

Article

YOLO-Based Models for Smoke and Wildfire Detection in Ground and Aerial Images

Leon Augusto Okida Gonçalves , Rafik Ghali  and Moulay A. Akhloufi * 

Perception, Robotics and Intelligent Machines (PRIME), Department of Computer Science, Université de Moncton, Moncton, NB E1A 3E9, Canada; laog19@inf.ufpr.br (L.A.O.G.); rafik.ghali@umoncton.ca (R.G.)

* Correspondence: moulay.akhloufi@umoncton.ca

Abstract: Wildland fires negatively impact forest biodiversity and human lives. They also spread very rapidly. Early detection of smoke and fires plays a crucial role in improving the efficiency of firefighting operations. Deep learning techniques are used to detect fires and smoke. However, the different shapes, sizes, and colors of smoke and fires make their detection a challenging task. In this paper, recent YOLO-based algorithms are adopted and implemented for detecting and localizing smoke and wildfires within ground and aerial images. Notably, the YOLOv7x model achieved the best performance with an mAP (mean Average Precision) score of 80.40% and fast detection speed, outperforming the baseline models in detecting both smoke and wildfires. YOLOv8s obtained a high mAP of 98.10% in identifying and localizing only wildfire smoke. These models demonstrated their significant potential in handling challenging scenarios, including detecting small fire and smoke areas; varying fire and smoke features such as shape, size, and colors; the complexity of background, which can include diverse terrain, weather conditions, and vegetation; and addressing visual similarities among smoke, fog, and clouds and the the visual resemblances among fire, lighting, and sun glare.

Keywords: fire; smoke; object detection; YOLO; deep learning; wildfire



Citation: Gonçalves, L.A.O.; Ghali, R.; Akhloufi, M.A. YOLO-Based Models for Smoke and Wildfire Detection in Ground and Aerial Images. *Fire* **2024**, *7*, 140. <https://doi.org/10.3390/fire7040140>

Academic Editors: Kunpeng Yi and Shuai Yin

Received: 26 January 2024

Revised: 30 March 2024

Accepted: 11 April 2024

Published: 14 April 2024



Copyright: © 2024 by the authors. 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

Wildfires are a complex phenomenon and present a great challenge for forest management. They happen naturally and are important to keep the forest's health and biodiversity. However, they have great destructive potential, threatening communities and destroying natural resources. For example, wildfires have burned 260,000 hectares in the European Union since January 2023 [1]. In Canada, since 1990, fires have consumed an average of 2.5 million hectares per year [2]. The impact of wildfires on communities, such as evacuations and loss of homes, is expected to increase in the next years as more people move to residential developments in forested areas, coupled with the effects of climate change [3,4]. Therefore, it is of the utmost importance to control natural fires so that it is possible to take advantage of their ecological benefits while limiting their damage and costs [2].

Forests are full of trees, wood, and leaves, which serve as fuel for fires. This allows wildfires to spread very quickly, easily becoming uncontrollable. Therefore, it is necessary to detect fires quickly, giving enough reaction time to manage them [5]. Numerous firefighting solutions have been developed, including predictive and simulation techniques based on machine learning to reduce forest fire damage. They offer accurate and early wildfire information as they can detect the presence of forest smoke and fires at early stages. They can also determine the severity of burnt forest areas, improving reforestation efforts and providing reliable strategies for monitoring the ecosystem. There are a number of techniques to detect fires. One traditional method involves using physical sensors in order to detect smoke and fires. However, this method has some weaknesses. It has an inherent delay until the smoke hits the sensor [6]. They also suffer from high false-alarm rates and it is difficult to cover the forest area with sensors. Other traditional methods of fire detection,

employed by authorities, are the usage of observers in patrols or watch towers and aerial and satellite monitoring [5]. However, people working on patrols or monitoring are prone to being distracted or may not see the fires on time, leading to delays in detecting wildfires.

Recently, computer vision methods have been employed by researchers to automatically detect wildfires using aerial and ground images, addressing the aforementioned limitations [7,8]. These methods can be divided into two categories: flame detection and smoke detection, that is, to detect wildfires either by focusing on detecting their flames or by identifying and localizing the presence of smoke.

Early studies used hand-crafted feature extraction techniques to obtain smoke and flame features, such as colors, shapes, textures, edges, etc., from images and videos [9–13]. The downside of the traditional computer vision approaches with handcrafted feature extraction is the requirement to select the relevant features. Also, along with the choice of features, it is necessary to fine-tune the parameters used. This can become very cumbersome and inefficient [14], not to mention that the chosen rules and features might not be the best or generalized enough. This greatly limits the potential of these algorithms.

More recently, deep learning models, including Convolutional Neural Networks (CNNs) and vision transformers, have been applied to detect and segment smoke and wildfires using aerial, ground, and satellite images [15]. These models have shown their ability to automatically extract relevant features from the raw pixels of images, overcoming the aforementioned limitations.

These models are also used to perform three main tasks: fire classification, fire detection, and fire segmentation. Fire classification is the task of identifying the presence of fire and/or smoke in the input images. Fire detection, along with identifying fire and/or smoke objects, determines their spatial location by drawing a bounding box around them. Fire segmentation consists of identifying and delineating fire and/or smoke objects by assigning each object the same color in the visual presentation. There is a focus on the study of CNNs for the fire detection task. There are many object detection algorithms, and among them, the YOLO (You Only Look Once) family stands out. It offers a good balance between speed and accuracy, possessing the ability to make predictions in real time. Since 2016, the YOLO family has expanded with many different versions [16]. Additionally, YOLO models have been used for many applications, such as the detection of pedestrians [17], autonomous vehicles [18,19], face detection [20], security applications like the detection of weapons [21], and medical diagnostics [22–24].

One of the main challenges of fire detection methods is their difficulty in accurately detecting small areas of fire and smoke [25]. In this paper, recent object detection techniques, specifically YOLO models, are employed for detecting and localizing smoke and wildfires as well as addressing this challenge and improving smoke and wildfire detection tasks. Moreover, there is a particular focus on the detection of smoke, as it is usually visible before the flames, allowing faster fire detection and response [9]. The detection of smoke is a very challenging task as it takes many different shapes, sizes, and colors. The characteristics of smoke vary considerably depending on the surrounding forest environment. Also, it is difficult to distinguish smoke from smoke-like objects, such as clouds and fog. Another factor is that early smoke is usually semitransparent, resulting in blurred boundaries and making it hard to draw a precise bounding box around it. The variety and complexity of backgrounds in the images with fires also contribute to making this task challenging [26,27]. This, combined with the difficulty of detecting small objects, complicates the detection of early smoke. There are also many limitations associated with the task of the detection of wildfires. Similar to the detection of smoke, the complexity of the background in the images of fire is a factor. Early fires appear in small sizes, presenting a challenge for fire detection models. Fires show many different sizes, shapes, and intensities. Also, there are fire-like objects, such as glares, the reflection of the sun, and lighting.

The main contributions of this paper are:

- The recent YOLO models are adopted and implemented in detecting and localizing smoke and wildfires using ground and aerial images, thereby reducing false

detection and improving the performance of deep learning-based smoke and fire detection methods;

- The reliability of YOLO models is shown using two public datasets, D-Fire and WSDY (WildFire Smoke Dataset YOLO). Extensive analysis confirms their performance over baseline fire detection methods;
- YOLO models showed a robust potential to address challenging limitations, including background complexity; detecting small smoke and fire zones; varying smoke and fire features regarding intensity, flow pattern, shape, and colors; visual similarity between fire, lighting, and sun glare; and the visual resemblances among smoke, fog, and clouds;
- YOLO models are introduced in this study, achieving fast detection times, which are useful for real-time fire detection and early fire ignition. This shows the reliability of YOLO models when used on wildland fire monitoring systems. They can also help enhancing wildfire intervention strategies and reducing fire spread and the area of burnt forest, thus providing effective protection for ecosystem and human communities.

The rest of this paper is structured as follows: Section 2 reviews the previous research on smoke and flame detection using deep learning techniques. Section 3 introduces the recent YOLO models, wildland fire challenges, the dataset used, and the evaluation metrics. In Section 4, the testing results are discussed. Finally, Section 5 summarizes the paper.

