fire

MDPI

Article

A Comparative Analysis of YOLOv9, YOLOv10, YOLOv11 for Smoke and Fire Detection

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: Forest fires cause extensive environmental damage, making early detection crucial for protecting both nature and communities. Advanced computer vision techniques can be used to detect smoke and fire. However, accurate detection of smoke and fire in forests is challenging due to different factors such as different smoke shapes, changing light, and similarity of smoke with other smoke-like elements such as clouds. This study explores recent YOLO (You Only Look Once) deep-learning object detection models YOLOv9, YOLOv10, and YOLOv11 for detecting smoke and fire in forest environments. The evaluation focuses on key performance metrics, including precision, recall, F1-score, and mean average precision (mAP), and utilizes two benchmark datasets featuring diverse instances of fire and smoke across different environments. The findings highlight the effectiveness of the small version models of YOLO (YOLOv9t, YOLOv10n, and YOLOv11n) in fire and smoke detection tasks. Among these, YOLOv11n demonstrated the highest performance, achieving a precision of 0.845, a recall of 0.801, a mAP@50 of 0.859, and a mAP@50-95 of 0.558. YOLOv11 versions (YOLOv11n and YOLOv11x) were evaluated and compared against several studies that employed the same datasets. The results show that YOLOv11x delivers promising performance compared to other YOLO variants and models.

Keywords: fire detection; forest fire; YOLOv9; YOLOv10; YOLOv11

check for updates

Academic Editor: Jinlong Zhao

Received: 5 November 2024
Revised: 18 December 2024
Accepted: 6 January 2025
Published: 13 January 2025

Citation: Alkhammash, E.H. A Comparative Analysis of YOLOv9, YOLOv10, YOLOv11 for Smoke and Fire Detection. *Fire* **2025**, *8*, 26. https://doi.org/10.3390/fire8010026

Copyright: © 2025 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Forest fires pose serious threats to forest safety and stability, increasing risks for ecosystems and nearby communities. They lead to massive losses of wildlife and plants, destroy millions of hectares of forest, disrupt ecological balance, and create substantial financial burdens for governments [1,2]. In 2020, a major wildfire in Australia damaged hundreds of structures, burned over 19 million hectares, killed 33 people, injured 450, and harmed over a million animals [3,4]. In 2020, the United States had disastrous wildfires, with over 50,000 fires consuming over 3.64 million acres of land, the largest number recorded in the 24-year period from 2000 to 2024 [5]. In June 2023, devastating wildfires in northeastern Kazakhstan resulted in the largest recorded annual death toll, with at least 15 firemen killed [6,7].

Figure 1 depicts the annual wildfire-burned area across continents from 2012 to 2024. While most regions show stable trends, Oceania shows substantial variations, reaching approximately 40,000 hectares in both 2018 and 2022.



Figure 1. Average area burnt per wildfire from 2012 to 2024 in hectares (Last update on 30 October 2024) [4].

Forest fires are frequently started by burning debris and smoke serves as an early warning indication. Detecting smoke accurately and quickly may significantly reduce response times and limit fire damage. However, current smoke detection systems require improvement to better monitor forest fires. Horizontal detection boxes, for example, may overlook key aspects, resulting in inaccuracy and false alarms by associating non-smoke components with smoke [8]. In dynamic forest environments, smoke-like elements such as clouds and fog frequently appear, closely resembling smoke and complicating distinction for feature extraction networks. Additionally, when fires are distant, smoke becomes less visible, reducing the detection box's confidence and often filtering out actual smoke, making accurate detection more challenging [8].

Advancements in computer vision have made video surveillance systems essential in forests, playing a key role in early fire detection. These systems offer clear advantages over traditional machine learning approaches, such as contactless operation, broad coverage, quick response times, and high detection accuracy [9]. By focusing on visual aspects such as color, texture, motion, and background contrast, video surveillance can reliably identify signs of smoke and fire. Traditional machine learning algorithms for detecting fire and smoke face significant challenges, including low accuracy, limited generalization, small-scale pixel features, and the complex backgrounds in forest fire images [9]. Deep learning algorithms, particularly YOLO, have made a major impact on object detection. Deep learning-based detection methods fall into two categories: region-based and regression-based. Region-based detection is a two-stage process that identifies candidate boxes for objects and then classifies them using a CNN. In contrast, regression-based algorithms use a one-stage detection approach, applying regression analysis to deliver fast, high-performance real-time detection [10].

There are many series of YOLO starting from YOLO1 [11], and followed by YOLOv2 [12], YOLOv3 [13], YOLOv4 [14], YOLOv5 [15], YOLOv6 [16], YOLOv7 [17], YOLOv8 [18], YOLOv9 [19], YOLOv10 [20], and YOLOv11 [21].

This study explores and evaluates the latest lightweight versions of YOLO models, specifically YOLOv9, YOLOv10, and YOLOv11, for their application in smoke and fire detection. While previous research has focused on earlier versions of YOLO, such as YOLOv5 and YOLOv8, there is limited work on the potential of YOLOv11, the latest iteration in the Ultralytics YOLO series of real-time object detectors, in the context of forest fire detection. This paper aims to fill that gap by evaluating the performance of

YOLOv11n and YOLOv11x and comparing them to other YOLO architectures across two benchmark datasets.

The main contributions of this paper are as follows:

1. Exploring the latest YOLO versions: This study investigates the lightweight versions of YOLOv9, YOLOv10, and YOLOv11 for smoke and fire detection, emphasizing the potential of these advanced deep learning models in forest fire detection. Among the models YOLOv9t, YOLOv10n, and YOLOv11n, the YOLOv11n model achieved the highest performance metrics, demonstrating its effectiveness in accurate smoke and fire detection.
2. Evaluation of YOLOv11n and YOLOv11x: The performance of YOLOv11n and YOLOv11x was evaluated against several studies utilizing different YOLO architectures, including YOLOv5, YOLOv8, and modified YOLO versions, using two benchmark datasets: the Smoke and Fire dataset and the D-Fire dataset. YOLOv11x demonstrated competitive and highly effective performance in these comparisons.

The study is organized as follows: Section 2 reviews related work, providing background on previous efforts in smoke and fire detection using YOLO. Section 3 outlines the methodology and materials. This section also describes the YOLOv9, YOLOv10, and YOLOv11 models for smoke and fire detection. Section 4 presents the results, where a comparative analysis reveals that YOLOv11n is the best-performing model among the lightweight models of YOLO. YOLOv11x, however, achieved competitive results compared with other studies using two different datasets. Section 5 concludes the study, summarizing the findings and emphasizing the potential of these models for real-time forest fire detection.

2. Related Works

Deep learning models are extensively used for fire and smoke detection, with the YOLO (You Only Look Once) family gaining significant attention for its effectiveness in this field. YOLO models are widely applied for real-time smoke and fire detection due to their high speed and accuracy. This section reviews studies focused on the use of YOLO models for fire and smoke detection. Table 1 summarizes several state-of-the-art studies in this area, highlighting advancements in fire and smoke detection.

Table 1. Several cutting-edge studies in fire and smoke detection of YOLO models.

References	Model Type	Description	Dataset	Results
[8]	YOLOv3, YOLOv5, YOLOv7	YOLOv5, YOLOv7, and YOLOv3 models are used. Data augmentation techniques were used such as image cropping and color adjustments to increase the number of training samples.	A publicly available Kaggle dataset was used, containing 737 images of different zones (close, medium, distant).	The mAP accuracy of YOLOv5 is 96.8%, YOLOv7 is 95%, and YOLOv3 is 94.8%.
[22]	YOLOv8	Smart Fire Detection System (SFDS) is created based on YOLOv8 algorithm and consists of four layers: Application, Fog, Cloud, and IoT.	The dataset includes 26,520 images with a wide variety of fire and smoke scenarios, including both indoor and outdoor fires, small to large-scale fires with varying lighting levels, as well as normal scenes without fire or smoke.	Precision rate of 97.1% for all classes.

Table 1. *Cont.*

References	Model Type	Description	Dataset	Results
[23]	YOLOv5	Improved YOLOv5 model with global attention mechanism, re-parameterized convolutional module, decoupled detection head, and weighted bi-directional feature pyramid network (BiFPN).	The dataset contains 2876 annotated images of forest and grassland fires from news, public datasets, VisiFire, and Forestry Images. It includes fires of low, medium, and high intensities.	Precision increased by 4.2%, recall by 3.8%, and mAP by 4.6% compared to the classic YOLOv5.
[24]	Yolov5s and Yolov5l	Ensemble approach combining YOLOv5 and CNN based on two-stage pipeline for wildfire and smoke detection: CNN detects abnormalities in the frame and YOLOv5 localizes smoke or fire.	Data were collected from different sources and datasets. It contains 937 annotated images.	The smoke detection model achieves an mAP@0.5 of 0.85 and an F1-score of 0.93, whereas the combined model achieves an mAP@0.5 of 0.76.
[25]	YOLOv5, YOLOv6, YOLOv7, YOLOv8, YOLO-NAS	Evaluation of YOLO models (YOLOv5, YOLOv6, YOLOv7, YOLOv8, YOLO-NAS).	The Foggia dataset consists of 8974 images for wildfire and smoke detection (3731 fire and 6791 smoke).	YOLOv5, YOLOv7, YOLOv8 balanced well across all metrics; YOLOv6 had a slightly lower recall on validation but balanced in testing; YOLO-NASl has the highest recall of 0.971 but lower precision 0.107.
[26]	EFA-YOLO	EFA-YOLO model with EAConv (Efficient Attention Convolution) and EADown (Efficient Attention Downsampling) modules to enhance feature extraction and downsampling.	The fire dataset includes 2060 images across diverse fire scenarios indoor/outdoor, day/night, and various environments like buildings, grasslands, and vehicles, with different fire sizes.	EFA-YOLO outperforms YOLOv5, YOLOv8, YOLOv9, and YOLOv10 in terms of accuracy (mAP) and speed, with 94.6% fewer parameters and an inference time improved by 88 times.
[27]	YOLOv5, YOLOv10, YOLOv7	Comparing performances of the YOLO-type family of object detectors for fire and smoke detection.	Fire and Smoke dataset is introduced with more than 22,000 images and 93,000 instances from 1200 YouTube videos and public sources. The dataset contains diverse fire/smoke scenarios across various environments (day/night, indoor/outdoor, urban/forest, etc.).	YOLOv10 achieved 89% accuracy and mAP@50 large than 91%.
[28]	YOLOv5	Lightweight YOLOv5-based model for forest smoke and fire detection. It integrates C3Ghost, Ghost modules, and Coordinate Attention (CA) to reduce parameters, enhance feature focus, and suppress irrelevant background information. PAN-weight added in Neck for improved feature fusion.	11,667 images were collected from sources including web-captured images/videos and a self-collected tree-burning dataset. Videos were processed by frame extraction (every 30 frames) and labeled with LabelImg. The dataset covers forest fires during the day, dusk, and night.	The model size was reduced by 44.75%, and FLOPs were reduced by 47.46%. Precision increased by 2.53%, and mAP@0.5 increased by 1.16%.
[9]	YOLOv8 and YOLOv7	YOLOv8 and YOLOv7 models are used for smoke and fire detection.	The same dataset was used in [28].	YOLOv8 achieved 92.6% mAP@0.5, 83.7% precision, and 95.2% recall.

This study presents a comparative analysis of YOLOv9, YOLOv10, and YOLOv11 for smoke and fire detection. YOLOv11 outperformed the other versions. The YOLOv11n and YOLOv11x models were evaluated on two datasets: the Smoke and Fire Dataset and the D-Fire Dataset. When compared to other models in different studies, YOLOv11x demonstrated competitive performance, underscoring its promising capabilities. A detailed comparison with other studies is provided in Section 4.

3. Materials and Methods

3.1. Smoke and Fire Dataset

The dataset used in this study contains 11,667 RGB images [28]. The dataset was gathered from two sources: web-based videos and images, and a self-collected tree burning dataset. Frame extraction was performed (30 frames apart) on the videos, and the extracted frames were added to the image dataset, which was then labeled using the LabelImg tool. The dataset was categorized as smoke and fire. The smoke category was the smaller category [28]. Figure 2 shows a sample of four images of forest fires at night. The number of training images: 8494, validation images: 2114, and testing images: 1059.



