybrid techniques that incorporate multiple deep-learning algorithms for fire detection have been proposed in several studies.

Ahn et al. [17] introduce a fire detection model leveraging computer vision-based Closed-Circuit Television (CCTV) for early fire identification in buildings, employing the YOLO algorithm. The proposed model demonstrates an overall accuracy with recall, precision, and mAP@0.5 performances of 0.97, 0.91, and 0.96, respectively. Notably, the findings indicate the ability to detect a fire within 1 second of the maximum visible range of the CCTV, revealing a time discrepancy of up to 307 seconds when comparing the fire detection times of the Early Fire Detection Model (EFDM) and conventional fire detectors. The paper has also recommended future research directions, emphasizing the development of a more robust and accurate model through the incorporation of additional features.

Pincott et al. [8] propose a vision-based indoor fire and smoke detection system using computer vision techniques. The study has explored and adopted existing models based on Faster R-CNN Inception V2 and SSD MobileNet V2 models. They have evaluated the performance results of Faster R-CNN (Model A) with an average accuracy of 95% for fire detection and 62% for smoke detection, while SSD (Model B) has an average accuracy of 88% for fire detection and 77% for smoke detection. The limitation of this methodology is that it does not evaluate the proposed approach in outdoor environments, which could be a potential area for future research.

Avazov et al. [4] propose a deep-learning-based fire detection method for smart city environments. The authors have developed a novel convolutional neural network to detect fire regions using an enhanced YOLOv4 network. The overall fire detection accuracy is 98.8%. The proposed method aims to improve fire safety in society using emerging technologies, such as digital cameras, computer vision, artificial intelligence, and deep learning. However, the method requires digital cameras and a Banana Pi M3 board, which may not be available in all settings, and may not be suitable for detecting fires in large areas, such as forests or industrial sites. Additionally, the method may struggle to detect fires that are hidden or occur in areas not visible to digital cameras.

Abdusalomov et al. [2] propose fire detection and classification methods using the YOLOv3 algorithm for surveillance systems. The study focuses on enhancing the accuracy and efficiency of fire detection algorithms for real-time monitoring, achieving an overall fire detection accuracy of 98.9%. The limitations of the proposed method include the possibility of errors in detecting electrical lamps as real fires, especially at night.

Muhammad et al. [18] propose a fire detection system based on convolutional neural networks (CNN) for effective disaster management. They have used two datasets: Foggia's video dataset and Chino's dataset. Foggia's dataset consists of 31 videos with both indoor and outdoor environments, of which 14 videos contain fire and the remaining 17 videos do not. The overall accuracy of the proposed fire detection scheme is 98.5%, which is higher than state-of-the-art methods. Future work suggested in this paper includes maintaining a balance between accuracy and false alarms.

Khan Muhammad [9] implement a cost-effective CNN architecture for fire detection in surveillance videos, inspired by GoogleNet1. The goal is to balance computational complexity and accuracy for real-time. Their proposed model is based on GoogleNet, which is fine-tuned for fire detection. Pre-trained weights from GoogleNet are used and further fine-tuned for the specific task. The proposed model achieves an accuracy of 94.43%. The model has reduced the false alarm to 5.4%. The model needs further tuning to handle both smoke and fire detection in more complex real-world scenarios.

Byoungjun Kim and Joonwhoan Lee [19] propose a deep learning-based method for detecting fires using video sequences. The goal is to improve accuracy and reduce false alarms compared to traditional methods. Their research has utilized a faster region-based convolutional neural network to identify suspected regions of fire and non-fire objects. The Long Short-Term Memory networks accumulate features over successive frames to classify fire presence. The research combines short-term decisions using majority voting for a final long-term decision. The proposed method significantly improves the fire detection accuracy by reducing false detections and misdetections and also successfully interprets the temporal behavior of flames and smoke, providing detailed fire information. The model had some false positives still occur, especially with objects like clouds, chimney smoke and sunsets.

Yakun Xia et al. [20] introduce a method to enhance fire detection accuracy in videos. The authors combine motion-flicker-based dynamic features with deep static features extracted using an adaptive lightweight convolutional neural network. By analyzing motion and flicker differences between fire and other objects and integrating these with deep static features, the method improves detection accuracy and reduces false alarms. Tested on three datasets, it outperformed state-of-the-art methods in terms of accuracy and run time, proving effective even in complex video scenarios and on resource-constrained devices. However, the paper notes challenges in achieving high robustness in complex scenarios and highlights the need for further research on fire spread prediction and spatial positioning.

Pu Li and Wangda Zhao [21] introduce advanced methods for detecting fires in images using convolutional neural networks. The researchers aim to enhance the accuracy and speed of fire detection by comparing four CNN-based object detection models: Faster-RCNN, R-FCN, SSD and Yolo v3. They utilize a self-built dataset containing 29,180 images of fire, smoke and disturbances and apply transfer learning with pre-trained networks on the COCO dataset. The results show that YOLO v3 achieves the highest average precision of 83.7 and the fastest detection speed of 28 FPS, making it the most robust among the tested models. However, the study also highlight some limitations, such as difficulties in detecting fires in complex scenes and small fire and smoke regions, as well as the significant computational costs of some methods, which can affect detection speed.

Sergio Saponara et al. [1] present a real-time video-based fire and smoke detection system using the YOLOv2 Convolutional Neural Network for antifire surveillance. Designed for low-cost embedded devices like the Jetson Nano and standard CCTV cameras, the system employs a lightweight neural network architecture for real-time processing. The model is trained offline with diverse indoor and outdoor fire and smoke image sets, with ground truth data generated using a labeling app. The system demonstrates high detection rates, low false alarm rates and fast processing speeds, achieving 93 accuracy in validation tests and 96.82 accuracy in experimental results. However, it faces limitations such as a small training dataset of 400 images which may affect generalization and false positives due to challenging features like clouds and sunlight and its performance is dependent on the capabilities of the embedded device used.

Yakhyokhuja Valikhujaev et al. [11] introduce a novel approach for detecting fire and smoke using deep learning techniques, specifically a convolutional neural network with dilated convolutions. The architecture of the CNN with dilated convolutions enhances feature extraction and reduces false alarms. The results demonstrate that this method outperformed other state-of-the-art architectures in terms of classification performance and complexity and it is effectively generalized to unseen data, minimizing false alarms. However, the method has limitations, such as potential errors when fire and smoke pixel values are similar to the background, particularly in cloudy weather, and its dependency on the custom dataset, which may affect its generalizability to other datasets.

Dali Sheng et al.[22] focus on developing a robust method for detecting smoke in complex environments. The authors combine simple linear iterative clustering for image segmentation and density-based spatial clustering of applications with noise for clustering similar super-pixels, which are then processed by a convolutional neural network to distinguish smoke from non-smoke features. The results show improved accuracy and reduced false positives compared to traditional methods.

However, the method's high sensitivity lead to a slight increase in false positives, indicating a need for further enhancements to ensure robustness in real-world applications.

Gaohua Lin and Yongming Zhang [23] focus on early smoke detection in video sequences to enhance fire disaster prevention. The authors develope a joint detection framework that combines faster RCNN for locating smoke targets and 3D CNN for recognizing smoke. They utilize various data augmentation techniques and preprocessing methods like optical flow to improve the training dataset. The framework achieves a detection rate of 95.23% and a low false alarm rate of 0.39%, significantly outperforming traditional methods. However, the study faced limitations such as data scarcity, which can lead to overfitting and the need for more realistic synthetic smoke videos to enhance training performance.

Arun Singh Pundir et al. [12] introduce a method for early and robust smoke detection using a dual deep learning framework. This framework leverages image-based features extracted by Deep Convolutional Neural Networks (CNN) from smoke patches identified by a superpixel algorithm and motion-based features captured using optical flow methods, also processed by CNN. The method achieves impressive accuracy rate of 98.29% for nearby smoke detection, 91.96% for faraway smoke detection and an average accuracy of 97.49% across various scenarios. However, the method faces challenges in distinguishing smoke from similar non-smoke conditions like clouds, fog and sandstorms.