2. Related Works

Wildfires pose a significant threat to human life and human health as well as the threat of economic losses. They also affect forest biodiversity and impact ecosystems. Detecting fire ignition and wildfires early is a crucial aspect in reducing their damage. First, early prediction and identification of wildland fires allows rapid firefighter interventions, enabling the evacuation of habitats and reducing habitat loss. This prevents the destruction of habitat and assures ecological balance. On the other hand, early intervention can also reduce fire intensity and spread. Consequently, it reduces carbon emissions, as wildfires are important sources of CO₂ (carbon dioxide). This can positively influence efforts to limit climate change. For these reasons numerous systems have been developed for reducing the damage of wildland fires. Many deep learning models, including CNNs and vision transformers, are used for detecting and localizing smoke and fires using ground and aerial images, as presented in Table 1. The main architecture of these models is described in three parts: the backbone, the neck, and the head. The backbone is used to extract both low-level and high-level features from the input data at different scales. It is usually a CNN. The neck enhances the semantic and spatial information at the different scales. It refines the features extracted by the backbone. These features are used by the head to predict the location of the bounding boxes and to classify the objects in them [16]. Many of the recent works aim to improve these parts of the architecture of the fire detection algorithm. Among them, Islam and Habib [28] made changes to the YOLOv5 architecture in order to improve the detection of small fires. They modified the neck part by adding a focus module, which enhances the feature propagation of small fires through the network. For tests, they used a private dataset, with 2462 images collected from open sources in Roboflow and Github. The modified YOLOv5n and YOLOv5x scaled models achieved mAP scores of 83.2% and 90.5%, respectively.

Wang et al. [29] used an object detector based on the vision transformer architecture to detect smaller flame and smoke areas. They used a private dataset and a public dataset, called the fire smoke dataset. The private data include 5900 images, mostly of indoor fires. Its main challenge is the presence of objects like clouds, occlusions, and reflections. The fire smoke dataset comprises 23,730 images under different lighting (indoor or outdoor) and weather conditions. To prevent overfitting and to make the model generalized enough, they applied data augmentation techniques, such as scaling, horizontal flipping, padding, cropping, and normalization. The DFFT (Decoder-Free Fully Transformer) architecture, which relies solely on the self-attention mechanism of transformers as an alternative to the

use of CNNs, is employed as it is effective in detecting targets of different sizes. The neck is comprised of encoder modules for the aggregation of the features extracted by the backbone at different scales. The detection head also includes encoder modules to divide the features into ones used for classification and ones employed for regression. Then, these generated features are adopted to predict the bounding boxes. The DFFT model achieved mAP scores of 87.40% and 81.12% using the private dataset and fire smoke dataset, respectively. It showed its efficiency in identifying and localizing fire and smoke with different sizes and outperformed state-of-the-art models.

Huang et al. [30] used the deformable DETR [26], which improves the slow convergence and the limited spatial resolution limitations of the original DETR [31], for detecting forest smoke. The deformable DETR employs the deformable attention module, which focuses on a small set of sampling points around a chosen reference point instead of paying attention to all the possible locations. The backbone of this model is improved by adding two modules, MCCL (Multi-scale Context Contrast Local Feature module) and DPPM (Dense Pyramid Pooling module). The MCCL module improves the detection of early smoke and the DPPM enhances the model's ability to differentiate between smoke and smoke-like objects. The authors also introduced a method to refine the predicted bounding boxes and to enhance the precision. Using 10,250 forest smoke images, the deformable DETR model achieved an mAP of 88.4%, surpassing the baseline deformable DETR by 2.6%.

Bahhar et al. [32] proposed a staged system, combining a CNN for classification and YOLO models. First, the CNN classifies the input image as normal or abnormal. When an abnormal image is identified, YOLOv5s and YOLOv5l models are employed for detecting and localizing smoke and flame, respectively. A dataset composed of 937 annotated images is utilized to train and evaluate these models, resulting in an mAP score of 76%. Chen et al. [33] made improvements to the architecture of YOLOv7 for detecting smoke. First, they integrated the RepVGG network into the backbone of the YOLOv7 model. The RepVGG network adopts multi-branch structures during training and converts them to a planar structure during the inference task with a lossless compression method. This allows a better detection performance during training while improving the speed for inference. They also added the ECA (Efficient Channel Attention) attention module to the backbone, allowing the proposed network to focus on the relevant objects and reduce the interference of the background. They used a dataset with 9005 images, of which 6605 of them contain smoke and 2400 non-smoke images. The improved model obtained an mAP of 93.7%, outperforming the original YOLOv7 by 1.5%.

Li et al. [34] proposed a modified YOLOv5s model for forest fire and smoke detection. They employed a Coordinate Attention (CA) module, which allows the model to focus on the relevant areas of the input image. They also added a Receptive Field Block (RFB) module to the backbone to generate feature maps at different scales, and a Bidirectional Feature Pyramid Network (Bi-FPN) module to the neck to reduce the computational cost. The improved YOLOv5s model reached an mAP of 58.8% using 450 forest fire and smoke images. It also outperformed the original YOLOv5s by 4.5%.

Sun et al. [35] introduced a deep learning model, namely ForestFireDetector, to identify and localize smoke. ForestFireDetector is a modified version of YOLOv8 adding the Space-to-Depth convolution (SPD-Conv) modules into the backbone to downsample feature maps to smaller scales by reducing the spatial dimension while increasing the channel dimension and avoiding the loss of fine-grained information. This allows for a better detection performance for both the detection of small zones of smoke and images with low resolution. The neck is modified with the employment of Ghost Shuffle Convolution (GSConv) modules, which are faster and lighter than the standard convolution modules, with a slight accuracy loss. Using a dataset composed of 3966 images of forest fire smoke, the ForestFireDetector model achieved an mAP of 90.2%, outperforming the baseline YOLOv8n by 3.3%.

Chen et al. [36] proposed a modified architecture for the YOLOv7 model, namely LMDFS, for detecting forest smoke on UAV (Unmanned Aerial Vehicle) images. To make

the LMDFS model lightweight, they employed the GSConv module in the neck, which is faster and lighter than the standard convolution modules while slightly losing accuracy. The Hardswish activation function is also applied instead of SiLU function thanks to its faster computation. To improve the accuracy of the proposed model, they added the CA module to the backbone and the CARAFE (Content-Aware Reassembly of Features) module at the neck for improving the upsampling stage in information fusion. The used dataset includes 5311 images of smoke from the point of view of UAVs. The LMDFS model achieved an mAP of 80.2%, surpassing the baseline YOLOv7 by 5.9%.

Yang et al. [37] modified the YOLOv3 model for the detection of smoke and wildfires. To improve the detection of small fires and smoke, an additional large-scale feature map is used for the neck. They added residual blocks to the YOLOv3 backbone to enhance the training of the model. They also employed K-means clustering to improve the setup of the anchor boxes and to reduce the complexity of the predictions. The modified model obtained an mAP of 95.0%, overcoming the baseline YOLOv3 by 14.1% using a large dataset consisting of 30,411 images of smoke and fire.

Sun et al. [38] proposed a fire detection method, called AERNet, based on YOLOv4 as its baseline. They employed Ghost modules in the backbone to generate feature maps using cheap linear transformations. This makes the model lightweight. They added SE (Squeeze-and-Excitation) attention modules between the Ghost modules to enhance the extraction of features. For the neck, they employed the PANet (Path Aggregation Network) module, which enhances the information fusion of the feature maps of different scales, combined with a Convolutional Block Attention Module (CBAM), which allows the network to focus on the relevant regions of the image. They introduced the SF-dataset, composed of 9246 fire and smoke images. The FIRESENSE dataset, which consists of 49 videos of smoke and fire, is also employed in the experiments. By training on the SF-Dataset and testing on the FIRESENSE dataset, the model achieved an mAP of 69.42%, better than the baseline YOLOv4 by 1.95%.

Sun and Feng [39] modified the YOLOv3 model for detecting fire and smoke. They added a CBAM attention module to the neck to improve the accuracy of the model. They modified the detection head to make it anchor-free, as it is more efficient for the detection of fire and smoke thanks to its flexibility for variable shapes and sizes. They also developed a lightweight version of the model. These methods are trained with a dataset composed of 10,029 images of wildfire and smoke. The modified YOLOv3 and its lightweight version achieved an mAP of 73.3% and 70.1%, respectively. Both surpassed the basic YOLOv3 model by 4.4% and 1.2%, respectively.

Jin et al. [40] modified the YOLOv7 model to detect fire and smoke. To the neck, they added a PANet module with radial connections to connect the weights of the module to different layers of the backbone. This enhances the fusion of information from feature maps at different scales and prevents the loss of relevant information. They also added a permutation self-attention module to the neck in order to focus the model efficiently on the important regions of the image. They used a dataset composed of 14,904 images of smoke, fire, and challenges such as clouds and illumination. The modified YOLOv7 achieved an mAP of 87.9%. While being surpassed by YOLOv7x with an mAP of 1.4%, the modified model is lighter, processing 4.9 more frames per second and costing less with 45.3 GFLOPS (Billions of Floating Point Operations per Second).