Figure 2. Sample of images for forest fires at night [28].

3.2. D-Fire Dataset

The D-Fire dataset, developed by Venâncio et al. [29,30] is used for detecting smoke and fire. It contains 21,527 images, including aerial and ground-level images. It consists of a total of 1164 images showing only fire, 5867 showing smoke, 4658 featuring both smoke and fire, and 9838 with neither smoke nor fire. The dataset captures smoke and fire in various forms, with different shapes, textures, intensities, sizes, and colors. It also includes challenging scenarios, such as insects blocking the camera, raindrops, lighting variations, fog, clouds, and sun glare. The D-Fire dataset used in this study is divided into training, testing, and validation sets. Initially, 20% of the dataset (4306 images) was allocated for testing. The remaining images were further split into a training set of 15,498 images and a validation set of 1723 images. A Sample of images for D-Fire dataset is shown in Figure 3.



Figure 3. Sample of images for D-Fire dataset [29,30].

3.3. YOLO (Version 9, 10, and 11) Architecture

This study utilizes three models for smoke and fire detection: YOLOv9, YOLOv10, and YOLOv11. Figure 4 illustrates the architectures of these models, highlighting their unique design features for effective real-time detection of smoke and fire.

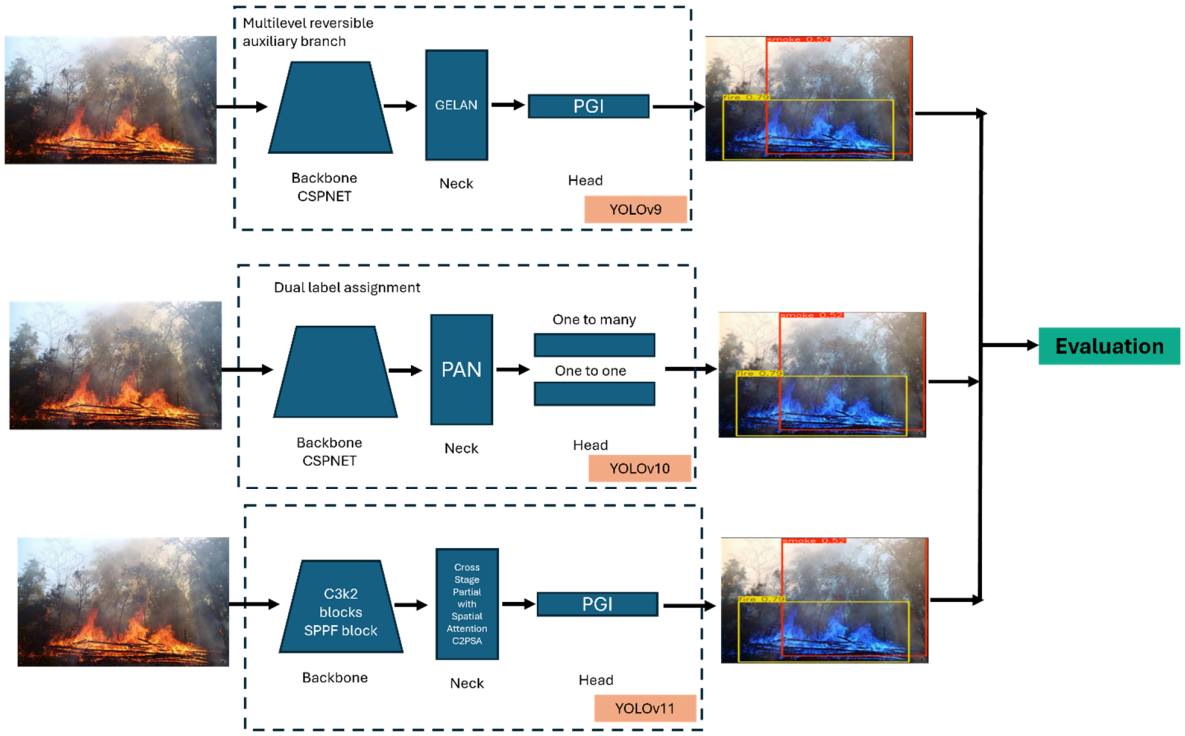


Figure 4. Overview of the YOLO models (YOLOv9, YOLOv10, and YOLOv11) for smoke and fire detection.

3.3.1. YOLOv9

YOLOv9 was introduced on 21 February 2024, and it included programmable gradient information (PGI) to solve the vanishing gradient problem in deep neural networks. By March 2024, YOLOv9 had integrated a generalized efficient layer aggregation network (GELAN), which improved multi-scale feature aggregation and gradient flow and minimized computational costs. In April 2024, more optimizations to YOLOv9 make it appropriate for lightweight deployment on edge devices, balancing outstanding performance with resource efficiency for IoT and mobile applications [29,30]. YOLOv9 uses a CNN-based backbone like YOLOv8, with GELAN incorporating CSPblocks, Resblocks, and Darkblocks for better feature extraction that effectively preserves key features across the network [30]. The neck of YOLOv9 uses PGI, which improves the feature fusion process. PGI reduces information loss as data progresses through the network and stabilizes gradient computations, enabling YOLOv9 to detect objects effectively for different sizes [30]. YOLOv9 uses the anchor-free bounding box prediction from YOLOv8 but is improved by PGI's reversible functions, which eliminate data loss in forward and backward passes, and ultimately improves inference speed and accuracy for real-time applications [30].

Five different variants for YOLOv9 based on variations in architecture size and complexity: YOLOv9t: tiny, YOLOv9s: small, YOLOv9m: medium, YOLOv9c: compact, and YOLOv9e: extended. YOLOv9 supports different tasks including instance segmentation and object detection.

Pseudo code of YOLOv9 [30]
Input: images, imgsz = 640
Backbone: This is the main body of the network that contains CNN-based backbone + GELAN (CSPblocks, Resblocks, and Darkblocks) for multi-scale feature extraction.
Neck: This part connects the backbone and the head. It has PANet + PGI for feature fusion to enhance spatial and semantic information.
Head: This part makes object detection predictions over the refined features that come from the neck and generates the final output (an anchor-free bounding box prediction, class prediction, objectness score). The reversible architecture ensures that no crucial data is lost during the forward and backward passes.

3.3.2. YOLOv10

YOLOv10 was developed at Tsinghua University and released in May 2024 [31]. YOLOv10 employs an updated version of CSPNet that improves gradient flow and reduces computational redundancy, enabling precise feature extraction. It also incorporates Path Aggregation Network (PAN) within the neck for multi-scale feature fusion. The dual head design is another improvement in YOLOv10 that offers rich and multiple predictions of each object during training, while the One-to-One approach during inference ensures faster and more efficient object detection [32]. In addition, YOLOv10 employs dual assignment strategies for NMS-free training, enabling faster response time without sacrificing accuracy [33]. YOLOv10 enables multi-scale feature processing through Parallel Split-Attention (PSA) module and the Compact Inverted Bottleneck (CIB). Scaled Residual Connection and Scaled Weight Shortcut techniques are used to improve accuracy [34]. The YOLOv10 series includes models for various needs: YOLOv10n: nano, YOLOv10s: small, YOLOv10m: medium, YOLOv10b: balanced, YOLOv10l: large, and YOLOv10x: extra-large [32].

Pseudo code of YOLOv10 [34]
Input: images, imgsz = 640
Backbone: This part uses an enhanced version of CSPNet (Cross Stage Partial Network) for feature extraction
Neck: This part includes PAN (Path Aggregation Network) layers for multiscale feature fusion. It combines features from various scales and passes them to the head.
One-to-Many Head: This part generates multiple predictions per object in training.
One-to-One Head: This part generates a single best prediction per object during inference.

3.3.3. YOLOv11

One of the major advancements in YOLOv11 is the adoption of the C3k2 block, which replaces the C2f block found in earlier versions. The multiple C3k2 block is an efficient type of Cross Stage Partial (CSP) module that has two smaller convolutions instead of a single larger one, as seen in YOLOv8. The "k2" in C3k2 signifies the smaller kernel size, which enhances processing speed without sacrificing model performance [35]. The Cross Stage Partial with Spatial Attention (C2PSA) block, introduced in YOLOv11, improves spatial attention within feature maps, helping the model to focus more precisely on critical areas of an image [35]. SPFF (Spatial Pyramid Feature Fusion) is a block that enables the model to emphasize regions of interest, which can boost detection accuracy for objects of different sizes and positions across the image [35]. The YOLOv11 detection series includes YOLOv11n: nano, YOLOv11s: small, YOLOv11m: medium, YOLOv11l: large, and YOLOv11x: Extra-large, with nano optimized for lightweight tasks, small and medium for balanced speed and accuracy, and large and xlarge for maximum performance. YOLOv11 expands beyond object detection to support a range of computer vision tasks, including

instance segmentation, image classification, pose estimation, and oriented object detection (OBB), making it as versatile as YOLOv8 [36].

Pseudo Code for YOLOv11 [21,36]
Input: images, imgsz = 640
Backbone: This part includes different layers or blocks: convolutional layers and C3K3 block, SPPF aggregate multi-scale spatial information, C2PSA attention mechanism to enhance spatial attention in the feature maps
Neck: This part includes C3k2 Block and C2PSA module. It involves upsampling layers for increasing spatial resolution and concatenation of feature maps from different layers to combine information.
Head: This part generates the final predictions of object detection and classification. The detect module output bounding box coordinates, abjectness scores, and class probabilities.

3.4. Performance Metrics

To evaluate fire and smoke detection models, four essential metrics are used: Precision, Recall, F1-score, and Mean Average Precision (mAP). Precision measures the accuracy of positive predictions. Recall indicates the model's ability to capture all relevant instances. F1-Score is the harmonic mean of Precision and Recall, providing a balanced metric for both accuracy and coverage. The Mean Average Precision (mAP) calculates the average precision across all classes, reflecting overall model performance. The definitions of these metrics are given as follows:

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

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

$$\text{F1 Score} = 2 \times \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}$$

$$\text{mAP} = \frac{1}{\text{N}} \sum_{\text{c}=1}^{\text{N}} \text{AvgPrecision}_{\text{c}}$$

where FP stands for False Positive, TN for True Negative, TP for True Positive, and FN for False Negative. N is the total number of classes, and $\text{AvgPrecision}_{\text{c}}$ is the Average Precision for class c. It measures how well the model detects instances of that particular class, taking both precision and recall into account across different thresholds.

4. Results

The experimental setup parameters of the study and the training parameters are shown in Tables 2 and 3.

Table 2. Setup parameters of the study.

Name	Type
CPU	CPU 4 cores
GPU	Nvidia Tesla P100 32 GB
RAM	32 GB
Framework	PyTorch
Accelerator	GPU P100

Table 3. Training parameters of the deep learning models.

	YOLOv11	**YOLOv10**	**YOLOv9**
Epochs	50, 100	50, 100	50, 100
Batch size	16	16	16
Optimal Algorithm	Adam	Adam	Adam
Model weight	YOLOv11n	YOLOv10n	YOLOv9t
Number of parameters (million)	2.6	2.3	2

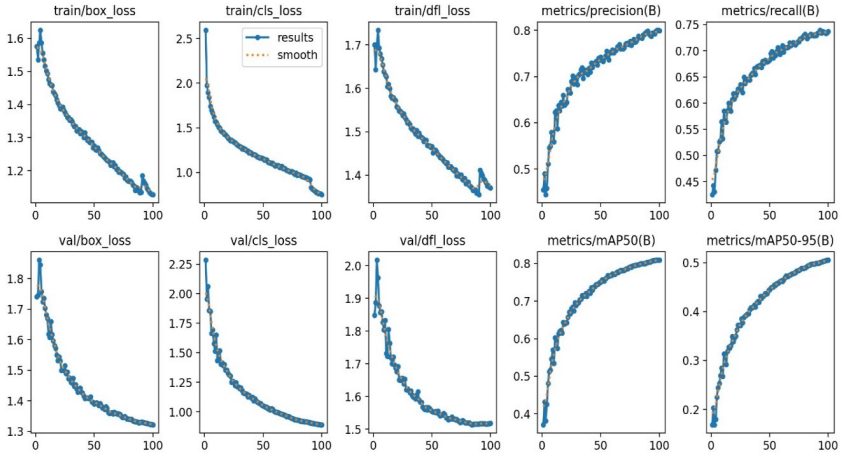
Table 4 compares the performance of YOLOv11n, YOLOv10n, and YOLOv9t models over 50 and 100 epochs, showing improvements in precision, recall, mAP@50, mAP@50-95, and fitness with extended training. YOLOv11n achieves the highest scores after 100 epochs, indicating that longer training boosts detection accuracy and that YOLOv11 is the most effective model among the three.