Yunji Zhao et al. [24] introduce the target awareness and depthwise separability algorithms for early fire smoke detection, which are vital for early warning systems. The authors utilize pre-trained convolutional neural networks to extract deep feature maps for modeling smoke objects and employ depthwise separable convolutions to enhance the algorithm's speed, making real-time detection feasible. The results demonstrate that the proposed algorithm can detect early smoke in real-time and surpasses current methods in both accuracy and speed. However, the study notes that detection accuracy decreases in certain video sequences under varying conditions and some information might be lost due to the depthwise separable method.

Pedro et al. [10] present an automatic fire detection system using deep convolutional neural networks (CNNs) tailored for low-power, resource-constrained devices like the Raspberry Pi 4. The objective is to create an efficient fire detection system by reducing computational costs and memory usage without sacrificing performance. The authors employ filter pruning to eliminate less important convolutional filters and utilize the YOLOv4 algorithm for fire and smoke detection, optimizing it through various pruning techniques. The results show a significant reduction in computational cost (up to 83.60) and memory consumption (up to 83.86), while maintaining high predictive performance. However, the system faces limitations in generalizing to new environments and dealing with high false alarm rates, especially under varying lighting conditions and camera movements.

Naqqash Dilshad et al. [25] introduce E-FireNet, a deep learning framework aimed at real-time fire detection in complex surveillance settings. The authors modify the VGG16 network by removing Block-5 and adjusting Block-4, using smaller convolutional kernels to enhance detail extraction from images. The framework operates in three stages: data preprocessing, fire detection and alarm generation. E-FireNet achieves impressive results with an accuracy of 0.98, 1 precision, 0.99 recall, and 0.99 F1-score, outperforming other models in accuracy, model size, and execution time, achieving 22.17 FPS on CPU and 30.58 FPS on GPU. However, the study notes the need for a more diverse dataset and suggests future research to expand the dataset and explore the use of vision transformers for fire detection.

Yunsov et al. [26] introduce a YOLOv8 model with transfer learning to detect large forest fires. For smaller fires, they use the TranSDet model, also leveraging transfer learning. Their approach achieves a total 97% accuracy rate. However, the model faces challenges in detecting fire when only smoke is present and occasionally misidentifies the sun and electric lights as fire. Table 1 shows a summary of related works on fire detection.

Table 1. Summary of related works on fire detection.

Ref.	Algorithm	Accuracy	Limitations	Future Directions
[17]	YOLO	0.97 (Recall), 0.91 (Precision)	Limited to CCTV range	Incorporate temperature and sound
[8]	R-CNN, SSD	95% (Fire), 62% (Smoke)	Not tested outdoors	Test in outdoor environments
[4]	YOLOv4	98.8%	Not suitable for large areas	Apply to larger areas
[2]	YOLOv3	98.9%	Errors with electrical lamps	Improve detection at night
[18]	CNN	98.5%	Balance between accuracy and false alarms	Optimize accuracy vs. false alarms
[9]	CNN	94.43%	High false alarms	Improve tuning for real-world scenarios
[20]	CNN	97.94% (Best)	Issues with multiple moving objects	Integrate with IoT for real-time detection
[21]	CNN	83.7%	Lower accuracy, delayed alarms	Improve detection accuracy and speed
[1]	YOLOv2	96.82%	False positives in challenging environments	Connect with cloud facilities
[11]	CNN	99.53%	Early detection issues in clouds	Develop lightweight, robust model
[22]	SLIC-DBSCAN	87.85%	High false positives	Improve sensitivity, reduce FPR
[23]	R-CNN, 3D CNN	95.23%	Small training and validation dataset	Collect more diverse dataset
[12]	CNN	97.49%	Affected by clouds, fog	Extend to real-time fire detection
[24]	Depthwise	93.98%	Reduced accuracy in varying conditions	Balance speed and accuracy
[10]	CNN	mAP 73.98%, F1 0.724	Generalization issues, false alarms	Explore layer pruning
[19]	R-CNN, LSTM	mAP 88.3%, Smoke 87.5%	False detection with non-fire objects	Enhance dataset, improve accuracy
[25]	E-FireNet	Acc 0.98, F1 0.99	Limited, monotonous datasets	Expand dataset, improve generalization
[26]	YOLOv8 and TranSDet	Acc 97%, F1 96.3%	Occasionally misidentifies the sun and electric lights as fire	Expand dataset to address the limitations.

3. Methodology

Fires can cause significant harm and substantial damage to both people and their properties. An efficient fire detection system is crucial to reducing these dangers by facilitating quicker response times. To create an advanced computer vision model for this purpose, we undertook a series of systematic steps.

Initially, we have gathered and labeled a robust dataset, enhancing it further through data augmentation techniques to ensure its comprehensiveness. Our initial training phase employs the standard YOLOv8 [27] architecture, serving as our baseline. We then delve into hyperparameter tuning, exploring a variety of parameters to identify the optimal set for peak performance. This involves multiple training iterations, each time tweaking the hyperparameters to achieve the best results. With the optimal hyperparameters identified, we retrain our model using a modified version of the YOLOv8 architecture, incorporating the previously trained weights from the initial model. This step is crucial for leveraging the learned features from the earlier training phase, a process known as deep transfer learning. Further refinement is achieved through additional training phases, continuously using the improved weights and our adjusted architecture. To ensure transparency in the model's decision-making process, we apply EigenCAM, enbling us to visualize and interpret how the model makes its predictions. This detailed methodology, which is visually summarized in Figure 1, outlines our approach to developing an effective fire detection system.