Kim and Muminov [41] modified the YOLOv7 model to detect smoke using aerial images. They added a CBAM to allow the network to focus on the important areas of the image. The spatial pyramid pooling fast (SPPF) module, on top of the original backbone, is modified to include more connections and feature reuse, making better use of the global information generated by the pooling of information at different scales. They integrated the Bi-FPN component at the neck to reduce computational cost. They also added decoupled detection heads, in which classification and bounding box tasks are performed as separate tasks. This enhances the detection accuracy. They used a dataset of 6500 images collected

from UAVs, of which 3500 contain smoke and 3000 do not. The improved YOLOv7 model obtained an mAP of 86.4%, surpassing the baseline YOLOv7 model by 3.9%.

Table 1. Deep learning models for smoke and fire detection.

Ref.	Methodology	Dataset	Object Detected	mAP (%)
[28]	Modified YOLOv5n Modified YOLOv5x	Private dataset: 2462 images	Fire Smoke	83.20 90.50
[29]	DFFT	Private dataset: 5900 fire images Fire smoke dataset: 23,730 images	Fire Smoke	87.40 81.12
[32]	YOLOv5s and YOLOv5l	Private dataset: 937 images	Fire Smoke	76.00
[33]	Modified YOLOv7	Private dataset: 9005 images (6605 smoke images and 2400 non-smoke images)	Smoke	93.70
[30]	Deformable DETR	Private dataset: 10,250 forest smoke images	Smoke	88.40
[34]	Improved YOLOv5s	Private dataset: 450 forest fire smoke images	Fire Smoke	58.80
[35]	ForestFireDetector	Private dataset: 3966 images of forest fires	Smoke	90.20
[36]	LMDFS	Private dataset: 5311 aerial smoke images	Smoke	80.20
[37]	Modified YOLOv3	Private dataset: 30,411 images	Fire Smoke	95.00
[38]	AERNet	SF-dataset: 9246 fire and smoke images; FIRESENSE dataset: 49 videos of smoke and fire	Fire Smoke	69.42
[39]	Modified YOLOv3	Private dataset: 10,029 images of smoke and fire	Fire Smoke	73.30
[40]	Modified YOLOv7	Private dataset: 14,904 images	Fire Smoke	87.90
[41]	Modified YOLOv7	Private dataset: 6500 aerial images	Smoke	86.40
[42]	Pruned YOLOv4	D-Fire: 21,527 images of smoke and fire	Fire Smoke	73.98
[43]	YOLOv5	D-Fire: 21,527 images of smoke and fire	Fire Smoke	79.46
[44]	Optimized YOLOv5	Public dataset: 6000 aerial images	Smoke	73.60

Venâncio et al. [42] tested different pruning techniques to make the YOLOv4 model lighter. Pruning consists of removing the convolutional filters of lesser importance to reduce the computational cost while maintaining the original accuracy. The pruned YOLOv4 is trained on the D-Fire dataset, which contains 21,527 images of fire, smoke, and fire-/smoke-like objects. The pruned YOLOv4 model reached an mAP of 73.98%, the same as the baseline YOLOv4, while reducing the computational cost by 83.60%. Venâncio et al. [43] also used YOLOv4 and YOLOv5 models to detect and localize smoke and fires in order to reduce false alarms. Two methods are employed: temporal persistence and area variation analysis. The first method consists of triggering the alarm only after the detection of smoke or fire objects in multiple consecutive frames. This is because elements that cause false alarms typically last a few frames. The area variation analysis consists of verifying the expansion of fire and smoke objects by calculating a persistence coefficient using the centroids of the objects detected in consecutive frames and activating the alarm when this coefficient exceeds a predefined threshold value. The principle of this method is that objects which cause false detections are usually static or do not grow in a short time. The Tiny YOLOv4, YOLOv4, YOLOv5s, and YOLOv5l models are trained and tested using the D-Fire dataset, achieving mAPs of 63.34%, 76.56%, 78.30%, and 79.46%, respectively.

Mukhiddinov et al. [44] developed an optimized YOLOv5 model for detecting forest smoke using UAV images. They integrated a K-means++ method with anchor box clustering to minimize the error of classification, a spatial pyramid pooling fast-plus layer into its backbone to focus on small areas of forest smoke, a Bi-FPN method to generate faster multi-scale fusion of features, and transfer learning and pruning techniques to refine performance and accelerate the optimized YOLO model. They collected 6000 aerial images from open-access fire datasets for training and testing the proposed method. They also applied various data augmentation techniques (rotation and horizontal flip), achieving an AP of 73.60%, better than baseline object detection methods, such as SSD, RefineDet, EfficientDet, DeepSmoke, YOLOv2, YOLOv3, YOLOv4, DeNet, Faster R-CNN, and Mask R-CNN.

As presented in Table 1, the detection of smoke and fires is more accurate as a result of deep learning methods. Nonetheless, there are still numerous challenging limitations, including the visual similarity between smoke, fog, and clouds, the varying features of smoke and fires in terms of colors, flow pattern, size, and shape, the detection of small smoke and fire zones, and the visual resemblance among fire, lighting, and sun glare. On the other hand, these advanced systems can require high consumption of energy, notably when running a large-scale wildfire detection system, which impact the environment as it increases carbon emissions.

3. Materials and Methods

This section consists of a description of the recent YOLO models and the D-Fire and WSDY datasets. First, the YOLO models used, namely YOLOv5, YOLOv5u, YOLOv7, and YOLOv8, are introduced before presenting challenging limitations in detecting wildland fires. Secondly, the D-Fire and WSDY datasets are described. Finally, the evaluation metrics used to compare the performance of each model are presented.

3.1. Proposed Models

YOLO models consist of the backbone, neck, and head parts. The backbone, responsible for feature extraction is composed of a CNN. It is used to extract features at multiple scales, generating feature maps with different sizes. The low-level feature maps generated are bigger and have high spatial information while having low semantic information. The high-level feature maps are smaller and have high semantic information with low spatial information. The neck refines the features extracted by the backbone by fusing information. It combines the information of the high-level and low-level feature maps. In the YOLO series, the neck is usually composed of an FPN (Feature Pyramid Network) module or an improved version of it, such as the PANet (Path Aggregation Network) and the Bi-FPN (Bidirectional Feature Pyramid Network) modules. The FPN module downsamples the low-level feature maps to fuse them with the high-level feature maps, improving their spatial information. The improved modules, PANet and Bi-FPN, add a stage in which the smaller feature maps are upsampled and fused with the bigger feature maps, enhancing their semantic information. The head uses the refined features to predict the bounding boxes and their classes. It consists of convolutional or fully connected layers. The head can be coupled, in which it consists of a single structure to predict the bounding boxes and perform the classification, or decoupled, in which the head is separated into a branch for localization and a branch for classification. The decoupled head structure shows better accuracy and training convergence [16].

Another important aspect of YOLO models is the employment of anchor boxes, which are boxes with predetermined shapes used to make predictions for the bounding boxes [16]. YOLO models can be divided among those using anchor boxes and those not incorporating them.

3.1.1. YOLOv5

YOLOv5 [45] was developed by Ultralytics. It uses anchor boxes for predictions. It adopts an algorithm, called AutoAnchor, which adjusts the anchor box prototypes for the

dataset. The backbone begins with a STEM layer, which is a convolutional layer with a large window size, to reduce computational cost. On top of the backbone, there is a spatial pyramid pooling fast (SPPF) module, which pools features of different scales into a single feature map, reflecting global information. The neck incorporates a PANet module for information fusion. It has a coupled head, in which the predictions of the locations of bounding boxes and the classification of their classes are computed in the same branch. It has five scaled versions, whose convolutional layers vary in width and depth. The YOLOv5x model reached an mAP of 50.7% on the MS COCO dataset [16].

3.1.2. YOLOv7

YOLOv7 [46] uses E-ELAN (Extended Efficient Layer Aggregation Network) blocks in its backbone and its neck to enhance the model's learning and convergence. It also employs reparameterization and batch normalization. It has a coupled head, making predictions for bounding boxes and classification in one single branch. YOLOv7 achieved an mAP of 52.8% using the MS COCO dataset [16].

3.1.3. YOLOv8

YOLOv8 [47] was developed by Ultralytics like YOLOv5. It has a backbone similar to YOLOv5, with improvements in the convolutional modules for speed and accuracy. It has a decoupled head, in which the classification and bounding box localization tasks are performed in different branches and does not use anchor boxes for the predictions. It uses the CIoU (complete Intersection over Union) loss and DFL (Distribution Focal Loss) loss functions for the localization task and the BCE (Binary Cross-Entropy) loss for the classification task. Similar to YOLOv5, YOLOv8 has five scaled versions. The extra-large scaled version, YOLOv8x, obtained an mAP of 53.9% on the MS COCO dataset [16].