Table 4. Performance metrics of YOLOv11n, YOLOv10n, and YOLOv9t models at 50 and 100 epochs, showing precision, recall, mAP@50, mAP@50-95, and fitness score.

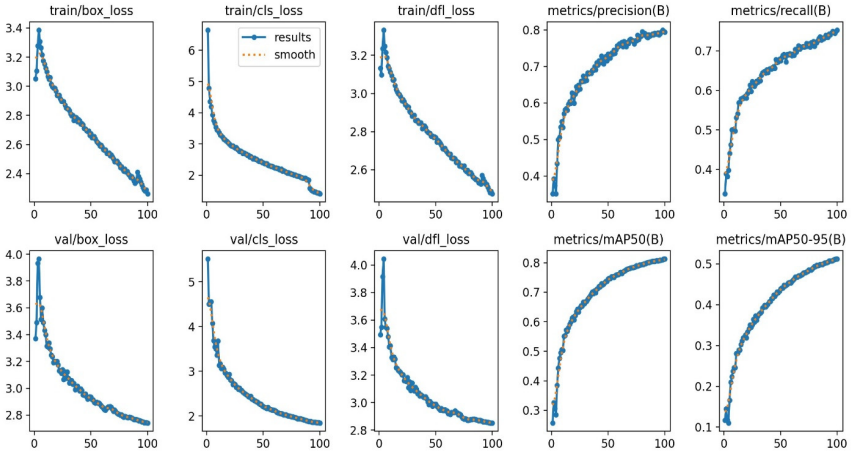
Models	**Epochs**	**Precision**	**Recall**	**mAP50**	**mAP50-95**	**Fitness**
YOLOv11n	50	0.766005	0.709466	0.7772	0.467005	0.498024
YOLOv11n	100	0.803112	0.758146	0.824964	0.514914	0.545919
YOLOv10n	50	0.740179	0.690027	0.755201	0.453826	0.483964
YOLOv10n	100	0.797005	0.749882	0.812536	0.512366	0.542383
YOLOv9t	50	0.74991	0.711167	0.773553	0.469511	0.499916
YOLOv9t	100	0.799071	0.737427	0.80877	0.505313	0.535659

Figure 5 presents a comparative analysis of YOLOv9, YOLOv10, and YOLOv11, evaluating their training and validation losses alongside key performance metrics precision, recall, and mean Average Precision (mAP) over 100 epochs. All three YOLO versions demonstrate a steady decrease in both training and validation losses, indicating improved accuracy in object detection and classification. YOLOv9 and YOLOv11 start with a classification training loss of approximately 2.5. Both models have an initial validation loss of 2.3 and 2.5, respectively, which decreases to around 1.0 by the end of the training period. In contrast, YOLOv10 starts with significantly higher initial losses, with a classification training loss of approximately 7.0 and a validation loss of 6.0, but it similarly converges to loss values close to 1.5 by the 100th epoch. This downward trend is particularly noticeable in the bounding box prediction loss (box_loss), reflecting enhanced accuracy in bounding box predictions for all models. Additionally, the classification loss (cls_loss) and distributional focal loss (dfl_loss) for all three versions decrease substantially, approaching zero by the final epoch, which signifies an improved focus on object regions during both training and validation phases. The validation loss closely mirrors the training loss across all models, indicating effective training without overfitting. In terms of performance metrics, precision and recall consistently improve for all versions throughout the training period, with YOLOv9 and YOLOv11 exhibiting slightly higher precision compared to YOLOv10, while recall remains nearly identical across all three versions. The mAP, evaluated at a 50% Intersection over Union (IoU) threshold and averaged from 50% to 90% IoU thresholds, initially shows YOLOv10 with a slightly lower mAP compared to YOLOv9 and YOLOv11. However, by the 100th epoch, YOLOv10 catches up, achieving mAP levels comparable to the other two versions, with all models reaching approximately 80% mAP during training

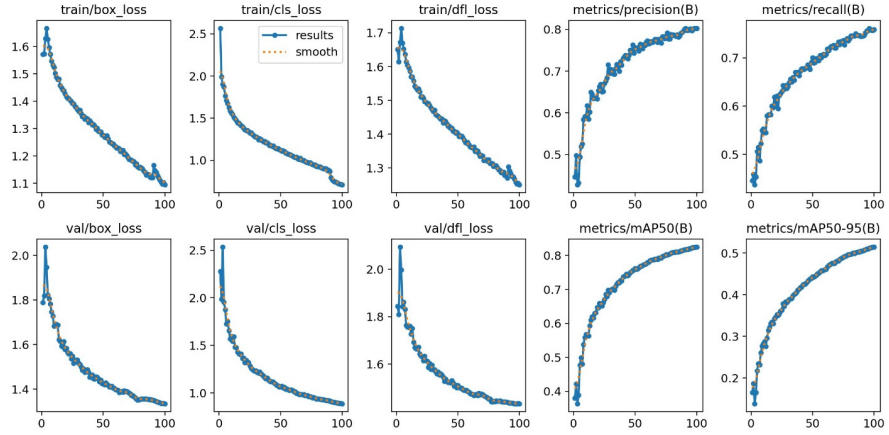
and validation. Overall, Figure 5 highlights that despite YOLOv10's higher initial loss values, all three YOLO versions exhibit similar convergence trends and achieve comparable final performance metrics by the end of 100 epochs. The consistent improvement in precision, recall, and mAP across all models underscores the robustness and effectiveness of each YOLO version in object detection tasks, while the alignment between training and validation losses confirms efficient training without overfitting, ensuring that the ultimate performance remains consistent despite differing starting points.



(**a**) YOLOv9



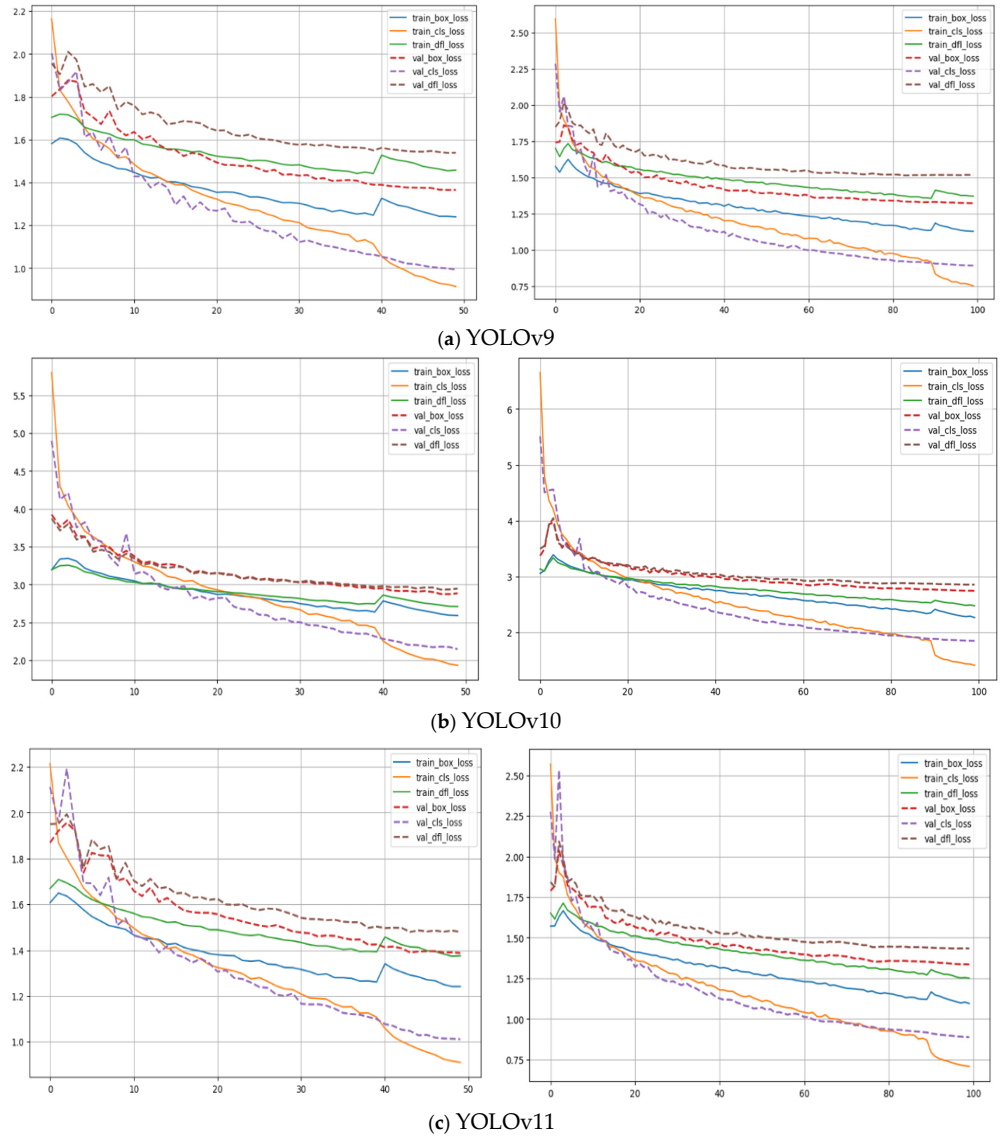
(**b**) YOLOv10



(**c**) YOLOv11

Figure 5. The training and validation losses of YOLOv9 (**a**), YOLOv10 (**b**), and YOLOv11 (**c**).

Figure 6 shows a comparative analysis between the 50 and 100 epochs of YOLOv9 (a), YOLOv10 (b), and YOLOv11 (c). The trends show that while there is a very slight change in the loss after 50 epochs, stability is maintained between the training and validation loss beyond this point. Up to 50 epochs, the difference suggests minor overfitting. Based on the results, we can conclude that although YOLO took slightly longer to train for 100 epochs, it achieved better test performance with minimal chance of overfitting. Overall, the results do not indicate any signs of underfitting.