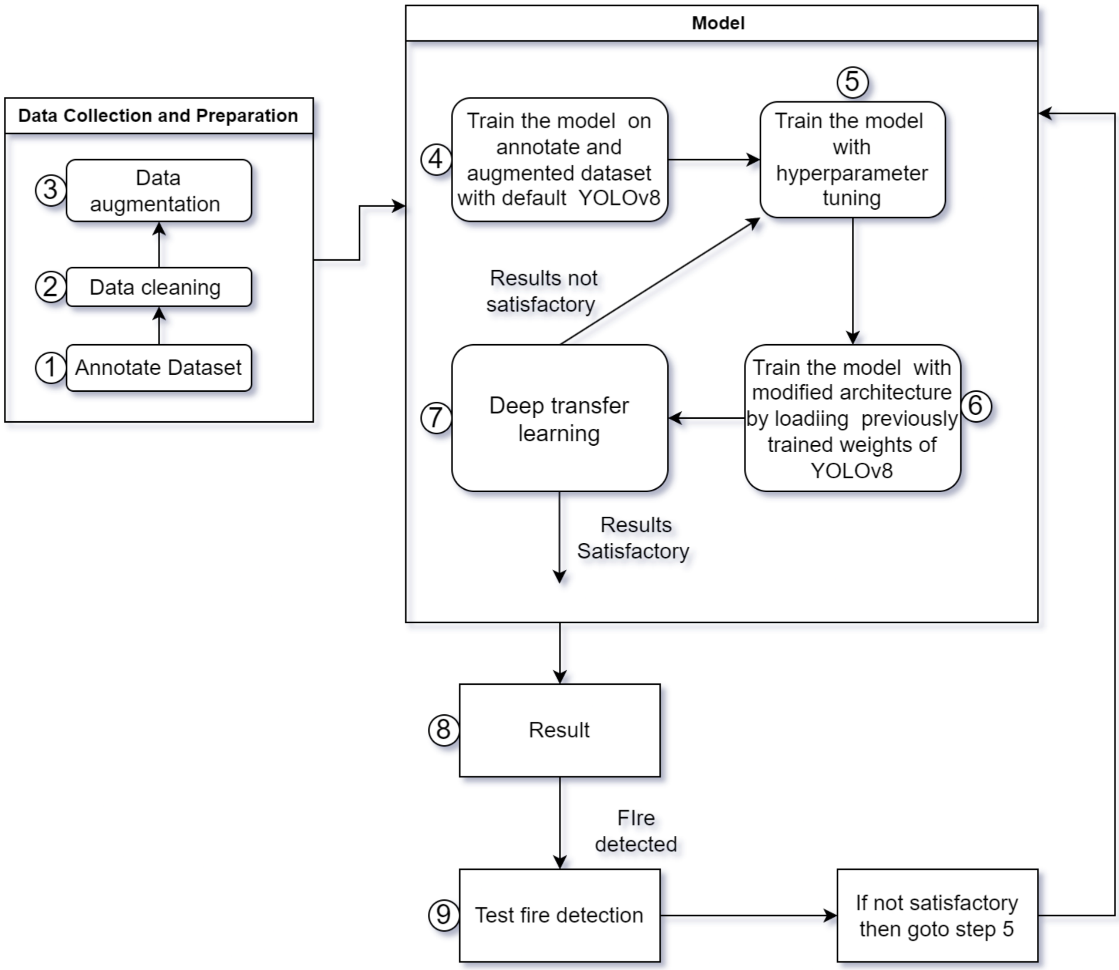

Model
Data Collection and Preparation
3 Data augmentation
2 Data cleaning
1 Annotate Dataset
4 Train the model on annotate and augmented dataset with default YOLOv8
5 Train the model with hyperparameter tuning
Results not satisfactory
7 Deep transfer learning
6 Train the model with modified architecture by loadiing previously trained weights of YOLOv8
Results Satisfactory
8 Result
FIre detected
9 Test fire detection
If not satisfactory then goto step 5

Figure 1. Detailed workflow of the proposed fire detection model development using YOLOv8.

3.1. Dataset Description

The initial dataset we collected consists of 4301 labeled fire and smoke images. To further enrich the dataset, we have added a curated selection of images showcasing different types of fire, sourced from reputable online platforms. The bounding boxes in the CVAT tool are labeled with the names "fire" and "smoke." Subsequently, the dataset is randomly partitioned into two separate sets, one designated for training purposes (80%) and the other allocated for validation (20%). Figure 2 includes several fire and smoke images that are clearly annotated, providing precise labels that highlight the specific areas associated with fire and smoke.

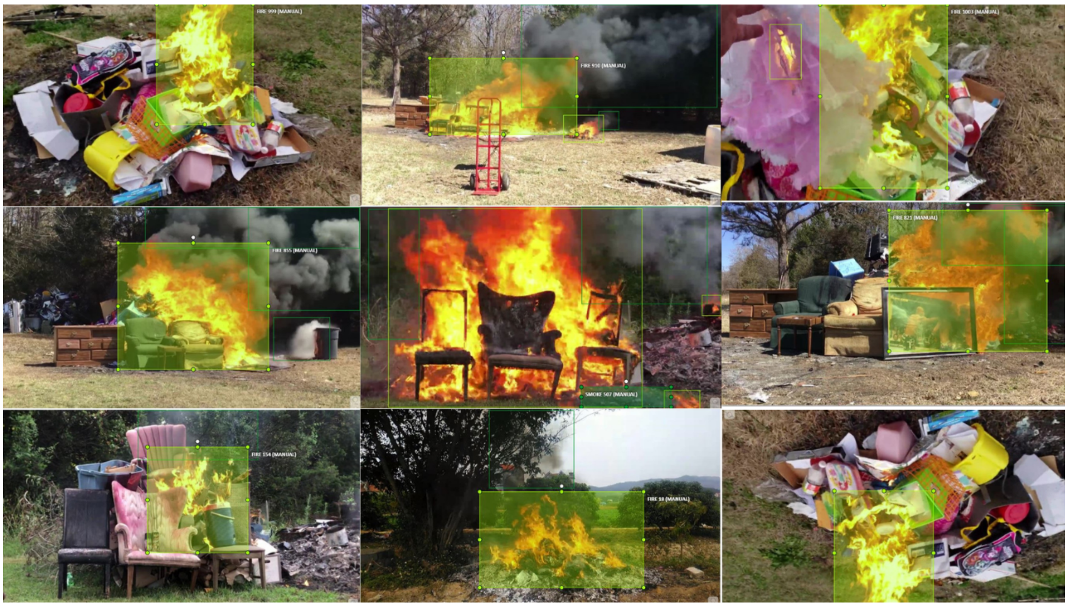


Figure 2. Examples of labeled fire and smoke images from the dataset.

3.2. Dataset Augmentation and Replication

The original dataset has extensive variation in terms of smoke and fire types and settings-ranging from outdoor and indoor fires, large and small, to bright and low-light conditions. After splitting the dataset and copying all images and labels, we halt the replication process to further enhance the training and validation sets. Specifically, we duplicate existing images and labels to increase the size of the relevant dataset portions. The goal is to make the dataset more diverse, enabling the model to learn from a wider range of examples. The original dataset contains 4,301 images, with 3,441 allocated for training and 860 for validation. After the replication process, the dataset is expanded to 5,663 total images. The new training set comprises 4,122 images, while the validation set grows to 1,541 images. During the copying process, we randomly apply some data augmentation techniques to further increase diversity. This repetition aims to enrich the dataset, ensuring that the model encounters a broader array of scenarios during training. The key objective of this enriched dataset is to expose the model to a wide variety of fire and smoke situations. By increasing variations in size, location, and other factors, the model becomes better equipped to handle a broader range of real-world cases. More diversity in training leads to better adaptability and performance.

3.3. Model Architecture and Modifications

Figure 3 shows the overall architecture of our proposed model. Our suggested method provides a significant improvement to the YOLOv8 design by carefully adding an extra Context to Flow (C2F) layer to the backbone network. The goal of this change is to enhance the model's ability to capture contextual information about fire, refine features, and improve fire detection accuracy [28].

Here is an outline of our suggested method. It is based on the YOLOv8 model, which was chosen due to its superior accuracy and efficiency compared to previous versions of the YOLO model. The C2F module was introduced to provide the model with enhanced capabilities to capture both contextual and flow information crucial for accurate fire detection [28]. We added an extra C2F module after the first conventional layer, which facilitates the model in gathering both environmental and flow data necessary for accurate fire detection, as shown in Figure 3.