3.1.4. YOLOv5u

YOLOv5u [48] is a new version of YOLOv5, also developed by Ultralytics. It has the same backbone and neck as YOLOv5 but features the same decoupled and anchor-free detection head as YOLOv8. The extra-large scaled version, YOLOv5xu, reached an mAP of 53.2% on the MS COCO dataset [48].

3.2. Wildland Fire Challenges

YOLO models showed a high and robust result in object detection, demonstrating a high trade-off between accuracy and speed and making them suitable for many real-time applications, including wildland fire detection and monitoring. However, many challenging limitations remain in detecting wildfires and smoke [7,49,50]. Firstly, the limited available wildfire data and the small size of existing fire data are important challenges. This can affect the performance of YOLO models in forest fire detection as YOLO models, like deep learning models, require a large amount of labeled fire data to efficiently train on different wildland fire scenarios. Furthermore, as wildfires can occur in many different environments, under varying weather conditions, and at different times of day, the lack of diverse data can prevent the generalizability of a reliable model when learning different scenarios. Moreover, wildfires and smoke are more clearly visible and detectable in open areas from a longer distance. However, varying canopy density, lighting conditions, and the presence of numerous natural items in dense forests all contribute to background complexity. They can mask early fire ignitions until they reach an advanced phase, complicating early monitoring and detection. This reduces the reliability of fire detection models. Additionally, extreme weather conditions, such as heavy fog and rain, considerably reduce wildfire visibility, thus also impacting the efficiency in detecting wildland fires. Finally, wildfires and smoke vary in terms of intensity, shape, and size, from small to medium to large. Small fire zones are the first stages of a fire, often with low visibility. Detecting them as soon as they appear is therefore a crucial limitation, enabling rapid intervention to prevent fire spread.

3.3. Dataset

To evaluate the performance of YOLO models, we used two available datasets, D-Fire [42,51] and WSDY (WildFire Smoke Dataset YOLO) [52].

1. The D-Fire dataset was introduced by Venâncio et al. [42,51] for the detection of smoke and fires. It includes aerial and ground images. It consists of a total of 21,527 images, of which 1164 images contain only fires, 5867 are smoke images, and 4658 are smoke and fire images, while the remaining 9838 images are non-fire and non-smoke images. It presents smoke and fires with different shapes, textures, intensities, sizes, and colors. The D-Fire dataset also includes images with challenging conditions, including scenarios with insects obstructing the camera, raindrops scattered, lighting, fog, clouds, and sun glare. These variations in the environmental factors provide diversity to the dataset and enhance its representation of the real challenges faced when detecting smoke and fires. Figure 1 presents some examples of the D-Fire dataset.



Figure 1. D-Fire dataset examples, from top to bottom: smoke images, fire images, fire/smoke images with challenging scenarios such as the presence of clouds, fog, and sun glare.

2. The WSDY dataset [52] is a publicly available dataset developed by Hemateja for detecting and localizing wildfire smoke. It contains 737 smoke images, divided into training, validation, and test sets with their corresponding YOLO annotations. It depicts numerous wildland fire smoke scenarios with challenging situations such as the presence of clouds, as depicted in Figure 2.



Figure 2. WSDY dataset examples.

3.4. Evaluation Metrics

One of the most common metrics for the evaluation of object detection models is the Average Precision (AP). It is based on two metrics: recall (R) and precision (P). Precision is the ratio of the true positives (TP) and the total number of predicted positives, which are true positives and false positives (FP), as given in Equation (1).

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

Recall is the ratio of the true positives and the ground-truth positives, including the false negatives (FN), as shown in Equation (2). It measures how many positives the model can retrieve.

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

By varying the confidence threshold, different recall and precision values are obtained. By increasing it, the precision increases, but the recall decreases; in contrast, by decreasing the threshold, the precision decreases and the recall increases. The Average Precision (AP) metric is obtained by computing the area under the precision–recall curve obtained by plotting the recall and precision values at different confidence threshold points. Mean Average Precision (mAP) consists of computing the AP for each class in the dataset. For object detection, true positives are defined by Intersection over Union (IoU), which is the ratio between the area of intersection of the predicted bounding box and the ground-truth box and the area of their union. The confidence, in this case, is the threshold of IoU values. In this paper, the mAP with the threshold of 0.50 is used as the main metric to compare the different models' performances. The AP for the individual smoke and fire classes is also measured.

A metric commonly used to measure the computational cost of the models is FLOPS, which is the number of floating-point operations per second performed by the model. Models with lower FLOPS have lower computational costs, while models with higher FLOPS have higher costs. As deep learning models perform billions of floating-point operations, it is common to measure performance using the GFLOPS metric, which represents billions of floating-point operations per second. The inference time presents the time taken by the proposed YOLO models to detect and localize smoke and fires. Models must make their inferences fast enough to reduce delays in the management of wildfires.

4. Results and Discussion

In this section, there is a description of the implementation of the experiment results as well as an analysis of these results.

4.1. Implementation Details

The experiments are performed on a machine with 32 GB of RAM and an NVIDIA Tesla V100 SXM2 16 GB GPU.

The D-Fire dataset used in the experiment is split into training, testing, and validation sets. First, the testing data contained 20% of the images in the dataset (4306 images). The remaining images are split into a training set, consisting of 15,498 images, and a validation set, comprising 1723 images. Moreover, we tested the performance of the YOLO models using the WSDY dataset, which are split into three sets: a training set (517 images), a validation set (110 images), and a test set (110 images).

The YOLOv8 and YOLOv5u models are developed using Python with the Ultralytics package. The different scaled models trained are the nano, small, medium, large, and extra-large models, denoted by the suffixes “n”, “s”, “m”, “l”, and “x”, respectively. In addition, the YOLOv5 (YOLOv5n, YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x) and YOLOv7 (YOLOv7 and YOLOv7x) models are also implemented using Python with the PyTorch package [53].

During training, the following hyperparameters are employed: 100 epochs, a batch size of 16, an input image size of 640×640 pixels, a momentum of 0.937, a weight decay of 0.0005, and a learning rate of 0.01. We also used YOLO models pretrained on the COCO dataset. Additionally, numerous data augmentation techniques are applied, such as rotation, translation, scale, shear, mixup, and flip, to diversify training data.

4.2. Result Analysis

The evaluation of the YOLO models included several crucial aspects. Firstly, their performance is evaluated in terms of AP, mAP, and GFLOPS compared to state-of-the-art methods using the D-Fire and WSDY testing sets, as shown in Tables 2 and 3. Secondly, their predicted visual detection using input images is presented in detail. We also calculate the training time and the power consumption [54], which represents the multiplication of the power consumption of the GPU used (GPUPower), 300 Watt for NVIDIA Tesla V100, its training time (TT), and PUE (Power Usage Effectiveness) of 1.1, as shown in Equation (3), for each YOLO model.

$$PowerConsumption = GPUPower * TT * PUE \quad (3)$$

The obtained results of the proposed YOLO models and the state-of-the-art models (YOLOv4, YOLOv5s, YOLOv5l, Tiny YOLOv4, and Faster R-CNN) [43,55] using the testing set are presented in Table 2. Based on the mAP metric, it is noticeable that all of the models performed well in the experiment. The minimum mAP observed is 75.90%, obtained by the YOLOv5n model. The YOLOv7x model showed a greater performance than the others, achieving an mAP of 80.40%. It outperformed YOLOv5l, YOLOv8x, and YOLOv5lu by 0.9%, 0.7%, and 0.7%, respectively. It also provided an improvement of 0.94%, 2.10%, 3.84%, 17.06%, and 44.45% compared to the published models YOLOv5l, YOLOv5s, YOLOv4, Tiny YOLOv4, and Faster R-CNN, respectively [43].

The detection of smoke objects is a challenging task due to their varying shape, size, and intensity as well as the detection of objects similar to smoke, such as clouds and fog. All of the YOLO models showed a good performance in detecting and localizing smoke based on the AP metric. For instance, YOLOv5l, YOLOv7x, YOLOv8x, YOLOv5lu, and YOLOv5xu achieved an AP of 85.70%, 85.50%, 85.60%, 86.00%, and 86.00%, respectively. YOLOv5n reached the lowest AP value of 80.70%. It still performed slightly worse than the baseline model, YOLOv5l [43]. However, It achieved a faster speed with a GFLOPS of 4.10 compared to the proposed YOLO models as well as the state-of-the-art methods.

Table 2. Comparative analysis of YOLO models on D-Fire dataset.