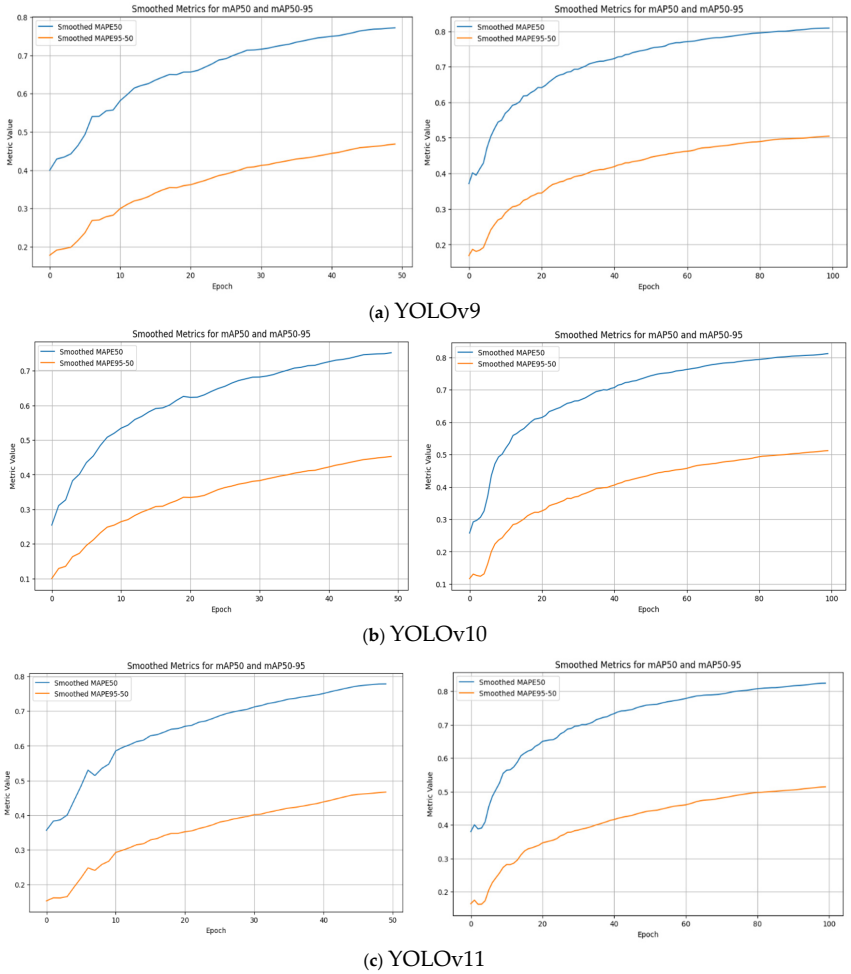
(**a**) YOLOv9

(**b**) YOLOv10

(**c**) YOLOv11

Figure 6. Comparative analysis between the 50 and 100 epochs of YOLOv9 (**a**), YOLOv10 (**b**), and YOLOv11 (**c**).

Figure 7 shows the smoothed metrics for mAP@50 and mAP@50-95 over 50 epochs and 100 epochs of YOLOv9 (a), YOLOv10 (b), and YOLOv11 (c). The 100-epoch plots yield higher mAP@50 and mAP@50-95 values than the 50-epoch plots, indicating that extended training improves detection performance. YOLOv11 (c) stabilizes at around 78% mAP after 50 epochs but reaches about 83% mAP at 100 epochs, demonstrating improved accuracy on test data.



(**a**) YOLOv9

(**b**) YOLOv10

(**c**) YOLOv11

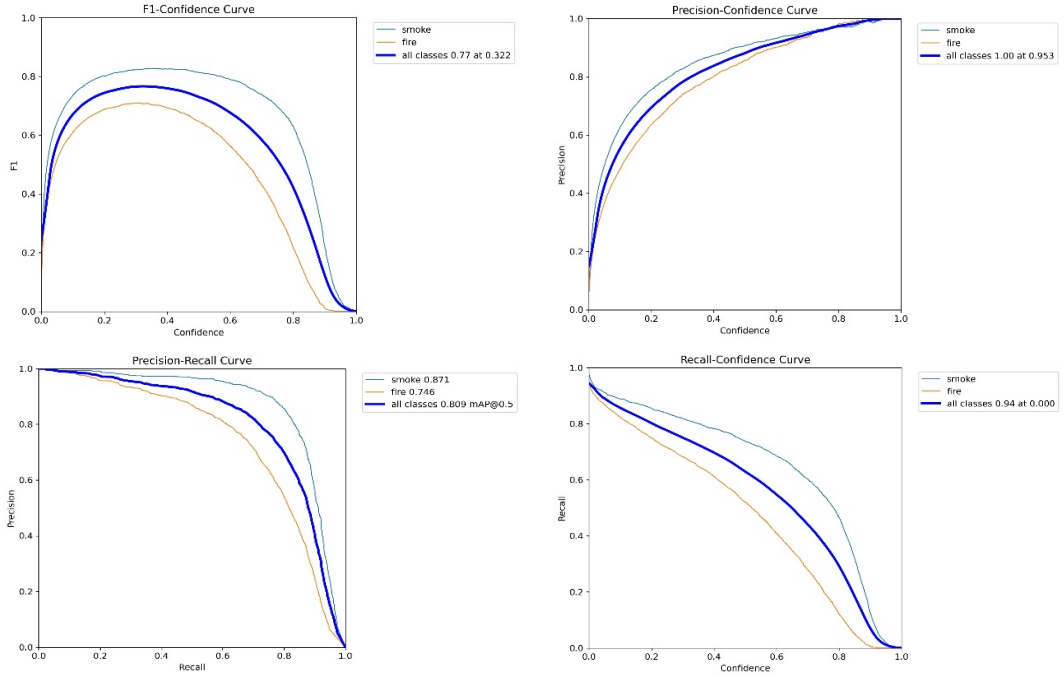
Figure 7. Smoothed metrics for mAP@50 and mAP@50-95 over 50 epochs and 100 epochs of YOLOv9 (**a**), YOLOv10 (**b**), YOLOv11 (**c**).

Training and validation losses, precision, and recall metrics are important for evaluating the model's behavior during training and for analyzing how well it is likely to perform on unseen data. However, noise in the data can bias the model during the training, Smoothed graphs, in particular, help in understanding the general trends of the model's behavior by reducing noise, making it easier to assess improvements and identify convergence points. Both YOLOv9 and YOLOv10 exhibit similar trends, with their mAP values steadily improving as training progresses. By 80 epochs, the curves start to smooth out, indicating stabilization of the models' learning. YOLOv10 achieves a higher smoothed mAP@50 value, reaching approximately 80% by the 100th epoch, whereas YOLOv9 performs slightly lower. Despite this difference, both models show consistent convergence patterns, with neither achieving perfect performance. YOLOv10's marginally higher mAP values suggest it may be better suited for precision-focused tasks. This analysis highlights the importance of monitoring these metrics to evaluate and compare the effectiveness of different versions in achieving reliable and accurate performance.

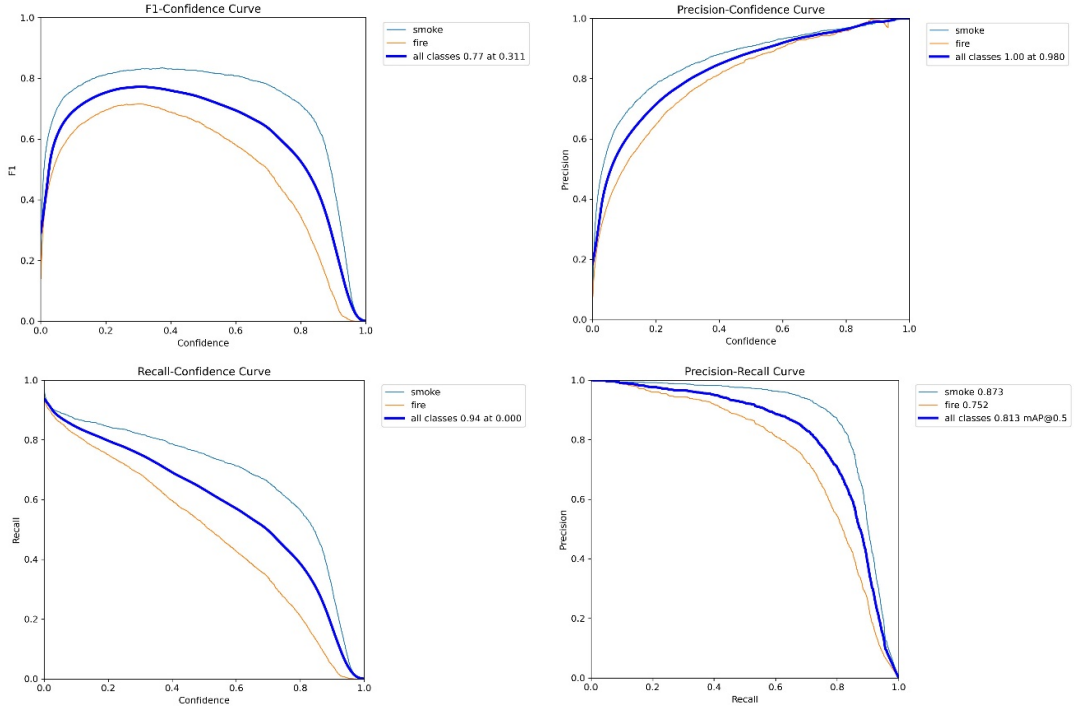
Although YOLO versions exhibit similar trends in training and validation behavior, YOLOv10 demonstrates superior smoothed performance metrics, making it the top performer among the three versions evaluated.

As observed in the previous graphs, as the number of epochs increases, the results stabilize, which is evident from the smoothing trends. However, it is worth incorporating statistical analysis, such as confidence intervals, to assess how reliable these training and validation results are. Figure 8 shows the confidence curve and highlights the relationship

between the model's confidence levels and its performance. Ideally, an efficient model should maintain high F1-scores consistently across all confidence levels. In practice, sometimes, the F1-scores exhibit a downward trend as confidence levels increase. This decline could stem from calibration issues, inherent model biases, or data quality limitations. The smooth curves in the graphs provide a clear indication of how confidence influences the system's behavior, offering a valuable tool to assess and refine model performance. YOLOv9 and YOLOv11 appear to be the top-performing models. YOLOv9 excels at lower confidence thresholds while YOLOv11 is better at higher confidence levels.

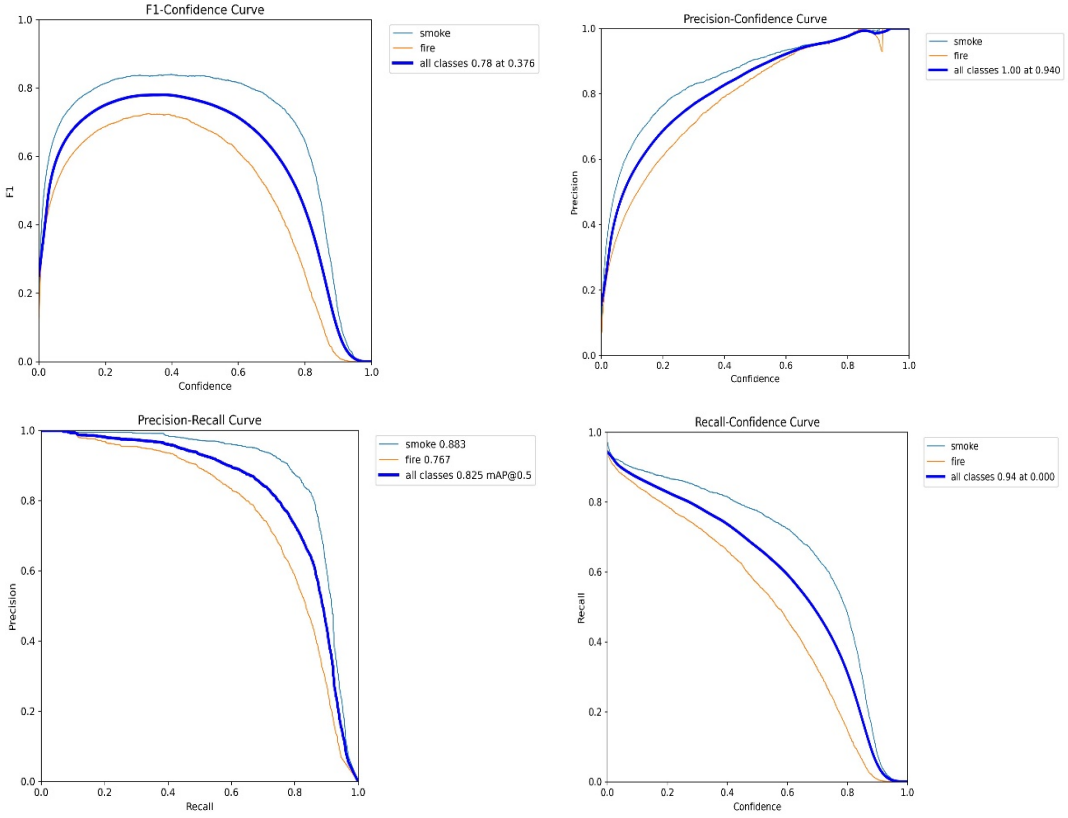


(**a**) YOLOv9



(**b**) YOLO 10

Figure 8. *Cont.*



(**c**) YOLOv11

Figure 8. Performance evaluation curves for YOLO models, including F1-Confidence, Precision-Confidence, Precision-Recall, and Recall-Confidence metrics, for smoke and fire detection using YOLOv9 (**a**), YOLOv10 (**b**), and YOLOv11 (**c**).