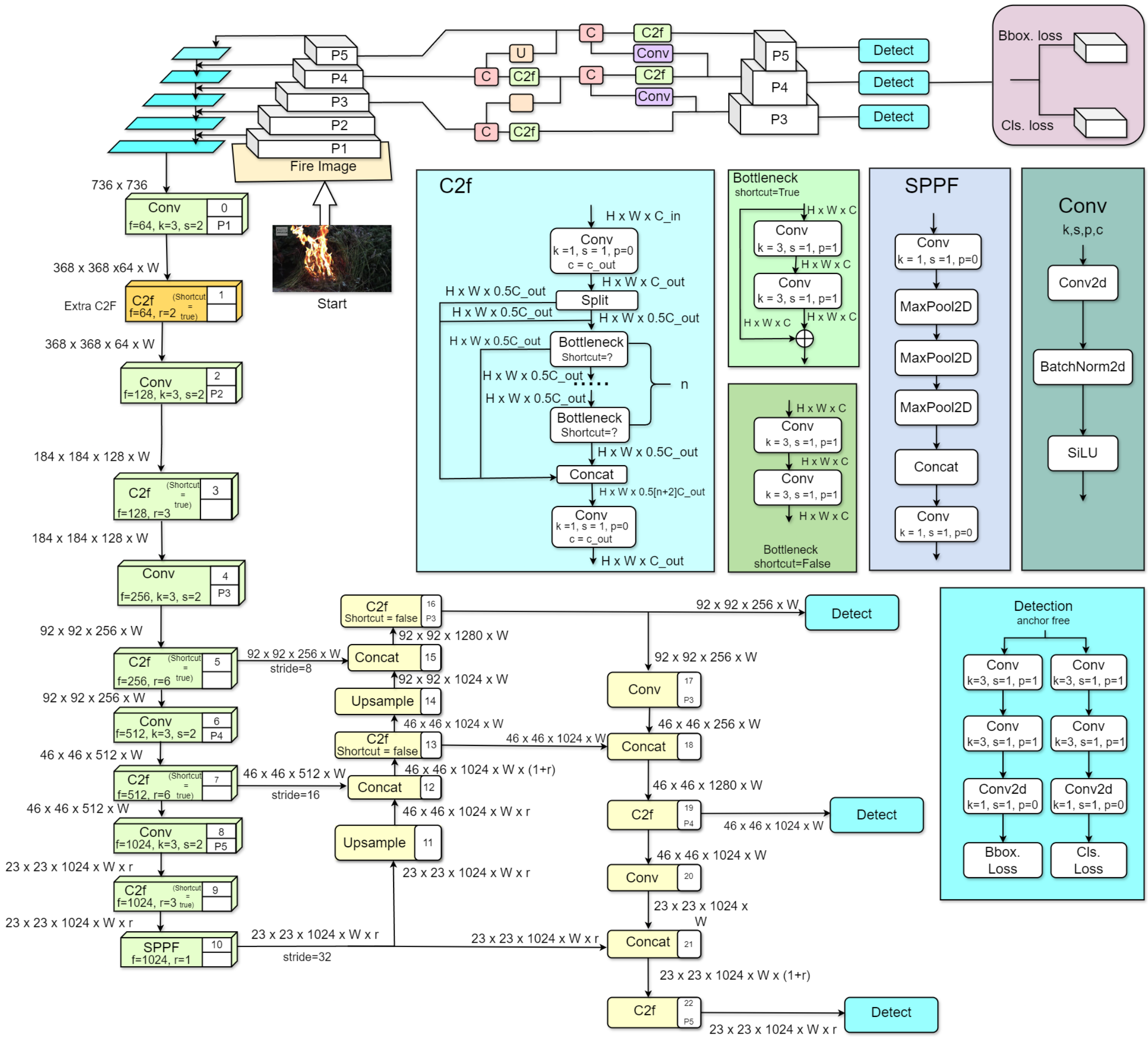


Figure 3. Overall architecture of the proposed model, highlighting the addition of the C2F module.

3.4. Changes to the YOLOv8 Backbone

The C2F module stands for "Context-to-Flow" and is a component of the deep learning architecture. As shown in Figure 4, it consists of two parallel branches: the context branch and the flow branch. The context branch captures the contextual information from the input data and processes it using convolutional layers [29]. In the backbone, after the first convolutional layer, we add a C2F layer to start combining contextual information right away. As shown in Figure 4, this layer, called C2F, is a key link between the first step of feature extraction and the subsequent steps. It identifies important contextual information in the feature representations, which helps the model understand the surrounding context and extract relevant features.

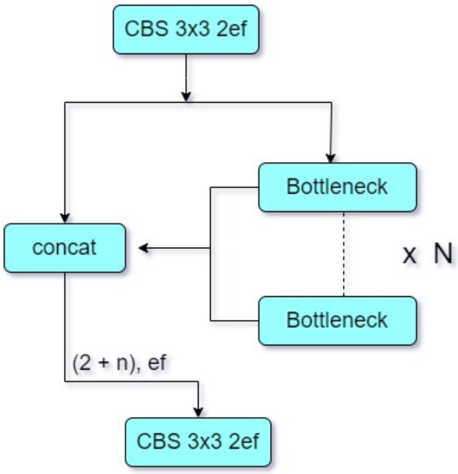


Figure 4. The structure of the C2F module, showing the context and flow branches.

The flow branch focuses on capturing the flow of information within the data. It uses convolutional layers to analyze the spatial and temporal changes in the input [29]. This branch aids the model in understanding the dynamic aspects of the data. The outputs from both branches are then combined to form a robust representation of the input data. This integration of context and flow information enhances the model's ability to capture complex patterns and make accurate predictions. Overall, the C2F module plays a crucial role in deep learning models by facilitating the integration of contextual and flow information, leading to improved performance and accuracy.

3.5. The Training and Success Measure Process

The training process is separated into several steps to achieve better output. Initially, the default YOLOv8 framework is used to train a model with a variety of hyperparameters and activation functions, such as LeakyReLU, ReLU, Sigmoid, Tanh, Softmax, and Mish. After that, training is performed on the modified architecture by loading the default YOLOv8 with the best-performing hyperparameters, as shown in Table 2. In particular, the model with the LeakyReLU activation function and the given hyperparameters performs exceptionally well. Moreover, we retrain the model using the same hyperparameters on our modified architecture by loading the previously trained weights. This method of training is essentially a form of deep transfer learning, where the knowledge obtained from the previous model is utilized by the next model to enhance its performance.

Table 2. Hyperparameters used in the training

Parameters	**Values**
Epochs	350
Batch Size	9
Image Size	736
Optimizer	Adamax
Learning Rate	0.001
Momentum	0.995
Weight Decay	0.00005
Validation	True
Rect	False
Warmup Epochs	4.0
Single Class	False
Patience	0

3.6. Improvements After Using C2f Layer

In the C2F layer, contextual information is added early on, allowing the model to capture global characteristics from the very first layers of the backbone. This additional background information helps the network understand the connections between different parts of the input image. By refining features through the C2F layer, the network gains improved discriminatory power. The C2F layer adapts to focus on important traits while ignoring noise and irrelevant data. By combining features from the first convolutional layer with those from subsequent layers, the C2F layer facilitates multi-scale fusion. This fusion enhances the model's ability to handle objects of varying sizes and scales in the input image, thereby improving detection accuracy.

Adding the C2F layer aids in establishing spatial ordering in the feature maps, which is crucial for accurate object detection and recognition. It enables the network to better understand the surrounding context more quickly during training by providing detailed background information earlier on. The YOLOv8 architecture has been improved by incorporating an additional Context-to-Flow (C2F) layer. This enhancement has resulted in a more comprehensive model, now comprising 387 layers, 81,874,710 parameters, and 81,874,694 gradients, compared to the default YOLOv8 model, which includes 365 layers, 43,631,382 parameters, and 43,631,366 gradients. The computational intensity of the model has significantly increased from 165.4 GFLOPs to 373.1 GFLOPs.

The change in architecture demonstrates how the additional C2F layer has improved the network's feature representation, making it more effective in capturing contextual and dynamic information. As a result, the model's overall accuracy has significantly improved. As previously mentioned, our modified model consists of 387 layers, 81,874,710 parameters, and 81,874,694 gradients. This increase in parameters and gradients enables the model to learn complex patterns and nuances in the data, highlighting the potential of this modified YOLOv8 architecture in advancing object detection performance.

Our proposed model demonstrates a substantial improvement in accuracy for object detection tasks. The C2F layer plays a key role in enhancing the detection capabilities of the YOLOv8 architecture by gathering more contextual and flow information from the input images. This proposed approach appears to be a promising direction for further advancement in deep learning-based object detection research.

4. Result Analysis

This section provides an in-depth analysis of the overall performance of the proposed model, evaluating its effectiveness across various metrics, including accuracy, precision, recall, and computational efficiency.

4.1. Evaluation Matrics

The average value between recall percentages and precision percentages is known as the F1 score. This metric takes into account both false positives and false negatives, making it a comprehensive measure of a model's performance. While accuracy is a widely used metric, it can be misleading when the costs of false positives and false negatives are significantly different. In such cases, it is better to consider both recall and precision alongside accuracy.

Precision is the proportion of correctly predicted positive observations to the total predicted positives. Recall, on the other hand, is the proportion of correctly predicted positive observations to all actual positives. The recall can be calculated using Equation (1):

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{1}$$

Precision is calculated as shown in Equation (2):

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2}$$

In these equations, TP stands for "true positive," FP stands for "false positive," TN stands for "true negative," and FN stands for "false negative." The F1 score, which is a harmonic mean of precision and recall, can be calculated using Equation (3):

$$\mathrm{F1} = 2 \times \frac{\mathrm{Precision} \times \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}} \tag{3}$$

The precision confidence curve, recall confidence curve, F1-score confidence curve, and mAP@50 confidence curve are shown in Figure 5.