Model	mAP (%)	AP-Smoke (%)	AP-Fire (%)	GFLOPS	Training Time (hour)	Power Consumption (Wh)
YOLOv5n	75.90	80.70	71.00	4.10	2.100	693.00
YOLOv5s	78.30	82.90	73.60	15.80	2.573	849.09
YOLOv5m	78.70	84.90	72.40	47.90	4.088	1349.04
YOLOv5l	79.50	85.70	73.40	107.70	6.610	2181.3
YOLOv5x	78.90	85.30	72.50	203.80	9.872	3257.76
YOLOv7	80.20	85.10	75.30	103.20	9.031	2980.23
YOLOv7x	80.40	85.50	75.40	188.00	11.655	3846.15
YOLOv8n	77.80	84.00	71.50	8.10	2.002	660.66
YOLOv8s	78.70	84.90	72.50	28.40	2.767	913.11
YOLOv8m	79.00	84.90	73.10	78.70	4.726	1559.58
YOLOv8l	79.10	85.00	73.20	164.80	6.806	2245.98
YOLOv8x	79.70	85.60	73.90	257.40	10.795	3562.35
YOLOv5nu	76.70	83.60	69.90	7.10	2.210	729.30
YOLOv5su	78.50	84.20	72.90	23.80	2.825	932.25
YOLOv5mu	79.40	85.90	73.00	64.00	4.535	1496.55
YOLOv5lu	79.70	86.00	73.40	134.70	6.415	2116.95
YOLOv5xu	79.50	86.00	72.90	246.00	11.032	3640.56
Faster R-CNN	35.95	31.90	40.00	406.00	2.420	798.60
Tiny YOLOv4 [43]	63.34	62.20	64.48	16.07	—	—
YOLOv4 [43]	76.56	83.48	69.94	59.57	—	—
YOLOv5s [43]	78.30	84.29	72.78	15.80	—	—
YOLOv5l [43]	79.46	86.38	72.84	107.80	—	—

The YOLO models faced challenges in detecting fire objects due to the smaller size of the fire areas in the images of the training dataset D-Fire compared to smoke objects. In addition, numerous challenges related to the detection of fire, including the variability and complexity in the backgrounds and the different sizes, shapes, and intensities of flames, affect the detection of fire objects. Notably, the YOLOv7x model showed a slightly superior performance in detecting and localizing fire objects compared to YOLOv5, YOLOv8, and YOLOv5u, achieving an AP of 75.40%. It also outperformed the baseline models YOLOv5l, YOLOv5s, YOLOv4, and Tiny YOLOv4 by 2.56%, 2.62%, 5.46%, and 10.92%, respectively.

Lightweight models are crucial thanks to their ease of use and deployment on mobile devices with limited computational power and constrained resources, such as UAVs or surveillance stations in forests. They also offer a faster inference speed, compensating for their lower accuracy compared to larger models. For instance, the lightweight models YOLOv5n, YOLOv5s, YOLOv8n, YOLOv8s, YOLOv5nu, and YOLOv5su showed a rapid processing speed (GFLOPS) of 4.10, 15.80, 8.10, 28.40, 7.10, and 23.80, respectively, surpassing state-of-the-art methods such as Tiny YOLOv4, YOLOv4, YOLOv5s, and Faster R-CNN. This allows for real-time detection of smoke and fires. These models also achieved interesting mAP values of 75.90%, 78.30%, 77.80%, 78.70%, 76.70%, and 78.50%, respectively, using testing data. However, their performance is slightly inferior compared to medium, large, and extra-large YOLO models. On the other hand, the large and extra-large YOLO models showed a higher cost compared to the lightweight YOLO models. For example, YOLOv7x demonstrated superior detection performance with an mAP of 80.40% but with GFLOPS exceeding YOLOv5n by 45 times, YOLOv8n by 23 times, and YOLOv5nu by 26 times.

On the other hand, the training time increases as the complexity of the YOLO model increases. For instance, the lightweight YOLO models YOLOv5n, YOLOv5s, YOLOv8n, and YOLOv8s achieved a rapid training time between 2.1 and 2.825 h. YOLOv7x, as a large and complex model, had the longest training time of 11.655 h. YOLOv5xu also showed a high training time of 11.032 h, 5 times longer than YOLOv8n and YOLOv5n. Similar to training time, power consumption also depends on the complexity of the YOLO models, as more complex models require high computational power and consume more energy. Among

them, the large YOLO models YOLOv5x, YOLOv7x, YOLOv8x, and YOLOv5xu require a high power consumption of 3257.76, 3846.15, 3562.35, and 3640.56 Wh, respectively. Nano YOLO models (YOLOv5n, YOLOv5nu, and YOLOv8n), as lightweight models, consume less power, between 660.66 and 729.30 Wh, approximately 5 times less than YOLOv5xu.

As shown in Figure 3, the YOLO models performed well in detecting and localizing smoke and fire areas using both ground and aerial images. They successfully addressed challenging situations related to smoke and fire detection, including background complexity, the detection of small smoke and fire zones, and varying smoke and fire characteristics. These models identified smoke and fire objects with high confidence scores in input images. For instance, the YOLOv7x, YOLOv5l, YOLOv8x, and YOLOv5lu models accurately identified and localized smoke in input images with complex backgrounds, achieving high confidence scores of 0.79, 0.76, 0.75, and 0.68, respectively (see Figure 3a). Additionally, these models demonstrated their potential in detecting small fire areas in input images with complex backgrounds and varying environmental conditions, as illustrated in Figure 3b. They also showed their ability to address challenging limitations, such as the visual similarity between smoke and clouds (see Figure 3c). YOLOv7x also identified small areas of smoke, notably non-annotated areas, better than the manual annotation and the other YOLO models, as shown in Figure 3b. However, YOLOv5l falsely detected and localized the background as smoke, as depicted in Figure 3b.

To confirm the reliability of the YOLO models, we performed a comprehensive benchmark of these models using the WSDY dataset, which includes only smoke images, as highlighted in Table 3. Testing results showed a significant improvement in the performance of these models in detecting smoke compared with the previous evaluations presented in Table 2. This improvement can be attributed to two main factors. First, the number of WSDY test images (110 images) employed in our tests is less than that used in the previous benchmarks (4306 images). Second, our analysis focused exclusively on the reliability of YOLO models for detecting only smoke, not both smoke and fire, as presented in Table 2. This allowed us to refine the abilities of the YOLO models, resulting in high performance with an mAP of 98.10% for the YOLOv8s model in identifying and localizing smoke plumes, which is crucial for accurate early fire ignition. YOLOv8n, YOLOv5n, YOLOv7x, YOLOv5nu, and YOLOv5mu achieved an mAP superior to 95% compared to other YOLO models and to Faster R-CNN, which had the worst performance with an mAP of 50.34%. Additionally, all YOLO models require a fast training time less than 1 hour using the WSDY dataset as a small learning database (517 images). Large YOLO models (including the suffixes x and l) require more training time, as well as power consumption, compared to lightweight YOLO models (including the suffixes n and s). As an example, YOLOv5xu had a training time 0.965 h and a power consumption of 318.45 Wh, 11 times, 9 times, 5 times, 4 times, and 3 times more than YOLOv5n, YOLOv5s, YOLOv8n, YOLOv8s, and both YOLOv5nu and YOLOv5su, respectively.

As depicted in Figure 4, the YOLO models demonstrated their potential in detecting smoke and addressing related challenging scenarios. YOLOv5nu, YOLOv7x, YOLOv8s, and YOLOv5n performed well in differentiating between smoke and complex backgrounds (see Figure 4c). They also correctly identified and localized small smoke areas, as shown in Figure 4b, and overcame the visual resemblance between smoke and clouds, as illustrated in Figure 4a.

In conclusion, the YOLO models performed well in detecting and localizing fire and smoke in both ground and aerial images. They demonstrated their ability to overcome challenges, including the detection of small smoke and fire areas, the prediction and localization of smoke and fire in images with complex backgrounds, and varying shapes, intensities, and sizes of fire and smoke objects. They also proved effective in differentiating fire and smoke from other elements with similar shapes, colors, and textures, such as differentiating clouds from smoke. Additionally, the YOLO models achieved fast inference speed, enabling real-time detection of smoke and fires. This demonstrates the effectiveness of these models when integrated with surveillance systems for wildland fires, allowing

rapid firefighting and response. This limits environmental damage and improves wildfire management strategies, reducing the spread of wildfires and the area of burnt forest and minimizing economic and human losses.