We have changed the parameter setting to the following as shown in Table 5.

Table 5. Changed training parameters of the YOLO models.

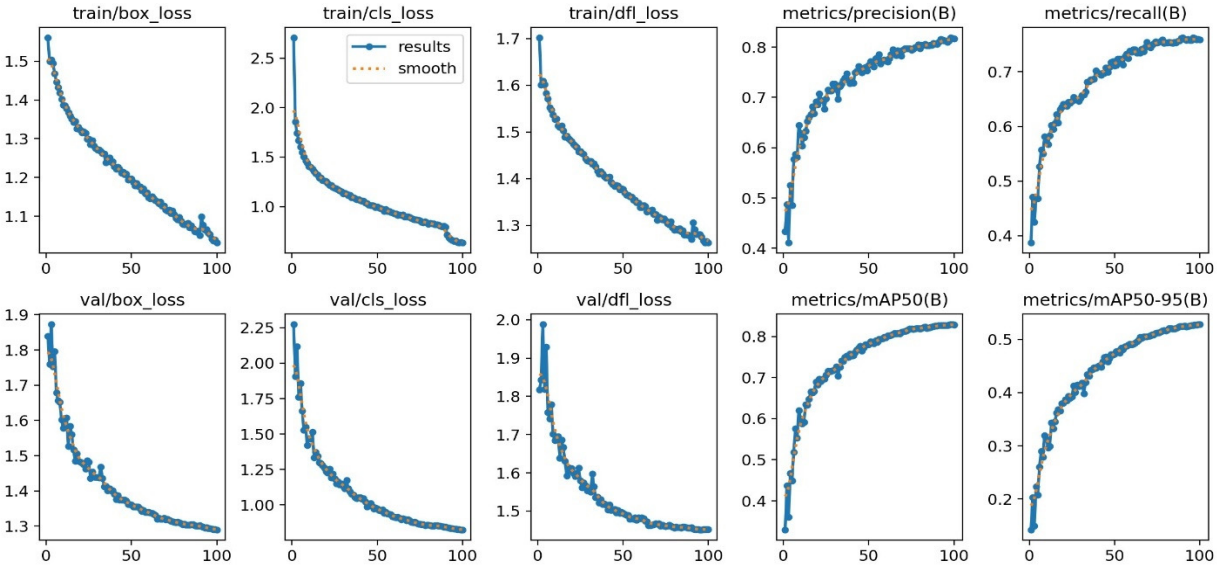
Parameter	Value
epochs	100
imgsz	640
batch	64
lr0	0.01
warmup_epochs	3.0
warmup_bias_lr	0.1
momentum	0.8
weight_decay	0.0005
optimizer	SGD
amp	True

The results of YOLO models YOLOv11n, YOLOv10n, and YOLOv9t models are shown in Table 6 YOLOv11n outperforms both YOLOv10n and YOLOv9t in terms of precision, recall, mAP50, and fitness. Additionally, YOLOv10n performs better than YOLOv9t across all metrics.

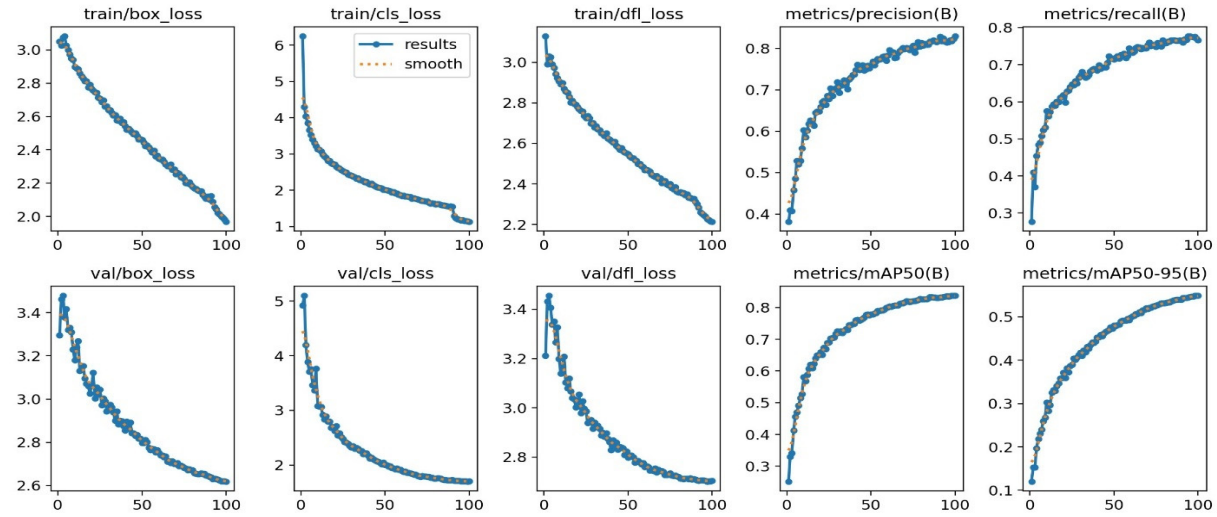
Table 6. Performance metrics of YOLOv11n, YOLOv10n, and YOLOv9t models at 100 epochs, showing precision, recall, mAP@50, mAP@50-95, and fitness score.

Models	Epochs	Precision	Recall	mAP50	mAP50-95	Fitness
YOLOv11n	100	0.845419	0.801229	0.859837	0.558822	0.588924
YOLOv10n	100	0.835622	0.763534	0.837255	0.548967	0.57779
YOLOv9t	100	0.817222	0.759236	0.828777	0.528051	0.558123

Figure 9 represents the training and validation loss graphical representation after the parameter changes. YOLOv11 achieves the most stable and lowest training and validation losses among all models (YOLOv9t, YOLOv10n, and YOLOv11n), indicating better convergence. YOLOv11 demonstrates high convergence metrics, including precision, recall, mAP50, and mAP50-95, outperforming both YOLOv9t and YOLOv10n. This suggests improved detection accuracy and robustness. YOLOv11 exhibits smoother and more consistent trends across losses and metrics, reflecting a well-optimized training process.

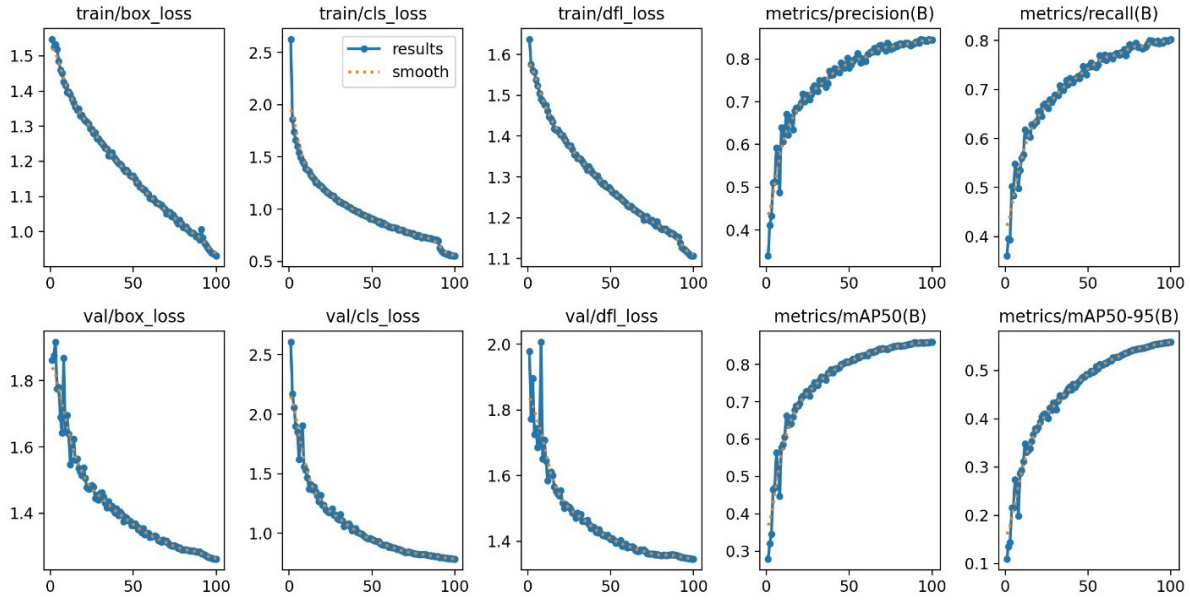


(**a**) YOLOv9t



(**b**) YOLOv10n

Figure 9. *Cont.*



(**c**) YOLOv11n

Figure 9. The training and validation losses of YOLOv9 (**a**), YOLOv10 (**b**), and YOLOv11 (**c**) after the parameter changes.

Figure 10 shows training and validation metrics on 100 epochs of YOLOv9 (a), YOLOv10 (b), and YOLOv11 (c) after parameter changes. In the graph, there is no significant difference between the training and validation patterns, indicating no clear signs of overfitting.