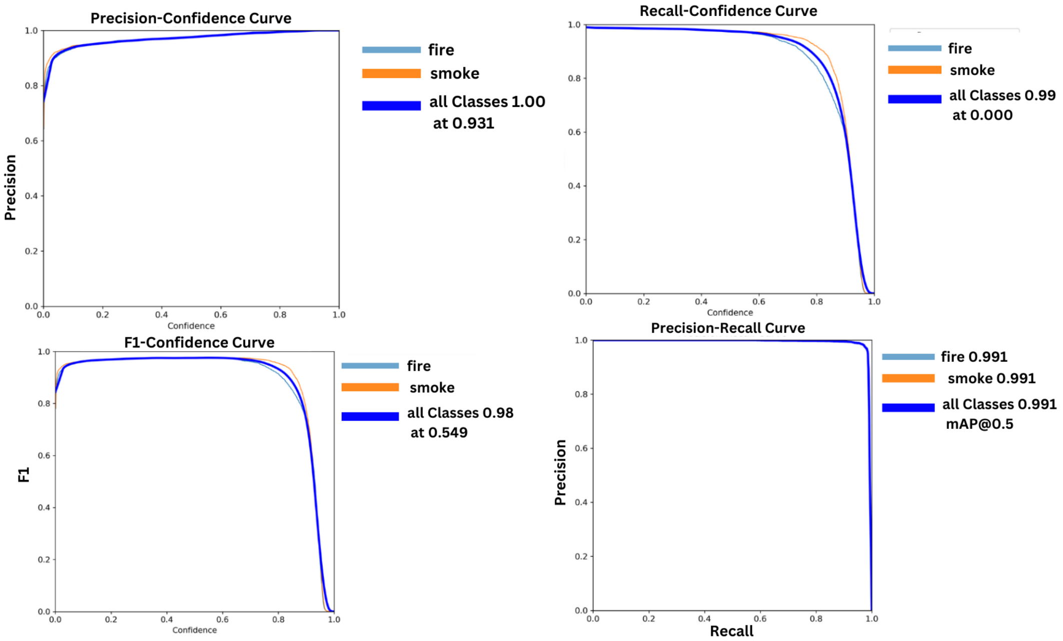
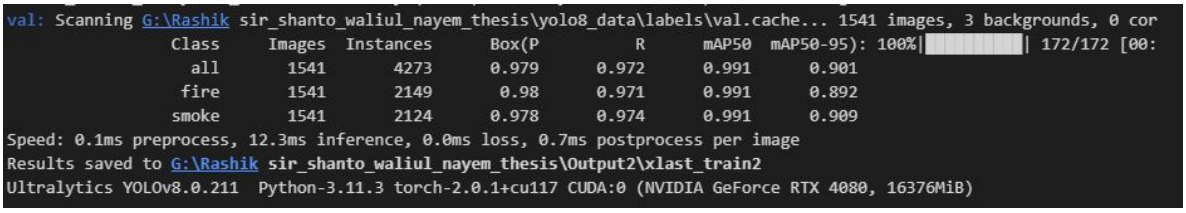


Figure 5. The evacuation matrix precision, recall, f1 score and mAP@50 of the proposed model.

4.2. Results of the Proposed Model

Our findings reveal that the proposed model accurately detects 98% of fires and 97.8% of smoke. The recall rate for fire is 97.1%, and the recall rate for smoke is 97.4%. The mean average precision (mAP) for both classes was 99.1%, as shown in Figure 6. The loss, precision, and recall metrics, which reflect the overall performance of the proposed model, are illustrated in Figure 7.



```
val: Scanning G:\Rashik sir_shanto_waliul_nayem_thesis\yolo8_data\labels\val.cache... 1541 images, 3 backgrounds, 0 cor
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 172/172 [00:
                   all       1541       4273      0.979      0.972      0.991      0.901
                  fire       1541       2149       0.98      0.971      0.991      0.892
                 smoke       1541       2124      0.978      0.974      0.991      0.909
Speed: 0.1ms preprocess, 12.3ms inference, 0.0ms loss, 0.7ms postprocess per image
Results saved to G:\Rashik sir_shanto_waliul_nayem_thesis\Output2\xlast_train2
Ultralytics YOLOv8.0.211  Python-3.11.3 torch-2.0.1+cu117 CUDA:0 (NVIDIA GeForce RTX 4080, 16376MiB)
```

Figure 6. The snippet outlines the training process of the proposed model.

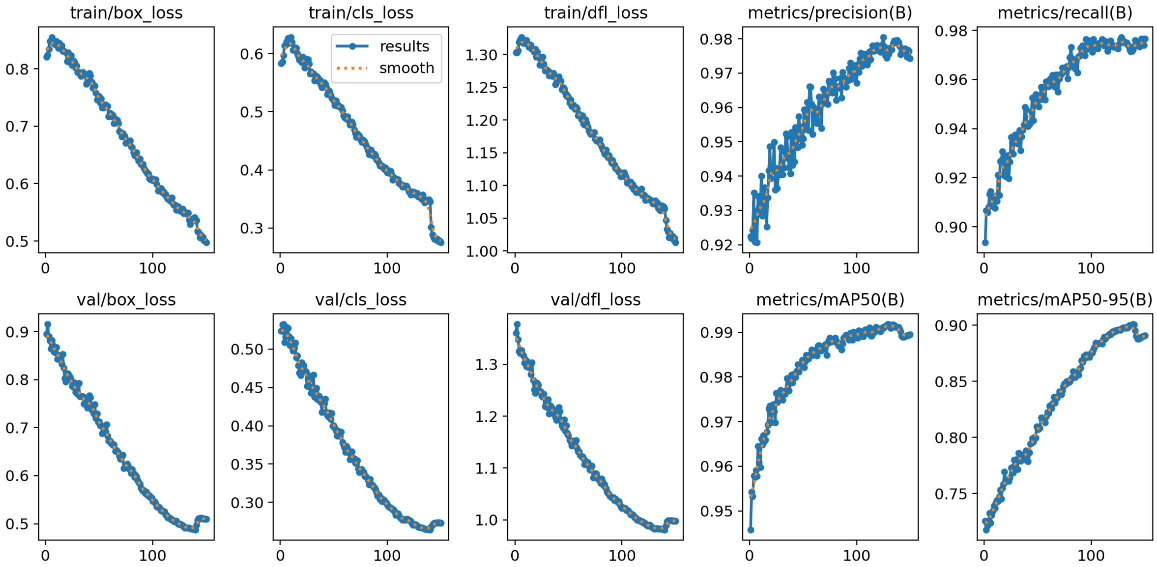


Figure 7. Training and validation performance metrics for proposed fire detection model.

4.3. *Comparing at YOLOv7 and YOLOv8*

We compare our modified YOLOv8 model against the standard YOLOv7 and YOLOv8 models to evaluate its performance improvements. While comparing precision, recall, mAP@50, and F1 scores between the YOLOv7, default YOLOv8, and our modified YOLOv8 model, we observe significant improvements. The bar chart in Figure 8 and Table 3 visually demonstrate these improvements, highlighting that our modified YOLOv8 outperforms across key metrics.

Table 3. Performance comparison of YOLOv7, default YOLOv8, and modified YOLOv8

Model	Precision	Recall	mAP@50	F1-score
YOLOv7	75.1%	98.0%	76.4%	77.0%
YOLOv8 (Default)	95.0%	92.5%	96.5%	94.0%
Proposed Model (YOLOv8)	97.9%	97.2%	99.1%	98.0%

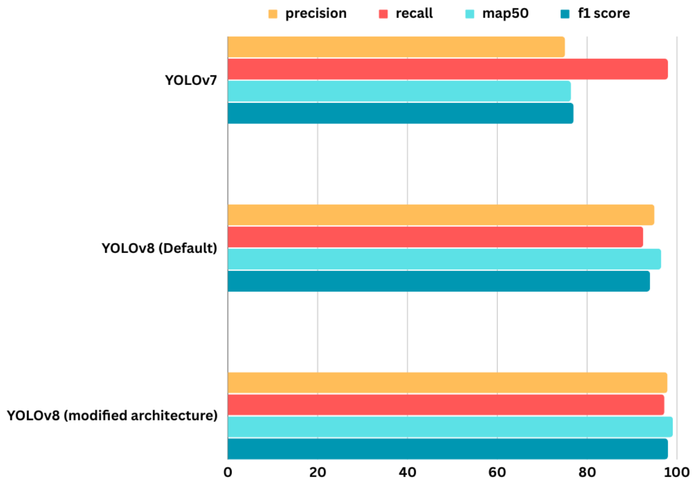


Figure 8. Comparison of YOLOv7, default YOLOv8 and modified YOLOv8.