(a) Smoke example

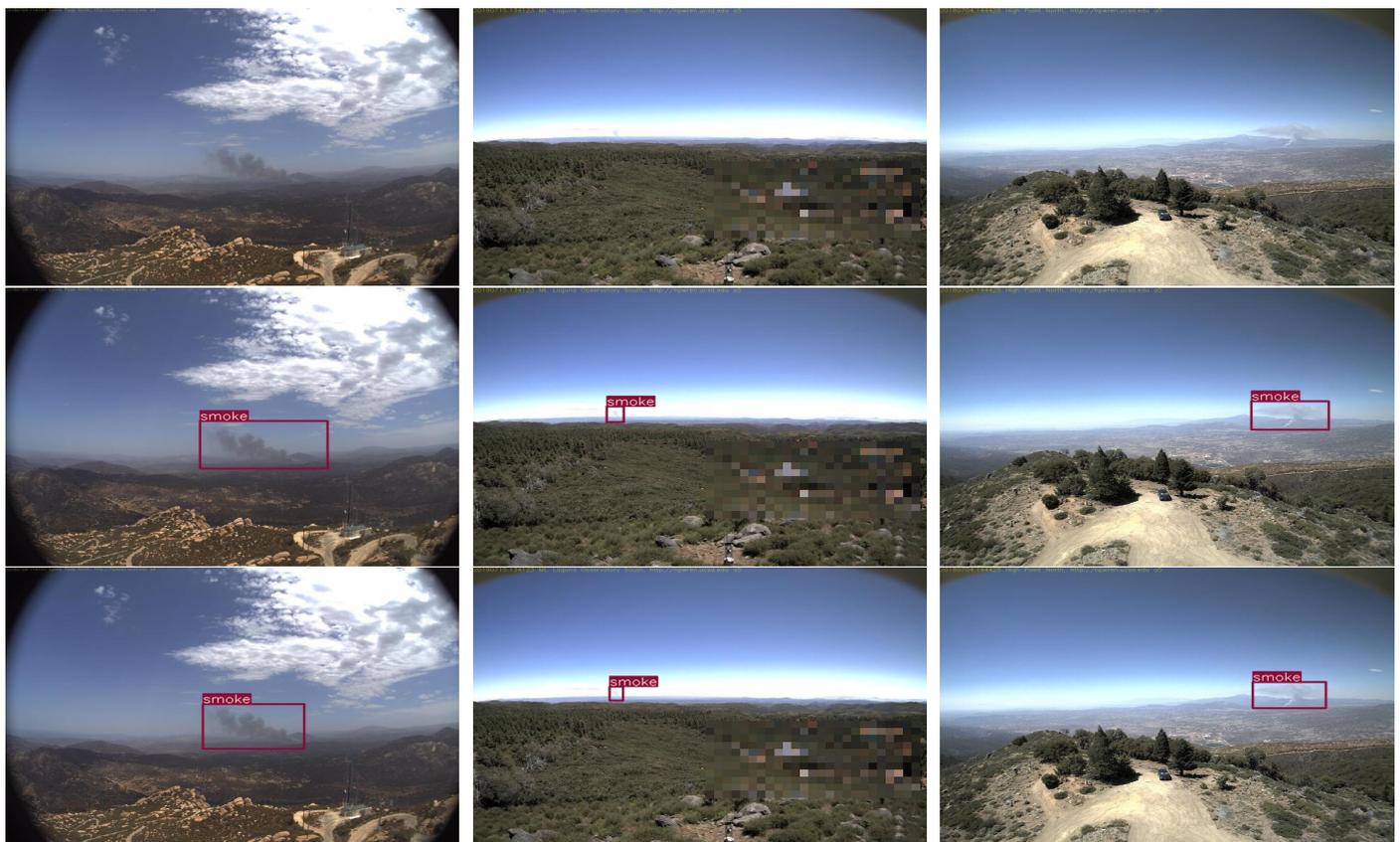
(b) Smoke and fire example

(c) Smoke with cloud example

Figure 3. YOLO model results using D-Fire dataset, from top to bottom: original images, ground-truth images, and predicted images by YOLOv5l, YOLOv7x, YOLOv8x, and YOLOv5lu, respectively.

Table 3. Comparative analysis of YOLO models on WSDY dataset.

Model	mAP (%)	GFLOPS	Training Time (Hour)	Power Consumption (Wh)
YOLOv5n	95.90	4.10	0.085	28.05
YOLOv5s	94.60	15.80	0.106	34.98
YOLOv5m	93.10	47.90	0.165	54.45
YOLOv5l	92.90	107.70	0.240	79.20
YOLOv5x	90.80	203.80	0.401	132.33
YOLOv7	92.60	103.20	0.355	117.15
YOLOv7x	96.00	188.00	0.353	116.49
YOLOv8n	97.30	8.10	0.179	59.07
YOLOv8s	98.10	28.40	0.241	79.53
YOLOv8m	93.50	78.70	0.401	132.33
YOLOv8l	92.40	164.80	0.573	189.09
YOLOv8x	94.10	257.40	0.886	292.38
YOLOv5nu	95.70	7.10	0.192	80.84
YOLOv5su	94.20	23.80	0.248	81.51
YOLOv5mu	95.10	64.00	0.275	90.75
YOLOv5lu	94.30	134.70	0.525	173.25
YOLOv5xu	93.80	246.00	0.965	318.45
Faster R-CNN	50.34	406.00	2.097	692.01

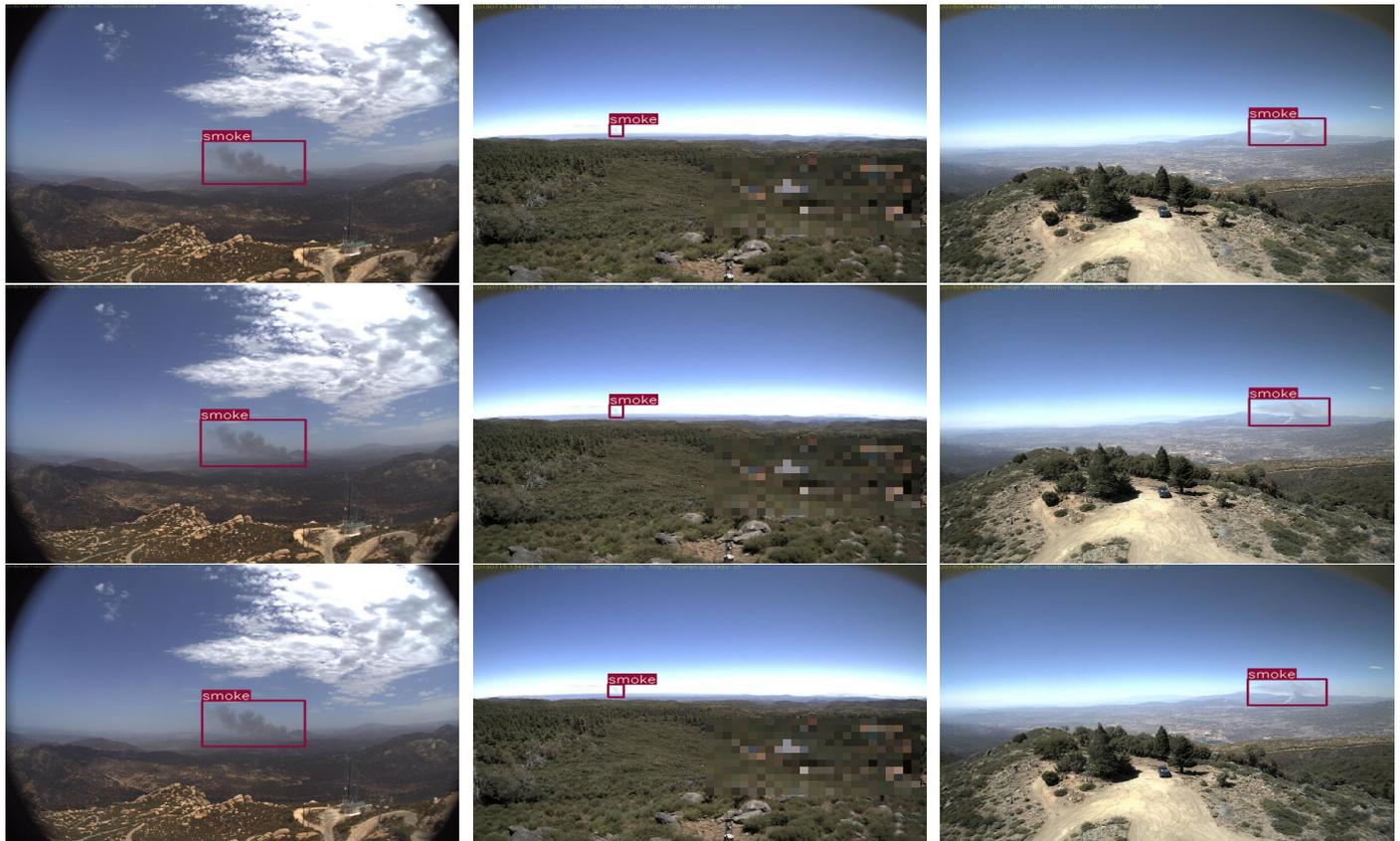


(a) Smoke with cloud example

(b) Small area of smoke

(c) Smoke example

Figure 4. Cont.