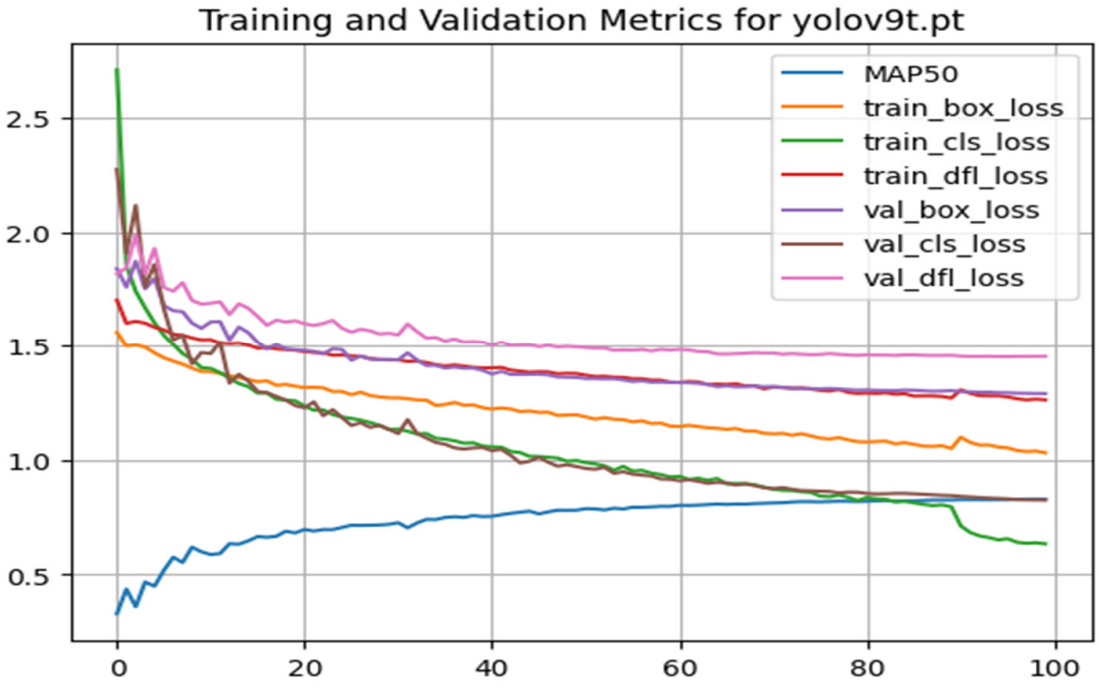


Figure 10. *Cont.*

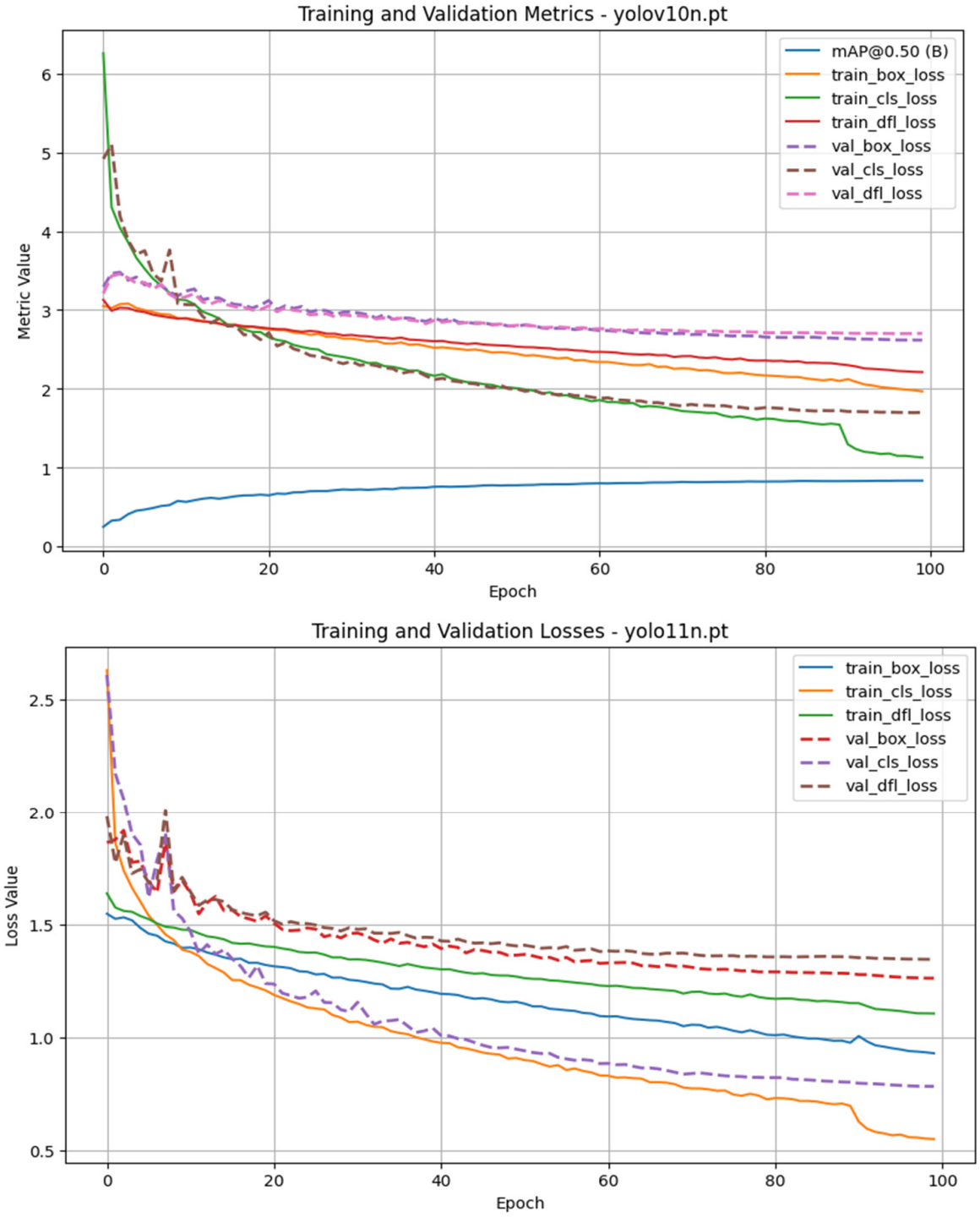


Figure 10. The training and validation metrics over 100 epochs of YOLOv9, YOLOv10, and YOLOv11 after the parameter changes.

Figure 11 shows smoothed metrics for mAP50 and mAP50-95 during training of YOLOv9, YOLOv10, and YOLOv11 after the parameter changes.

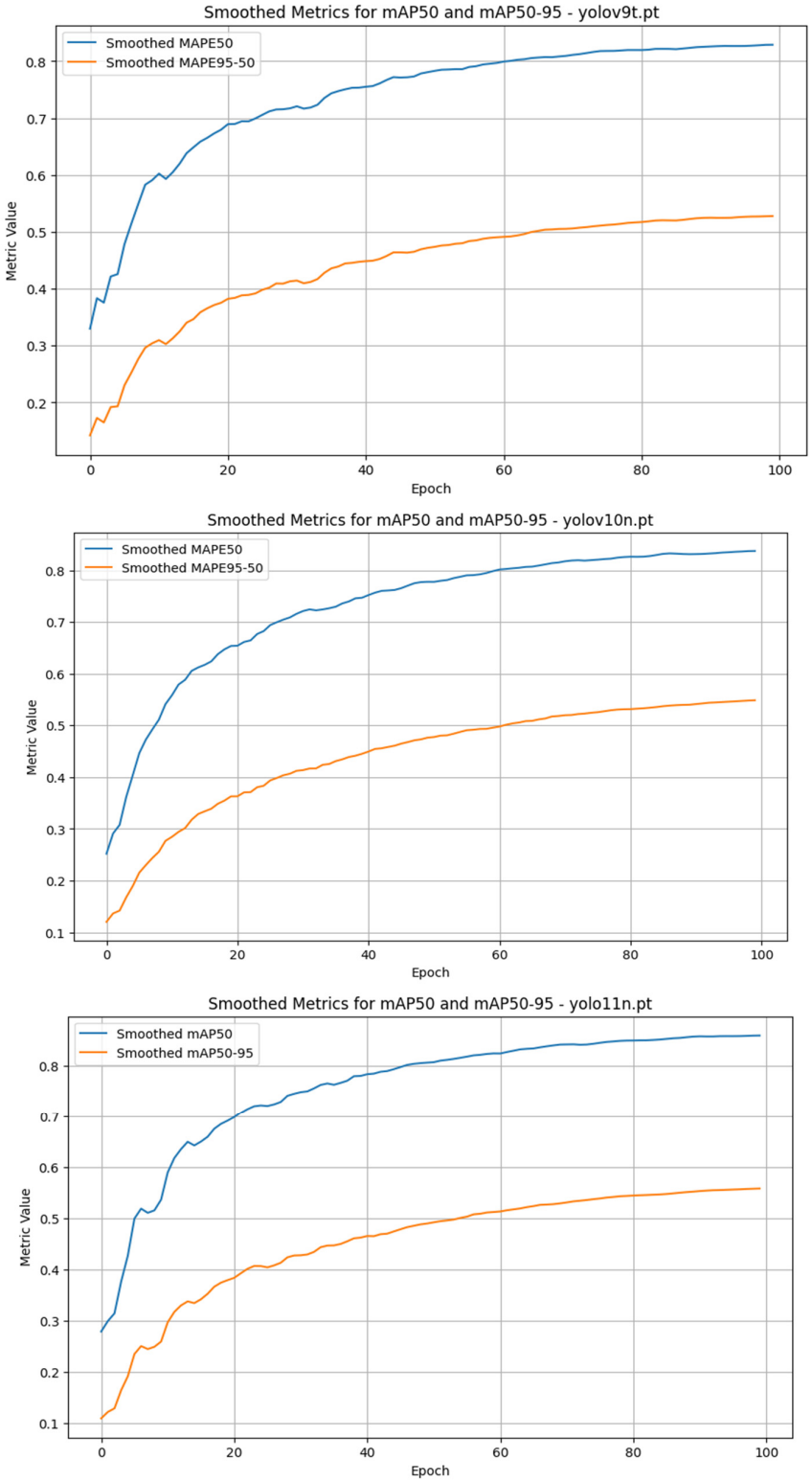
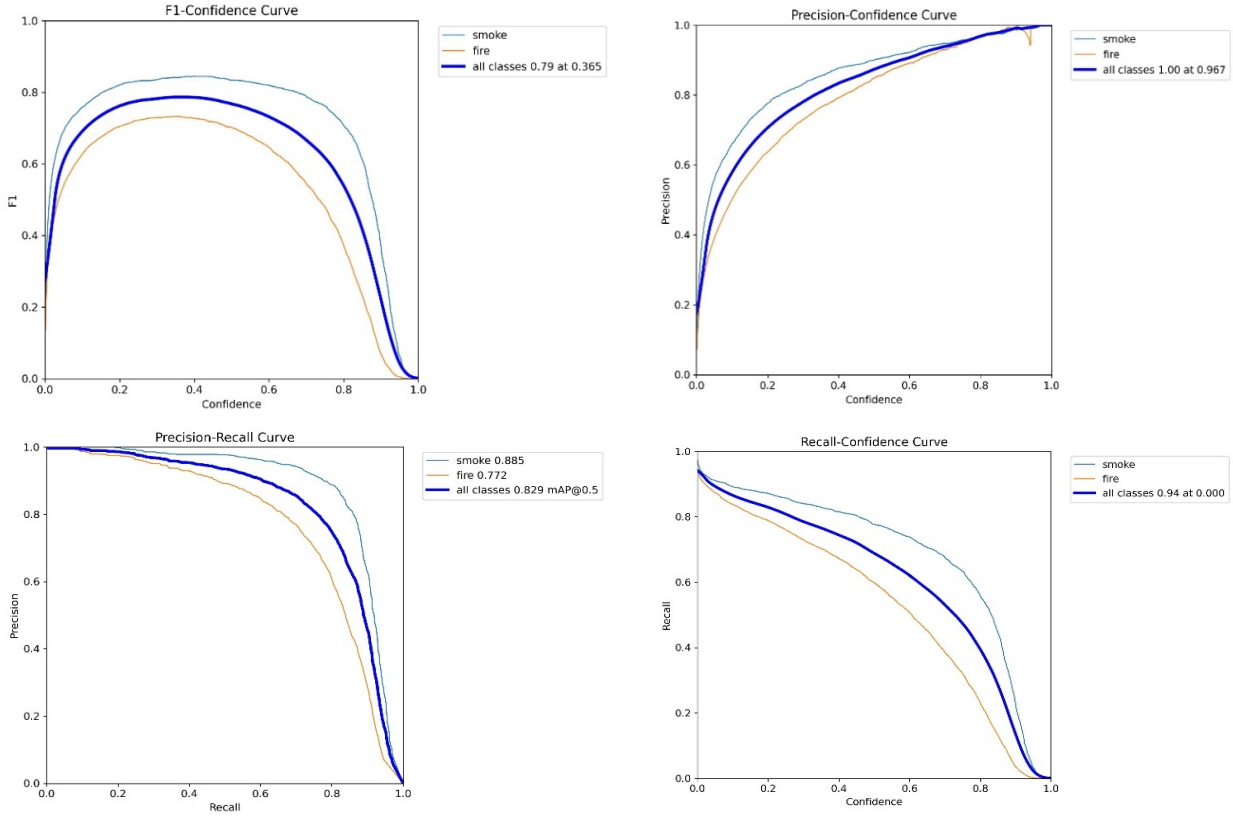


Figure 11. The smoothed metrics for mAP50 and mAP50-95 during the training of YOLOv9, YOLOv10, and YOLOv11 after parameter changes.

All three models show an upward trend in both mAP50 and mAP50-95, indicating performance improvement as training progresses. YOLOv11n significantly outperforms YOLOv10n and YOLOv9t in mAP50 and mAP50-95, demonstrating superior object detection performance and faster convergence.

Figure 12 shows the performance evaluation curves for YOLOv9, YOLOv10, and YOLOv11, illustrating F1-Confidence, Precision-Confidence, Precision-Recall, and Recall-Confidence metrics for smoke and fire detection. These figures show the various aspects of the model performance over 100 epochs after the parameter changes. All the three models demonstrate high F1-scores and precision at optimal confidence thresholds. However, a trade-off between precision and recall is observed, particularly for the smoke class. While YOLOv10 shows promising results, YOLOv11 appears to offer slightly better performance, especially in terms of average F1-score and precision.