4.4. Impact of Hyperparameter Tuning on YOLOv8

In this analysis, we delve into the details of hyperparameter tuning. We provide a comprehensive table that outlines the outcomes obtained through different configurations of parameters, such as batch size, image size, learning rate (set at 0.001 for all combinations), momentum, activation function, layer count, optimizer choice, and epochs, which are all integral components of this process. Different F1-scores are obtained for various parameter configurations, as shown in Table 4. The most impactful parameters used in hyperparameter tuning are highlighted in the table.

The results of this tuning process are displayed in Figure 9. The impact of hyperparameter tuning on the overall performance of the model is illustrated in the bar chart, which provides a comparison of precision, recall, mAP@50, and F1 scores.

Table 4. Different parameters used in hyperparameter tuning. In this table, YOLOv8l and YOLOv8n stand for YOLOv8 large and YOLOv8 nano.

No.	Model Type	Activation Function	Epochs	Optimizer	Layers	Momentum	F1-score
1	YOLOv8l	LeakyReLU	350	Adamax	365	0.995	84%
2	YOLOv8l	Softmax	250	Adamax	365	0.994	12%
3	YOLOv8l	Tanh	200	AdamW	365	0.993	89%
4	YOLOv8n	Tanh	200	AdamW	225	0.993	87%
5	YOLOv8l	ReLU	200	AdamW	365	0.999	88%
6	YOLOv8n	ReLU	200	Adamax	225	0.999	84%
7	YOLOv8l	Mish	200	SGD	365	0.995	89%
8	YOLOv8n	Mish	200	AdamW	225	0.995	88%

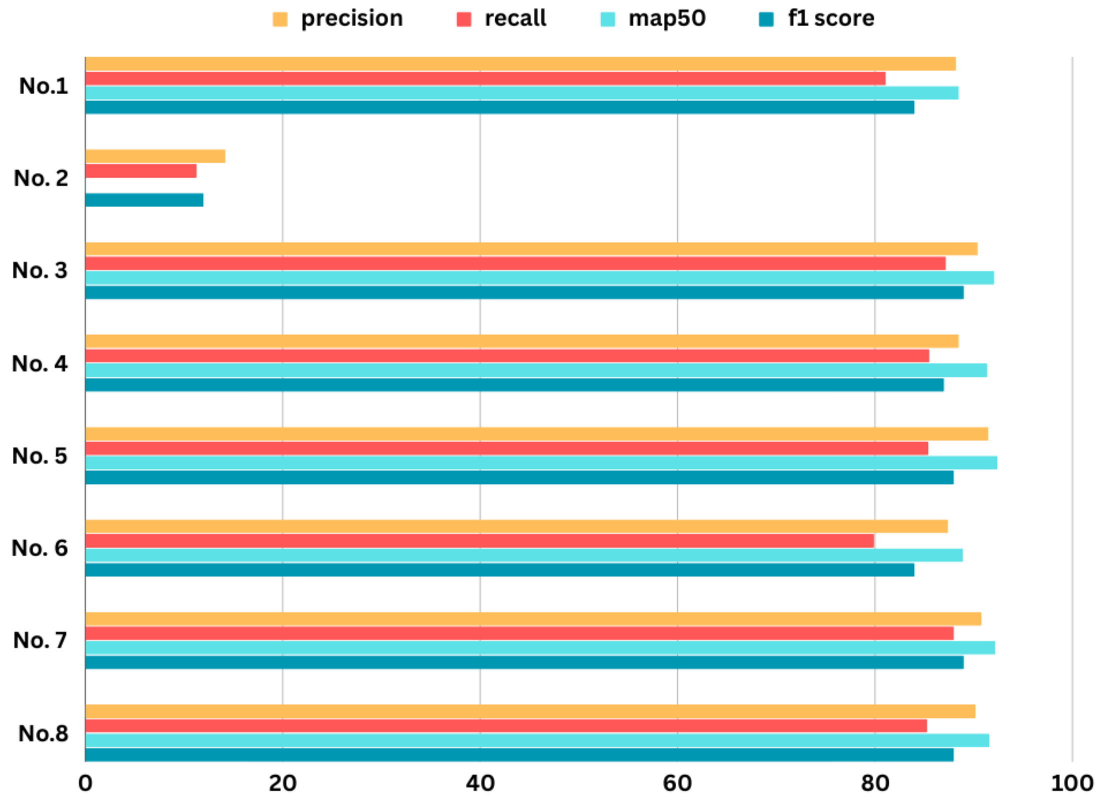


Figure 9. Performance matrix like the precision, recall, map50 and f1 score for the combinations of the table 3.

4.5. Activation Function

Activation functions are mathematical functions that act as gateways for input values. For an artificial neural network to perform complex calculations, it requires more than just a linear representation. Activation functions introduce non-linearity to the network, enabling it to handle complex tasks. Without activation functions, neural networks would be reduced to simple linear regression models [30].

Among all the tested combinations of hyperparameters, the combination used in Model No. 1 in Figure 6 is chosen to train our modified model. This decision was influenced by the behavior of different activation functions. The Tanh function, for instance, performs complex calculations, but it also has limitations. ReLU neurons can be forced into inactive states, where they remain unresponsive to almost all inputs [30]. When in an unresponsive state, no gradients propagate backward through the neuron, rendering it perpetually inactive and entirely unusable [30]. In some cases, a significant portion of neurons within a network may become trapped in these unresponsive states.

Mish, the default activation function of the YOLOv8 model, also performs complex calculations. However, LeakyReLU addresses the problem of unusable states in ReLU neurons and is not as computationally complex as Tanh or Mish. This balance between simplicity and effectiveness is why we selected the LeakyReLU activation function for our proposed model.

4.6. Performance in a Range of Fire Situations

To start our evaluation, we presented some examples of images from inferencing in Figure 10. These images illustrate both small and large fires, as well as smoke. The model demonstrates a strong ability to detect and classify these different fire-related traits, indicating that it can effectively handle a wide range of fire situations.

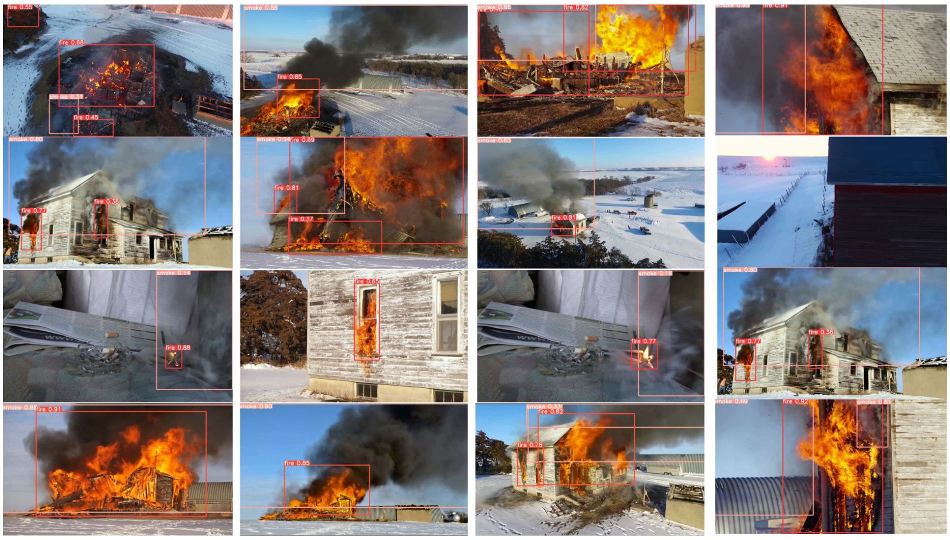


Figure 10. Sample images of inferencing.