(d) Smoke with cloud example

(e) Small area of smoke

(f) Smoke example

Figure 4. YOLO model results using WSDY dataset, from top to bottom: original images, ground-truth images, and predicted images by YOLOv5nu, YOLOv7x, YOLOv8s, and YOLOv5n, respectively.

5. Conclusions

In this paper, recent YOLO models (YOLOv5, YOLOv7, YOLOv8, and YOLOv5u) are adopted in detecting and localizing smoke and wildfires in both ground and aerial images. Testing results are performed using the D-Fire and WSDY datasets. The D-Fire dataset includes smoke, fire, non-fire, non-smoke, and fire-/smoke-like object images. The WSDY dataset contains smoke images and smoke-like object images. Notably, YOLOv7x showed the best detection performance, achieving an mAP of 80.40% and outperforming the baseline models using the D-Fire dataset. YOLOv8s achieved a high performance with an mAP of 98.10% using the WSDY dataset. Additionally, the lightweight models YOLOv5n, YOLOv5s, YOLOv8n, YOLOv8s, YOLOv5nu, and YOLOv5su outperformed the baseline models in processing speed with 4.10, 15.80, 8.10, 28.40, 7.10, and 23.80 GFLOPS, respectively. They also required less training time and power consumption compared to larger models, YOLOv7x, YOLOv5x, YOLOv5l, YOLOv8l, YOLOv8x, YOLOv5lu, and YOLOvxu. The YOLO models also showed their potential in addressing challenging situations, including the detection of small fire and smoke areas; the detection of fire and smoke considering their varying shapes, sizes, colors, textures, and intensities; handling background complexity, such as diverse terrains, varying vegetation, and different weather conditions; and distinguishing smoke from similar objects, such as fog and clouds, and fire from similar objects, such as lighting and the glare of the sun.

As future work, we plan to optimize the YOLO models and deploy them on edge devices in order to improve wildland fire detection performance.

Author Contributions: Conceptualization, L.A.O.G., R.G. and M.A.A.; methodology, L.A.O.G., R.G. and M.A.A.; software, L.A.O.G.; validation, L.A.O.G., R.G. and M.A.A.; formal analysis, L.A.O.G., R.G. and M.A.A.; writing—original draft preparation, L.A.O.G.; writing—review and editing, R.G. and M.A.A.; funding acquisition, M.A.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research was enabled in part by support provided by the Natural Sciences and Engineering Research Council of Canada (NSERC), funding reference number RGPIN-2018-06233.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: This work uses a publicly available dataset; see references [42,51,52] for data availability. More details about this dataset are available in Section 3.3.

Acknowledgments: This research was enabled in part by support provided by Calcul Québec (www.calculquebec.ca) and the Digital Research Alliance of Canada (www.alliancecan.ca) (accessed on 30 March 2024).

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

Abbreviations

The following abbreviations are used in this manuscript:

YOLO	You Only Look Once
CNN	Convolutional Neural Network
MAP	Mean Average Precision
DFFT	Decoder-Free Fully Transformer
MCCL	Multi-scale Context Contrast Local Feature Module
DPPM	Dense Pyramid Pooling module
CA	Coordinate Attention
RFB	Receptive Field Block
CARAFE	Content-Aware Reassembly of Features
SE	Squeeze and Excitation
UAV	Unmanned Aerial Vehicle
GFLOPS	Billions of Floating-Point Operations per Second
FPN	Feature Pyramid Network
PANet	Path Aggregation Network
E-ELAN	Extended Efficient Layer Aggregation Network
ECA	Efficient Channel Attention
Bi-FPN	Bidirectional Feature Pyramid Network
SPD-Conv	Space-to-Depth Convolution
GSConv	Ghost Shuffle Convolution
CBAM	Convolutional Block Attention Module
SPPF	Spatial Pyramid Pooling Fast
IoU	Intersection over Union
AP	Average Precision
P	Precision
TP	True Positive
FP	False Positive
R	Recall
FN	False Negative
DFL	Distribution Focal Loss
CIoU	Complete Intersection over Union
BCE	Binary Cross-Entropy
WSDY	WildFire Smoke Dataset YOLO
PUE	Power Usage Effectiveness

References

1. European Commission. Wildfires in the Mediterranean. Available online: https://joint-research-centre.ec.europa.eu/jrc-news-and-updates/wildfires-mediterranean-monitoring-impact-helping-response-2023-07-28_en (accessed on 14 November 2023).
2. Government of Canada. Forest Fires. Available online: <https://natural-resources.canada.ca/our-natural-resources/forests/wildland-fires-insects-disturbances/forest-fires/13143> (accessed on 14 November 2023).
3. Government of Canada. Protecting Communities. Available online: <https://natural-resources.canada.ca/our-natural-resources/forests/wildland-fires-insects-disturbances/forest-fires/protecting-communities/13153> (accessed on 14 November 2023).
4. Government of Canada. *Social Aspects of Wildfire Management*; Natural Resources Canada: Ottawa, ON, Canada, 2020.
5. Alkhatib, A. A Review on Forest Fire Detection Techniques. *Int. J. Distrib. Sens. Netw.* **2014**, *10*, 597368. [\[CrossRef\]](#)
6. Geetha, S.; Abhishek, C.S.; Akshayanat, C.S. Machine Vision Based Fire Detection Techniques: A Survey. *Fire Technol.* **2021**, *57*, 591–623. [\[CrossRef\]](#)
7. Ghali, R.; Akhloufi, M.A. Deep Learning Approaches for Wildland Fires Remote Sensing: Classification, Detection, and Segmentation. *Remote Sens.* **2023**, *15*, 1821. [\[CrossRef\]](#)
8. Ghali, R.; Akhloufi, M.A. Wildfires Detection and Segmentation Using Deep CNNs and Vision Transformers. In Proceedings of the Pattern Recognition, Computer Vision, and Image Processing, ICPR 2022 International Workshops and Challenges, Montreal, QC, Canada, 21–25 August 2023; pp. 222–232.
9. Yuan, F. Video-based Smoke Detection with Histogram Sequence of LBP and LBPV Pyramids. *Fire Saf. J.* **2011**, *46*, 132. [\[CrossRef\]](#)
10. Long, C.; Zhao, J.; Han, S.; Xiong, L.; Yuan, Z.; Huang, J.; Gao, W. Transmission: A New Feature for Computer Vision Based Smoke Detection. In Proceedings of the Artificial Intelligence and Computational Intelligence, Sanya, China, 23–24 October 2010; pp. 389–396.
11. Ho, C.C. Machine Vision-based Real-time Early Flame and Smoke Detection. *Meas. Sci. Technol.* **2009**, *20*, 045502. [\[CrossRef\]](#)
12. Tian, H.; Li, W.; Wang, L.; Ogunbona, P. Smoke Detection in Video: An Image Separation Approach. *Int. J. Comput. Vis.* **2014**, *106*, 192. [\[CrossRef\]](#)
13. Calderara, S.; Piccinini, P.; Cucchiara, R. Vision-based Smoke Detection System Using Image Energy and Color Information. *Mach. Vis. Appl.* **2011**, *22*, 705. [\[CrossRef\]](#)
14. O'Mahony, N.; Campbell, S.; Carvalho, A.; Harapanahalli, S.; Hernandez, G.V.; Krpalkova, L.; Riordan, D.; Walsh, J. Deep Learning vs. Traditional Computer Vision. In Proceedings of the Advances in Computer Vision, Las Vegas, NV, USA, 2–3 May 2020; p. 128.
15. Ghali, R.; Akhloufi, M.A.; Souidene Mseddi, W.; Jmal, M. Wildfire Segmentation Using Deep-RegSeg Semantic Segmentation Architecture. In Proceedings of the 19th International Conference on Content-Based Multimedia Indexing, Graz, Austria, 14–16 September 2022; pp. 149–154.
16. Terven, J.; Córdova-Esparza, D.M.; Romero-González, J.A. A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS. *Mach. Learn. Knowl. Extr.* **2023**, *5*, 1680–1716. [\[CrossRef\]](#)
17. Lan, W.; Dang, J.; Wang, Y.; Wang, S. Pedestrian Detection Based on YOLO Network Model. In Proceedings of the IEEE International Conference on Mechatronics and Automation (ICMA), Changchun, China, 5–8 August 2018; pp. 1547–1551.
18. Rjoub, G.; Wahab, O.A.; Bentahar, J.; Bataineh, A.S. Improving Autonomous Vehicles Safety in Snow Weather Using Federated YOLO CNN Learning. In Proceedings of the Mobile Web and Intelligent Information Systems, Virtual, 23–25 August 2021; pp. 121–134.
19. Andrie Dazlee, N.M.A.; Abdul Khalil, S.; Abdul-Rahman, S.; Mutalib, S. Object Detection for Autonomous Vehicles with Sensor-based Technology Using YOLO. *Int. J. Intell. Syst. Appl. Eng.* **2022**, *10*, 129–134. [\[CrossRef\]](#)
20. Yang, W.; Jiachun, Z. Real-time Face Detection Based on YOLO. In Proceedings of the 1st IEEE International Conference on Knowledge Innovation and Invention (ICKII), Jeju Island, Republic of Korea, 23–27 July 2018; pp. 221–224.
21. Ashraf, A.H.; Imran, M.; Qahtani, A.M.; Alsufyani, A.; Almutiry, O.; Mahmood, A.; Attique, M.; Habib, M. Weapons Detection For Security and Video Surveillance Using CNN and YOLOv5s. *CMC-Comput. Mater. Contin.* **2022**, *70*, 2761–2775. [\[CrossRef\]](#)
22. Nie, Y.; Sommella, P.; O'Nils, M.; Liguori, C.; Lundgren, J. Automatic Detection of Melanoma with YOLO Deep Convolutional Neural Networks. In Proceedings of the E-Health and Bioengineering Conference (EHB), Iasi, Romania, 21–23 November 2019; pp. 1–4.
23. Salman, M.E.; Çakirsoy Çakar, G.; Azimjonov, J.; Kösem, M.; Cedimoğlu, I.H. Automated Prostate Cancer Grading and Diagnosis System Using Deep Learning-based YOLO Object Detection Algorithm. *Expert Syst. Appl.* **2022**, *201*, 117148. [\[CrossRef\]](#)
24. Yao, Z.; Jin, T.; Mao, B.; Lu, B.; Zhang, Y.; Li, S.; Chen, W. Construction and Multicenter Diagnostic Verification of Intelligent Recognition System for Endoscopic Images From Early Gastric Cancer Based on YOLOv3 Algorithm. *Front. Oncol.* **2022**, *12*, 815951. [\[CrossRef\]](#)
25. Ghali, R.; Akhloufi, M.A.; Jmal, M.; Mseddi, W.S.; Attia, R. Forest Fires Segmentation using Deep Convolutional Neural Networks. In Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC), Melbourne, Australia, 17–20 October 2021; pp. 2109–2114.
26. Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; Dai, J. Deformable DETR: Deformable Transformers for End-to-End Object Detection. *arXiv* **2021**, arXiv:2010.04159. [\[CrossRef\]](#)
27. Ghali, R.; Akhloufi, M.A. BoucaNet: A CNN-Transformer for Smoke Recognition on Remote Sensing Satellite Images. *Fire* **2023**, *6*, 455. [\[CrossRef\]](#)