(**a**) YOLOv9

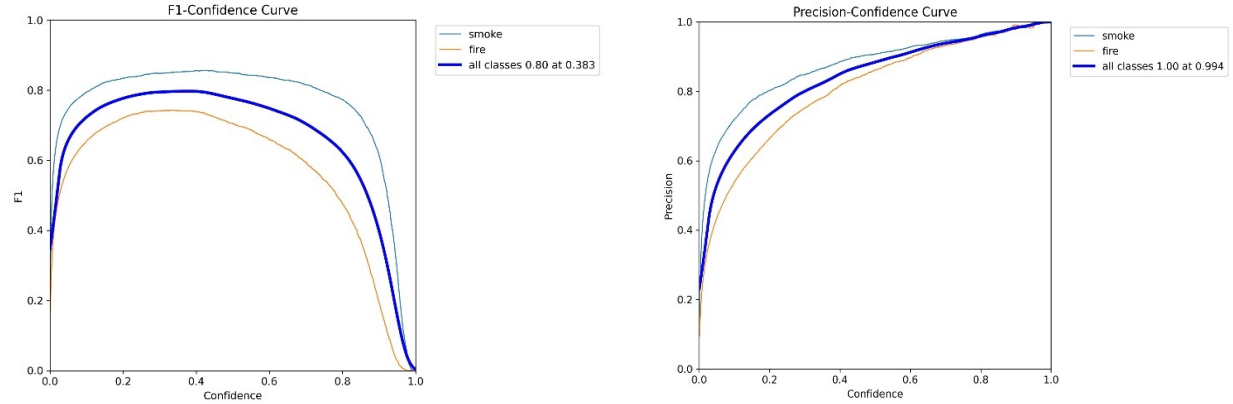
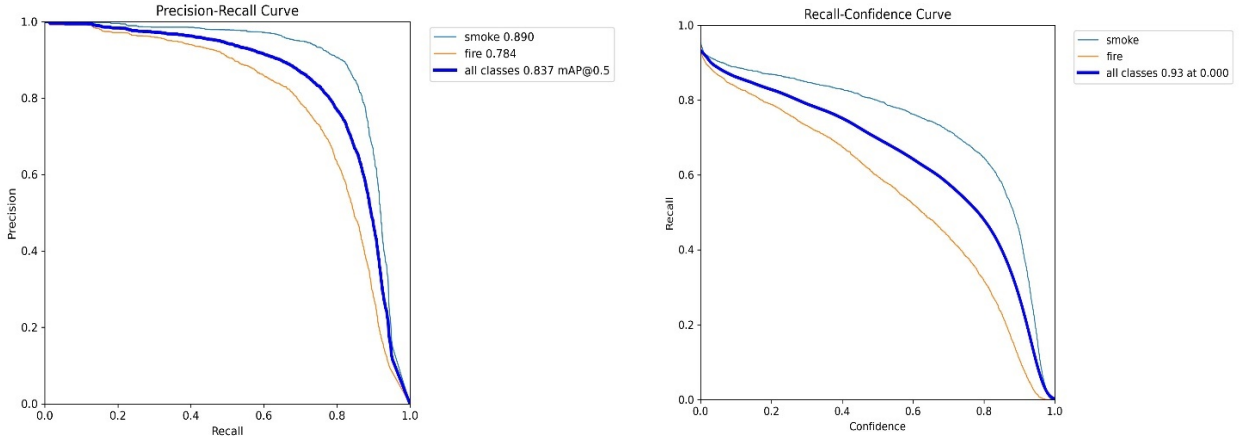
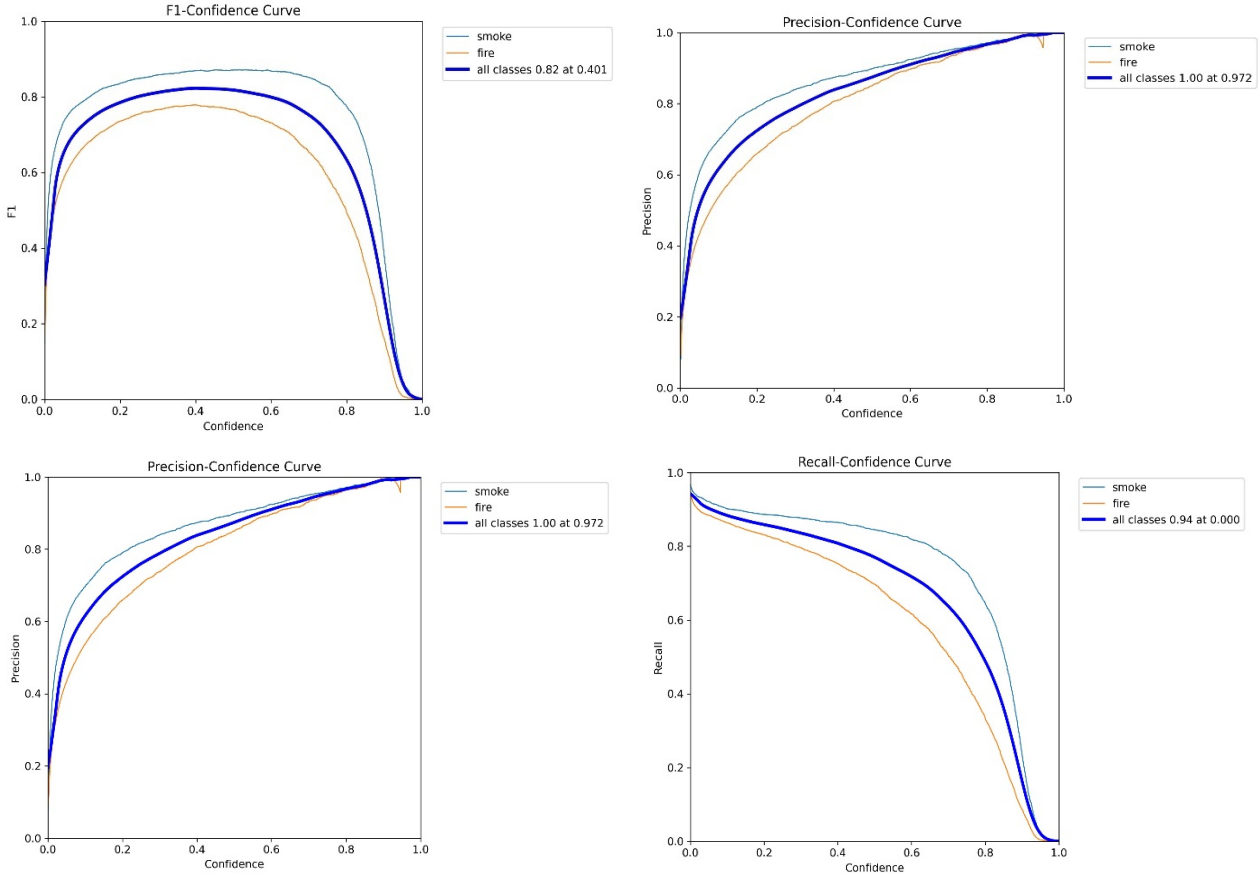


Figure 12. *Cont.*



(**b**) YOLOv10



(**c**) YOLOv11

Figure 12. Performance evaluation curves for YOLO models, including F1-Confidence, Precision-Confidence, Precision-Recall, and Recall-Confidence metrics, for smoke and fire detection using YOLOv9 (**a**), YOLOv10 (**b**), and YOLOv11 (**c**) after parameter changes.

The comparison of YOLOv11n and YOLOv11x models with other studies using the smoke and fire dataset and D-Fire dataset are shown in Table 7. It shows that YOLOv11x, (56.9 M parameters) the largest model in YOLOv11, offers more accuracy, while YOLOv11n (2.6 M parameters), a lightweight model offers the fastest inference. For 300 epochs, we utilized the NVIDIA RTX A1000, Intel(R) Xeon(R) CPU @ 2.20 GHz, and A100 GPU. We used the same settings as in Table 5 but reduced the image size to 416 to avoid out-of-memory errors.

Table 7. Comparisons of YOLOv11n and YOLOv11x with other studies.

	Dataset	Precision	Recall	mAP50	mAP 50-95	Epochs
modified YOLOv5 [28]	Smoke and Fire	0.892	0.827	0.873	0.566	500
YOLOv3-tiny [28]		0.834	0.762	0.814	0.437	500
YOLOv5n [28]		0.812	0.79	0.823	0.475	500
YOLOv5s-M [28]		0.778	0.688	0.762	0.4	500
YOLOv5s-S [28]		0.774	0.72	0.759	0.387	500
YOLOv5s [28]		0.87	0.827	0.863	0.559	500
YOLOv8x [9]		0.837	**0.952**	**0.890**	-	300
YOLOv11n (proposed)		**0.87**	0.802	0.866	0.594	300
YOLOv11x (proposed)		**0.904**	0.816	**0.889**	**0.709**	300
YOLOv11n (proposed)	D-Fire	0.772	0.718	0.78	0.458	100
YOLOv11x (proposed)		**0.795**	**0.743**	**0.78**	0.47	100
ESFD-YOLOv8n [37]		**80.1**	72.7	**79.4**	-	300
YOLOv10n [37]		78.3	69.5	77.1	-	300
YOLOv9 [37]		78.8	70.3	78.0	-	300
YOLOv8n [37]		77.8	70.0	77.4	-	300
YOLOv5n [37]		78.3	69.7	76.7	-	300
YOLOv6n [37]		77.8	69.8	77.6	-	300

The results in Table 7 demonstrate the performance of various YOLO models, including existing state-of-the-art models and the adapted YOLO models on the smoke and fire dataset and D-Fire dataset. Among these, the proposed YOLOv11x model outperforms with the highest precision of 90.4%, representing its capability to make highly accurate predictions by minimizing false positives. It also achieves a competitive mAP50 of 88.9%, closely following YOLOv8x [9], which recorded the highest at 89.0%. Furthermore, YOLOv11x leads in the mAP50-95 metric with a score of 70.9%, showing its robustness across varying IoU thresholds. However, the proposed YOLOv11x model lags in recall (81.6%) compared to YOLOv8x [9], which achieved the highest recall of 95.2%, indicating that YOLOv11x is more likely to miss some true positives. This trade-off suggests that while YOLOv11x excels in precision and overall robustness, its lower recall could limit its effectiveness in scenarios where it fails to detect all important instances. Despite this shortcoming, the proposed model demonstrates significant advantages over several prior models, particularly in precision and mAP50-95 with 300 epochs compared to 500. It concluded that, despite limited resources, we managed to make our model effectively detect the instances. Similarly, in the D-Fire dataset, the proposed YOLOv11x competes with other robust models, such as ESFD-YOLOv8n [37]. YOLOv11x achieved the highest recall of 0.743 among all models, with a precision of 0.795 and a mAP@50 of 0.78%. These results are comparable to the ESFD-YOLOv8n [37], which achieved a mAP@50 of 79.4%, without introducing any hybrid mechanisms or additional layers to enhance performances.

Figure 13 shows examples of fire detections of YOLOv9, YOLOv10, and YOLOv11 in various scenes, displaying the identification and labeling with confidence scores for each detected instance. Here, one can discuss the results frame by frame to analyze the performance of the models in detail.



(**a**) YOLOv9



(**b**) YOLOv10

Figure 13. *Cont.*



(**c**) YOLOv11

Figure 13. Examples of fire detections of YOLOv9 (**a**), YOLOv10 (**b**), and YOLOv11 (**c**).

In the first frame, we can see fire covering the entire forest. YOLOv9 captured only a limited area of the fire, whereas YOLOv10 and YOLOv11 marked two distinct points as fire. It is good to know that fire was detected by all models. Both YOLOv10 and YOLOv11 show excellent performance in detecting both fire and smoke in the second frame, with YOLOv10 being more precise in smoke detection. This highlights the models' differing capabilities in distinguishing and identifying fire-related features. This frame-by-frame analysis demonstrates the strengths and limitations of each model in various scenarios.

There are instances where YOLOv11 fails to detect or correctly classify objects, as shown in Figure 14, which highlights examples of false negatives and false positives. For false negatives: (a) the model fails to detect smoke in the area; (b) certain occurrences of fire are not labeled; and (c) another instance of fire goes undetected. For false positives: (d) a detection is incorrectly labeled as fire where no fire exists; (e) car headlights are mistakenly identified as fire; and (f) fire is erroneously classified as smoke.