5. Explainability with EigenCAM

The YOLOv8 is a CNN-based framework, and the interpretability of convolutional neural networks (CNNs) remains a critical area of research, particularly for applications in safety-critical domains such as fire and smoke detection. To provide visual explanations of our modified YOLOv8 model's predictions, we utilized EigenCAM, a method for generating class activation maps that highlight important regions in the input image contributing to the model's decision. In any CNN classifier, the learning process can be considered a mapping function, where a transformation matrix captures salient features from images using convolutional layers [15,16]. Optimizers adjust the weights of these convolutional filters to learn important features, as well as the weights of fully connected layers to determine the non-linear decision boundary [15,16]. The hierarchical representation mapped onto the last convolutional layer provides a basis for visual explanations of CNN predictions.

5.1. Eigen Class Activation Maps (EigenCAM)

The exceptional performance of CNNs on various computer vision tasks is attributed to their ability to extract and preserve relevant features while discarding irrelevant ones. EigenCAM leverages this capability by identifying features that maintain their relevance through all local linear transformations and remain aligned with the principal components of the learned representation. Given an input image I of size $(i \times j)$, represented as $I \in \mathbb{R}^{i \times j}$ and the combined weight matrix W of the first k layers with size $(m \times n)$, the class activation output O is obtained by projecting the input image onto the last convolutional layer $L = k$:

$$O_{L=k} = W_{L=k}^{T} \times I \tag{4}$$

To compute the principal components of $O_{L=k}$, we factorize $O_{L=k}$ using singular value decomposition (SVD):

$$O = U\Sigma V^{T} \tag{5}$$

Here, U is an $M \times M$ orthogonal encoding matrix with columns as the left singular vectors, Σ is a diagonal matrix of size $M \times N$ with singular values along the diagonal, and V is an $N \times N$ orthogonal matrix with columns as the right singular vectors. The class activation map for EigenCAM, $L_{(\text{EigenCAM})}$, is derived by projecting O onto the first eigenvector V_1:

$$L_{(\text{EigenCAM})} = OV_1 \tag{6}$$

where V_1 is the first eigenvector in the V matrix.

5.2. Implementation and Results of EigenCAM

In our study, we trained a modified YOLOv8 model on a custom dataset of fire and smoke images. To explain the model's predictions, we applied EigenCAM to generate heatmaps for the test images. The heatmap, shown in Figure 11, visually highlights regions of the image that the model focused on when making predictions. For instance, when the model detects fire in an image, the heatmap typically shows red regions corresponding to the areas with flames or smoke, indicating that the model correctly identifies these critical features. By overlaying the heatmaps on the original images, we can visually verify that the model's focus aligns with the expected regions, thereby providing an interpretable explanation of its decision-making process.

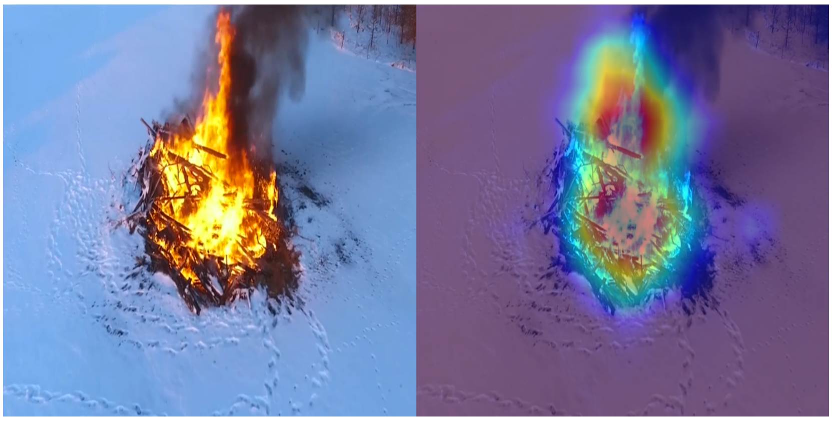


Figure 11. Heatmap generated using EigenCAM, highlighting regions of the image that the model focused on for prediction.

The use of EigenCAM in our research not only enhances the interpretability of our modified YOLOv8 model but also adds an additional layer of validation by confirming that the model's attention is correctly placed on the relevant features in the input images. This explainability is crucial for trust and reliability in real-world applications, particularly in fire and smoke detection systems.

6. Discussion

The results demonstrate that using YOLOv8 improves accuracy, recall, mAP@50, and F1 scores, underscoring the significance of the architectural changes made in the most recent version. The enhanced YOLOv8 model exhibits exceptional object detection performance, particularly in recognizing fires, showcasing its proficiency in this domain. The hyperparameters have been fine-tuned through extensive effort, yielding insightful results. The detailed table and bar chart illustrate how different configurations impact performance metrics, providing both researchers and practitioners with a framework for optimizing model parameters, thereby enabling better customization of the model for specific use cases. Our data analysis indicates that YOLOv8 is more proficient at detecting fires compared to its predecessors. Additionally, the data obtained from hyperparameter adjustments highlights the importance of careful configuration in achieving superior results. Furthermore, EigenCAM validates that the model focuses on the most relevant features for identifying fire and smoke. Our research makes a significant contribution to the ongoing discourse on object detection in both academic and practical contexts.

7. Conclusions

In our research, we successfully optimized the YOLOv8 model to enhance its effectiveness in detecting fire and smoke. We trained the model using a custom dataset and employed EigenCAM, an explainable AI tool, to ensure that our model focuses on the most relevant features when identifying fire and smoke. This validation step is crucial, as it not only confirms the accuracy of our model but also builds trust in its decision-making process. The findings of this study are pivotal for the future of autonomous fire detection technologies. These advancements have the potential to save lives and minimize property damage by enabling quicker and more reliable fire detection. By accurately identifying fire and smoke, these technologies can trigger faster emergency responses, thereby improving safety and reducing risks. Additionally, this research opens up new possibilities in artificial intelligence and computer vision. Our methodology and results provide valuable insights for other researchers tackling similar challenges, offering a foundation for further development. The combination of advanced machine learning models like YOLOv8 with explainability tools like EigenCAM can drive innovation, leading to the creation of smarter and more efficient safety systems.

Author Contributions: Conceptualization, M.W.H., S.S., J.N., R.R., and T.H.; Formal analysis, M.W.H., S.S., and J.N.; Funding acquisition, Z.R. and S.T.M.; Investigation, M.W.H., S.S., and J.N.; Methodology, M.W.H., S.S., J.N., T.H., R.R., and S.T.M.; Supervision, R.R., T.H.; Validation, R.R., T.H., S.T.M, and Z.R.; Visualization, M.W.H., S.S., and J.N..; Writing—original draft, M.W.H., S.S., and J.N.; Writing—review and editing, M.W.H., S.T.M, T.H., R.R. and Z.R. All authors have read and agreed to the published version of the manuscript.

Informed Consent Statement: All information used in this study was sourced from publicly available articles and databases, and therefore, informed consent and ethical approval were not required.

Data Availability Statement: The data presented in this manuscript are available on request.