28. Islam, A.; Habib, M.I. Fire Detection From Image and Video Using YOLOv5. *arXiv* **2023**, arXiv:2310.06351. [[CrossRef](#)]
29. Wang, X.; Li, M.; Gao, M.; Liu, Q.; Li, Z.; Kou, L. Early Smoke and Flame Detection Based on Transformer. *J. Saf. Sci. Resil.* **2023**, *4*, 294. [[CrossRef](#)]
30. Huang, J.; Zhou, J.; Yang, H.; Liu, Y.; Liu, H. A Small-Target Forest Fire Smoke Detection Model Based on Deformable Transformer for End-to-End Object Detection. *Forests* **2023**, *14*, 162. [[CrossRef](#)]
31. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-End Object Detection with Transformers. In Proceedings of the Computer Vision—ECCV, Glasgow, UK, 23–28 August 2020; pp. 213–229.
32. 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](#)]
33. Chen, X.; Xue, Y.; Hou, Q.; Fu, Y.; Zhu, Y. RepVGG-YOLOv7: A Modified YOLOv7 for Fire Smoke Detection. *Fire* **2023**, *6*, 383. [[CrossRef](#)]
34. Li, J.; Xu, R.; Liu, Y. An Improved Forest Fire and Smoke Detection Model Based on YOLOv5. *Forests* **2023**, *14*, 833. [[CrossRef](#)]
35. Sun, L.; Li, Y.; Hu, T. ForestFireDetector: Expanding Channel Depth for Fine-Grained Feature Learning in Forest Fire Smoke Detection. *Forests* **2023**, *14*, 2157. [[CrossRef](#)]
36. Chen, G.; Cheng, R.; Lin, X.; Jiao, W.; Bai, D.; Lin, H. LMDFS: A Lightweight Model for Detecting Forest Fire Smoke in UAV Images Based on YOLOv7. *Remote Sens.* **2023**, *15*, 3790. [[CrossRef](#)]
37. Yang, Y.; Pan, M.; Li, P.; Wang, X.; Tsai, Y.T. Development and Optimization of Image Fire Detection on Deep Learning Algorithms. *J. Therm. Anal. Calorim.* **2023**, *148*, 5089. [[CrossRef](#)]
38. Sun, B.; Wang, Y.; Wu, S. An Efficient Lightweight CNN Model for Real-time Fire Smoke Detection. *J. Real-Time Image Process.* **2023**, *20*, 74. [[CrossRef](#)]
39. Sun, Y.; Feng, J. Fire and Smoke Precise Detection Method Based on the Attention Mechanism and Anchor-Free Mechanism. *Complex Intell. Syst.* **2023**, *9*, 5185. [[CrossRef](#)]
40. Jin, C.; Zheng, A.; Wu, Z.; Tong, C. Real-Time Fire Smoke Detection Method Combining a Self-Attention Mechanism and Radial Multi-Scale Feature Connection. *Sensors* **2023**, *23*, 3358. [[CrossRef](#)]
41. Kim, S.Y.; Muminov, A. Forest Fire Smoke Detection Based on Deep Learning Approaches and Unmanned Aerial Vehicle Images. *Sensors* **2023**, *23*, 5702. [[CrossRef](#)]
42. Venâncio, P.V.A.B.; Lisboa, A.C.; Barbosa, A.V. An Automatic Fire Detection System Based on Deep Convolutional Neural Networks for Low-Power, Resource-Constrained Devices. *Neural Comput. Appl.* **2022**, *34*, 15349. [[CrossRef](#)]
43. Venâncio, P.V.A.B.; Campos, R.J.; Rezende, T.M.; Lisboa, A.C.; Barbosa, A.V. A Hybrid Method for Fire Detection Based on Spatial and Temporal Patterns. *Neural Comput. Appl.* **2023**, *35*, 9349. [[CrossRef](#)]
44. Mukhiddinov, M.; Abdusalomov, A.B.; Cho, J. A Wildfire Smoke Detection System Using Unmanned Aerial Vehicle Images Based on the Optimized YOLOv5. *Sensors* **2022**, *22*, 9384. [[CrossRef](#)]
45. Ultralytics. YOLOv5. Available online: <https://github.com/ultralytics/yolov5> (accessed on 27 November 2023).
46. Lin, H.; Liu, Z.; Cheang, C.; Fu, Y.; Guo, G.; Xue, X. 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 (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 7464–7475.
47. Ultralytics. YOLOv8. Available online: <https://github.com/ultralytics/ultralytics> (accessed on 28 November 2023).
48. Ultralytics. YOLOv5u. Available online: <https://docs.ultralytics.com/models/yolov5/> (accessed on 27 November 2023).
49. Taylor, S.W.; Woolford, D.G.; Dean, C.B.; Martell, D.L. Wildfire Prediction to Inform Fire Management: Statistical Science Challenges. *Stat. Sci.* **2013**, *28*, 586–615. [[CrossRef](#)]
50. Artés, T.; Oom, D.; De Rigo, D.; Durrant, T.H.; Maianti, P.; Libertà, G.; San-Miguel-Ayanz, J. A global wildfire dataset for the analysis of fire regimes and fire behaviour. *Sci. Data* **2019**, *6*, 296. [[CrossRef](#)] [[PubMed](#)]
51. Venâncio, P. D-Fire: An Image Dataset for Fire and Smoke Detection. Available online: <https://github.com/gaiasd/DFireDataset/tree/master> (accessed on 14 November 2023).
52. Hemateja, A.V.N.M. WildFire Smoke Dataset YOLO. Available online: <https://www.kaggle.com/datasets/ahemateja19bec1025/wildfiresmokedatasetyolo?resource=download> (accessed on 25 March 2024).
53. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; pp. 8026–8037.
54. Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. LLaMA: Open and Efficient Foundation Language Models. *arXiv* **2023**, arXiv:2302.13971. [[CrossRef](#)]
55. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [[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.