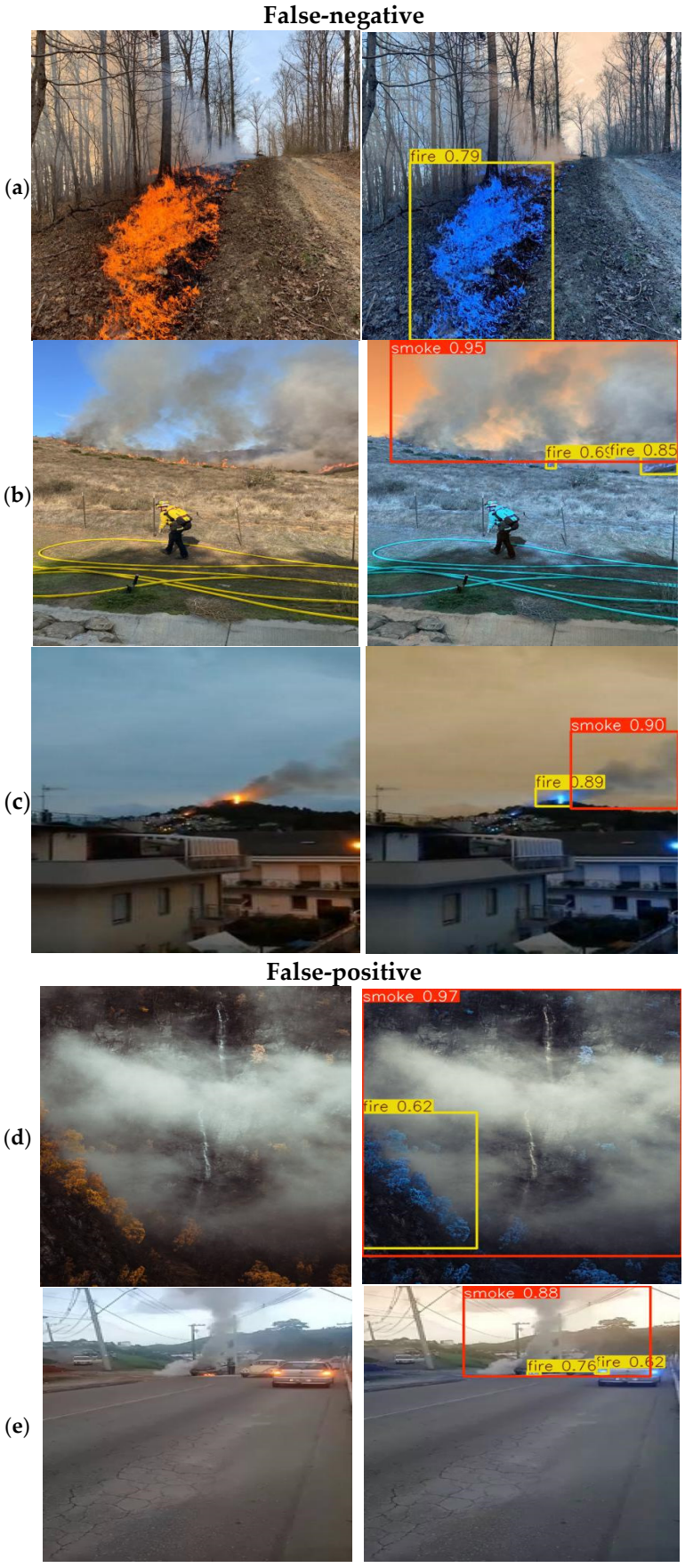


Figure 14. *Cont.*

(**f**)



Figure 14. Examples of false negatives (**a**, **b**, **c**) and false positives (**d**, **e**, **f**).

5. Conclusions

This study evaluates the performance of YOLOv9, YOLOv10, and YOLOv11 in detecting smoke and fire in forest environments. YOLOv11n stands out with the highest precision (0.845), recall (0.801), mAP@50 (0.859), and mAP@50-95 (0.558), showing high accuracy across various lighting conditions and complex forest backgrounds for the smoke and fire dataset.

The model's training and validation loss graphs reveal a consistent downward trend across YOLO versions, indicating effective learning without overfitting. All models achieved near-similar final loss values below 1, reflecting their reliability in bounding boxes and classification predictions. Decreases in classification loss suggest a more precise focus on object regions, with YOLOv11 achieving particularly high-performance metrics. Improved precision and recall over 100 epochs illustrate that all three models steadily enhanced their capability to detect objects accurately. The mAP scores (both mAP@50 and mAP@50-95) reflect strong object detection performance across different overlap thresholds, reaching approximately 84%, which underscores the models' high accuracy during testing. Overall, YOLOv11n outperformed YOLOv9t and YOLOv10n in detecting both smoke and fire. The extra-large version of YOLOv11 (YOLOv11x) was also evaluated on the Smoke and Fire dataset and the D-Fire dataset to maximize accuracy and performance. The results demonstrate that YOLOv11x delivers high performance, making it highly competitive with other models used in similar studies. For future work, we plan to explore a broader range of models and transformer architectures, such as RT-DETR, to enhance smoke and fire detection in challenging forest environments.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

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 author declares there is no conflict of interest.

References

1. Martínez, J.R.M.; Machuca, M.H.; Díaz, R.Z.; y Silva, F.R.; González-Cabán, A. Economic losses to Iberian swine production from forest fires. *For. Policy Econ.* **2011**, *13*, 614–621. [CrossRef]

2. Chaturvedi, S.; Khanna, P.; Ojha, A. A survey on vision-based outdoor smoke detection techniques for environmental safety. *ISPRS J. Photogramm. Remote Sens.* **2022**, *158*, 158–187. [CrossRef]
3. 3 Billion Animals Harmed by Australia's Fires. Available online: https://www.worldwildlife.org/stories/3-billion-animals-harmed-by-australia-s-fires (accessed on 30 October 2024).
4. Annual Area Burnt by Wildfires. Available online: https://ourworldindata.org/grapher/annual-area-burnt-by-wildfires (accessed on 30 October 2024).
5. National Centers for Environmental Information. Annual 2023 Wildfires Report. Available online: https://www.ncei.noaa.gov/access/monitoring/monthly-report/fire/202313 (accessed on 30 October 2024).
6. Chan, C.C.; Alvi, S.A.; Zhou, X.; Durrani, S.; Wilson, N.; Yebra, M. IoT Ground Sensing Systems for Early Wildfire Detection: Technologies, Challenges and Opportunities. *arXiv* **2023**, arXiv:2312.10919. [CrossRef]
7. European Civil Protection and Humanitarian Aid Operations ECHO Daily Flash—Kazakhstan. Available online: https://www.ncei.noaa.gov/access/monitoring/monthly-report/fire/202313 (accessed on 30 October 2024).
8. Al-Smadi, Y.; Alauthman, M.; Al-Qerem, A.; Aldweesh, A.; Quaddoura, R.; Aburub, F.; Mansour, K.; Alhmiedat, T. Early wildfire smoke detection using different yolo models. *Machines* **2023**, *11*, 2. [CrossRef]
9. Chetoui, M.; Akhloufi, M.A. Chetoui. Fire and Smoke Detection Using Fine-Tuned YOLOv8 and YOLOv7 Deep Models. *Fire* **2024**, *7*, 4. [CrossRef]
10. Wang, X.; Li, H.; Yue, X.; Meng, L. A comprehensive survey on object detection YOLO. In Proceedings of the 5th International Symposium on Advanced Technologies and Applications in the Internet of Things (ATAIT 2023), Kusatsu, Japan, 28–29 August 2023.
11. Redmon, J. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016.
12. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.
13. Redmon, J. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
14. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
15. Jocher, G.; Stoken, A.; Borovec, J.; Changyu, L.; Hogan, A.; Diaconu, L.; Ingham, F.; Poznanski, J.; Fang, J.; Yu, L.; et al. ultralytics/yolov5: v3. 1-bug fixes and performance improvements. *Zenodo* **2020**. [CrossRef]
16. Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W.; et al. YOLOv6: A single-stage object detection framework for industrial applications. *arXiv* **2022**, arXiv:2209.02976.
17. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of the IEEE/Cvf Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 7464–7475.
18. Jocher, G.; Chaurasia, A.; Qiu, J. Yolo by Ultralytics, Version 8.0.0; Ultralytics. 2023. Available online: https://ultralytics.com (accessed on 30 October 2024).
19. Wang, C.Y.; Yeh, I.H.; Liao, H.Y.M. Yolov9: Learning what you want to learn using programmable gradient information. *arXiv* **2024**, arXiv:2402.13616.
20. Sapkota, R.; Qureshi, R.; Calero, M.F.; Badjugar, C.; Nepal, U.; Poulose, A.; Zeno, P.; Vaddevolu, U.B.P.; Khan, S.; Shoman, M.; et al. YOLOv10 to Its Genesis: A Decadal and Comprehensive Review of The You Only Look Once (YOLO) Series. *arXiv* **2024**, arXiv:2406.19407.
21. Jocher, J.; Qiu, J. *Ultralytics YOLO11*; GitHub: San Francisco, CA, USA, 2024.
22. Talaat, F.M.; ZainEldin, H. An improved fire detection approach based on YOLO-v8 for smart cities. *Neural Comput. Appl.* **2023**, *35*, 20939–20954. [CrossRef]
23. Cao, L.; Shen, Z.; Xu, S. Efficient forest fire detection based on an improved YOLO model. *Vis. Intell.* **2024**, *2*, 20. [CrossRef]
24. Bahhar, C.; Ksibi, A.; Ayadi, M.; Jamjoom, M.M.; Ullah, Z.; Soufiene, B.O.; Sakli, H. Wildfire and smoke detection using staged YOLO model and ensemble CNN. *Electronics* **2023**, *12*, 228. [CrossRef]
25. Casas, E.; Ramos, L.; Bendek, E.; Rivas-Echeverría, F. Assessing the effectiveness of YOLO architectures for smoke and wildfire detection. *IEEE Access* **2023**, *11*, 96554–96583. [CrossRef]
26. Pan, W.; Wang, X.; Huan, W. EFA-YOLO: An Efficient Feature Attention Model for Fire and Flame Detection. *arXiv* **2024**, arXiv:2409.12635.
27. Catargiu, C.; Cleju, N.; Ciocoiu, I.B. A Comparative Performance Evaluation of YOLO-Type Detectors on a New Open Fire and Smoke Dataset. *Sensors* **2024**, *24*, 5597. [CrossRef] [PubMed]
28. 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]
29. YOLOv9: A Leap Forward in Object Detection Technology. Available online: https://docs.ultralytics.com/models/yolov9/ (accessed on 30 October 2024).

30. Yaseen, M. What is YOLOv9: An In-Depth Exploration of the Internal Features of the Next-Generation Object Detector. *arXiv* **2024**, arXiv:2409.07813.
31. Wang, A.; Chen, H.; Liu, L.; Chen, K.; Lin, Z.; Han, J.; Ding, G. Yolov10: Real-time end-to-end object detection. *arXiv* **2024**, arXiv:2405.14458.
32. YOLOv10: Real-Time End-to-End Object Detection. Available online: https://docs.ultralytics.com/models/yolov10/ (accessed on 30 October 2024).
33. Sapkota, R.; Meng, Z.; Karkee, M. Synthetic Meets Authentic: Leveraging LLM Generated Datasets for YOLO11 and YOLOv10-Based Apple Detection through Machine Vision Sensors. *Smart Agric. Technol.* **2024**, *9*, 100614. [CrossRef]
34. Hussain, M.; Khanam, R. In-depth review of yolov1 to yolov10 variants for enhanced photovoltaic defect detection. *Solar* **2024**, *4*, 351–386. [CrossRef]
35. Khanam, R.; Hussain, M. YOLOv11: An Overview of the Key Architectural Enhancements. *arXiv* **2024**, arXiv:2410.17725.
36. Ultralytics YOLO11. Available online: https://docs.ultralytics.com/models/yolo11/#how-do-i-train-a-yolo11-model-for-object-detection (accessed on 30 October 2024).
37. Mamadaliev, D.; Touko, P.L.M.; Kim, J.-H.; Kim, S.-C. Esfd-yolov8n: Early smoke and fire detection method based on an improved yolov8n model. *Fire* **2024**, *7*, 303. [CrossRef]

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.