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

Abbreviations

The following abbreviations are used in this manuscript:

Adam	Adaptive Moment Estimation
CCVT	Closed Circuit Television
C2F	Context To Flow
CNN	Convolutional Neural Networks
CVAT	Computer Vision Annotation Tool
DBSCAN	Density-Based Spatial Clustering Of Applications With Noise
EigenCAM	Eigen Class Activation Maps
GFLOPS	Giga Floating-Point Operations Per Second
IR	Infrared
LeakyReLU	Leaky Rectified Linear Unit
LSTM	Long Short-Term Memory
mAP	Mean Average Precision
MQTT	Message Queuing Telemetry Transport
R-CNN	Region-Based Convolutional Neural Network
ReLU	Rectified Linear Unit
RNN	Recurrent Neural Network
SECSP	Spatially Enhanced Contextual Semantic Parsing
SGD	Stochastic Gradient Descent
SIoU	Scylla Intersection over Union
SLIC	Simple Linear Iterative Clustering
Softmax	Softargmax Or Normalized Exponential Function
SPPF	Spatial Pyramid Pooling Fast
SRoFs	Suspected Regions Of Fire
SSD	Single-Shot Detector
SVD	Singular Value Decomposition
Tanh	Hyperbolic Tangent
VGG	Visual Geometry Group
YOLO	You Only Look Once

References

1. Saponara, S.; Elhanashi, A.; Gagliardi, A. Real-time video fire/smoke detection based on CNN in antifire surveillance systems. *Journal of Real-Time Image Processing* **2021**, *18*, 889–900. doi:10.1007/s11554-020-01044-0.
2. Abdusalomov, A.; Baratov, N.; Kutlimuratov, A.; Whangbo, T. An improvement of the fire detection and classification method using YOLOv3 for surveillance systems. *Sensors* **2021**, *21*, 6519. doi:10.3390/s21196519.
3. Barmpoutis, P.; Stathaki, T.; Dimitropoulos, K.; Grammalidis, N. Early fire detection based on aerial 360-degree sensors, deep convolution neural networks and exploitation of fire dynamic textures. *Remote Sensing* **2020**, *12*, 3177. doi:10.3390/rs12193177.
4. Avazov, K.; Mukhiddinov, M.; Makhmudov, F.; Cho, Y. Fire detection method in smart city environments using a deep-learning-based approach. *Electronics* **2022**, *11*, 73. doi:10.3390/electronics11010073.
5. Ibraheem, M.K.I.; Mohamed, M.B.; Fakhfakh, A. Forest Defender Fusion System for Early Detection of Forest Fires. *Computers* **2024**, *13*, 36. doi:10.3390/computers13020036.
6. Liu, J.; Yin, J.; Yang, Z. Fire Detection and Flame-Centre Localisation Algorithm Based on Combination of Attention-Enhanced Ghost Mode and Mixed Convolution. *Applied Sciences* **2024**, *14*, 989. doi:10.3390/app14030989.
7. Wang, B.; Zhao, X.; Zhang, Y.; Song, Z.; Wang, Z. A capacitive particle-analyzing smoke detector for very early fire detection. *Sensors* **2024**, *24*, 1692. doi:10.3390/s24051692.
8. Pincott, J.; Tien, P.; Wei, S.; Calautit, J. Indoor fire detection utilizing computer vision-based strategies. *Journal of Building Engineering* **2023**, *61*, 105154. doi:10.1016/j.jobe.2022.105154.
9. Muhammad, K.; others. Convolutional neural networks based fire detection in surveillance videos. *IEEE Access* **2018**, *6*, 18174–18183. doi:10.1109/ACCESS.2018.2812835.

10. de Venancio, P.; Lisboa, A.; Barbosa, A. An automatic fire detection system based on deep convolutional neural networks for low-power, resource-constrained devices. *Neural Computing and Applications* **2022**, *34*, 15349–15368. doi:10.1007/s00521-022-07467-z.
11. Valikhujaev, Y.; Abdusalomov, A.; Cho, Y. Automatic fire and smoke detection method for surveillance systems based on dilated CNNs. *Atmosphere* **2020**, *11*, 1241. doi:10.3390/atmos11111241.
12. Pundir, A.; Raman, B. Dual deep learning model for image based smoke detection. *Fire Technology* **2019**, *55*, 2419–2442. doi:10.1007/s10694-019-00872-2.
13. Park, J.H.; Lee, S.; Yun, S.; Kim, H.; Kim, W.T. Dependable fire detection system with multifunctional artificial intelligence framework. *Sensors* **2019**, *19*, 2025. doi:10.3390/s19092025.
14. Park, M.; Ko, B.C. Two-step real-time night-time fire detection in an urban environment using Static ELASTIC-YOLOv3 and Temporal Fire-Tube. *Sensors* **2020**, *20*, 2202. doi:10.3390/s20082202.
15. Muhammad, M.B.; Yeasin, M. Eigen-CAM: Visual Explanations for Deep Convolutional Neural Networks. *SN Computer Science* **2021**, *2*, 47. doi:10.1007/s42979-021-00449-3.
16. Bany Muhammad, M.; Yeasin, M. Eigen-CAM: Visual Explanations for Deep Convolutional Neural Networks. *SN Computer Science* **2021**, *2*, 47. doi:10.1007/s42979-021-00449-3.
17. Ahn, Y.; Choi, H.; Kim, B. Development of early fire detection model for buildings using computer vision-based CCTV. *Journal of Building Engineering* **2023**, *65*, 105647. doi:10.1016/j.jobe.2022.105647.
18. Muhammad, K.; Ahmad, J.; Baik, S. Early fire detection using convolutional neural networks during surveillance for effective disaster management. *Neurocomputing* **2019**, *288*, 30–42. doi:10.1016/j.neucom.2017.04.083.
19. Kim, B.; Lee, J. A video-based fire detection using deep learning models. *Applied Sciences* **2019**, *9*, 2862. doi:10.3390/app9142862.
20. Xie, Y.; others. Efficient video fire detection exploiting motion-flicker-based dynamic features and deep static features. *IEEE Access* **2020**, *8*, 81904–81917. doi:10.1109/ACCESS.2020.2991338.
21. Li, P.; Zhao, W. Image fire detection algorithms based on convolutional neural networks. *Case Studies in Thermal Engineering* **2019**, *19*, 100625. doi:10.1016/j.csite.2020.100625.
22. Sheng, D.; Deng, J.; Xiang, J. Automatic smoke detection based on SLIC-DBSCAN enhanced convolutional neural networks. *Sensors* **2020**, *20*, 5608. doi:10.1109/ACCESS.2021.3075731.
23. Lin, G.; Zhang, Y.; Xu, G.; Zhang, Q. Smoke detection on video sequences using 3D convolutional neural networks. *Fire Technology* **2019**, *55*, 1827–1847. doi:10.1007/s10694-019-00832-w.
24. Zhao, Y.; Zhang, H.; Zhang, X.; Chen, X. Fire smoke detection based on target-awareness and depthwise convolutions. *Multimedia Tools and Applications* **2020**, *80*, 27407–27421. doi:10.1007/s11042-021-11037-1.
25. Dilshad, N.; Khan, T.; Song, J. Efficient deep learning framework for fire detection in complex surveillance environment. *Computer Systems Science and Engineering* **2023**, *46*, 749–764. doi:10.32604/csse.2023.034475.
26. Yunusov, N.; Islam, B.M.S.; Abdusalomov, A.; Kim, W. Robust Forest Fire Detection Method for Surveillance Systems Based on You Only Look Once Version 8 and Transfer Learning Approaches. *Processes* **2024**, *12*. doi:10.3390/pr12051039.
27. Jocher, G.; Chaurasia, A.; Qiu, J.; Stoken, R.; others. YOLOv8: You Only Look Once - Version 8. https://github.com/ultralytics/ultralytics, 2023. Accessed: 2024-09-01.
28. Fahy, R.; Petrillo, J. Firefighter Fatalities in the US in 2022. https://www.usfa.fema.gov/downloads/pdf/publications/firefighter-fatalities-2022.pdf, 2022. Accessed: 2024-06-01.
29. Solawetz, J.; Francesco. What is YOLOv8? The Ultimate Guide. https://blog.roboflow.com/whats-new-in-yolov8/, 2023. Accessed: 2024-07-19.
30. Inside, A.I. Introduction to deep learning with computer vision — learning rates & mathematics — part 1. https://medium.com/hitchhikers-guide-to-deep-learning/13-introduction-to-deep-learning-with-computer-vision-learning-rates-mathematics-part-1-4973aacea801, 2020. Accessed: 2024-09-03.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.