forests

MDPI

Article

FSNet: Enhancing Forest-Fire and Smoke Detection with an Advanced UAV-Based Network

Donghua Wu [1,*], Zhongmin Qian [2], Dongyang Wu [3] and Junling Wang [3]

[1] College of Continuing Education (Higher Vocational and Technical College), Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China
[2] Computer Basic Teaching and Experimental Center, Public Experimental Teaching Department, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China; qzm_mail@nuaa.edu.cn
[3] College of Information Science and Technology, Nanjing Forestry University, Nanjing 210037, China; wudongyang@njfu.edu.cn (D.W.); wangjunling@njfu.edu.cn (J.W.)
* Correspondence: wdh@nuaa.edu.cn

Abstract: Forest fires represent a significant menace to both the ecological equilibrium of forests and the safety of human life and property. Upon ignition, fires frequently generate billowing smoke. The prompt identification and management of fire sources and smoke can efficiently avert the occurrence of extensive forest fires, thereby safeguarding both forest resources and human well-being. Although drone patrols have emerged as a primary method for forest-fire prevention, the unique characteristics of forest-fire images captured from high altitudes present challenges. These include remote distances, small fire points, smoke targets with light hues, and complex, ever-changing background environments. Consequently, traditional target-detection networks frequently exhibit diminished accuracy when handling such images. In this study, we introduce a cutting-edge drone-based network designed for the detection of forest fires and smoke, named FSNet. To begin, FSNet employs the YOCO data-augmentation method to enhance image processing, thereby augmenting both local and overall diversity within forest-fire images. Next, building upon the transformer framework, we introduce the EBblock attention module. Within this module, we introduce the notion of "groups", maximizing the utilization of the interplay between patch tokens and groups to compute the attention map. This approach facilitates the extraction of correlations among patch tokens, between patch tokens and groups, and among groups. This approach enables the comprehensive feature extraction of fire points and smoke within the image, minimizing background interference. Across the four stages of the EBblock, we leverage a feature pyramid to integrate the outputs from each stage, thereby mitigating the loss of small target features. Simultaneously, we introduce a tailored loss function, denoted as L_{forest}, specifically designed for FSNet. This ensures the model's ability to learn effectively and produce high-quality prediction boxes. We assess the performance of the FSNet model across three publicly available forest-fire datasets, utilizing mAP, Recall, and FPS as evaluation metrics. The outcomes reveal that FSNet achieves remarkable results: on the Flame, Corsican, and D-Fire datasets, it attains mAP scores of 97.2%, 87.5%, and 94.3%, respectively, with Recall rates of 93.9%, 87.3%, and 90.8%, respectively, and FPS values of 91.2, 90.7, and 92.6, respectively. Furthermore, extensive comparative and ablation experiments validate the superior performance of the FSNet model.

Keywords: forest-fire prevention; drone surveillance; object detection; attention mechanism

Citation: Wu, D.; Qian, Z.; Wu, D.; Wang, J. FSNet: Enhancing Forest-Fire and Smoke Detection with an Advanced UAV-Based Network. *Forests* **2024**, *15*, 787. https://doi.org/10.3390/f15050787

Academic Editor: Giorgos Mallinis

Received: 7 March 2024
Revised: 28 April 2024
Accepted: 28 April 2024
Published: 30 April 2024

1. Introduction

Forests are one of the most important ecosystems on Earth, nurturing a multitude of plants and animals, maintaining ecological balance. Additionally, forests serve as natural shields to prevent soil erosion, and are crucial for absorbing carbon dioxide and releasing oxygen. Forest fires not only disrupt the ecological balance but also result in the loss of wildlife and the destruction of their habitats, potentially causing ecological disasters [1,2].

During forest fires, a large amount of carbon dioxide and harmful gases are released, negatively impacting air pollution. As the fire spreads, it can endanger human lives and properties, burning up a vast amount of forest resources, thereby affecting the supply of timber and wood products, and even ruining tourist attractions. When fires spread to cities and populated areas, they threaten the safety of nearby communities and residents, causing large-scale disasters.

To ensure the ecological balance of forests and protect human economic interests, detecting forest fires is crucial. Common detection methods include the use of satellites, weather stations, Unmanned Aerial Vehicles (UAVs), etc., for the early detection and rapid location of fires. This allows people to detect fires quickly and respond accurately and swiftly, extinguishing the sources of the fires, thereby protecting forest resources and the safety of human lives and properties. The cost of using satellites and weather stations to detect forest fires is high, and these methods are easily affected by factors such as clouds, terrain, and wind direction, which may lead to inaccurate detection results. Compared with satellites and weather stations, the cost of using UAVs to monitor forest fires is lower, making drone surveillance the mainstream method today. UAVs equipped with cameras fly over forests, taking real-time fire-monitoring photos and transmitting them to servers for analysis. There are also examples of hyperspectral and multispectral sensors being mounted on UAVs and applied to forest-fire prevention [3–5], which have the following problems compared to using RGB cameras: (1) hyperspectral and multispectral sensors cost much more than RGB cameras, making it easier for individuals and organizations to acquire and use RGB cameras without significant financial support; (2) RGB camera technology is very mature, is easier to integrate with the system, and can provide more comprehensive information; and (3) RGB cameras can provide real-time video streams and can quickly respond to emergency situations such as forest fires. Although hyperspectral and multispectral sensors can also provide real-time data, processing hyperspectral and multispectral data requires more complex processing and analysis, which requires more hardware equipment. When a forest fire occurs, target-detection technology can be used to pinpoint the location of the fire. Several critical challenges remain to be addressed in the task of utilizing Unmanned Aerial Vehicles (UAVs) for the detection of forest fires [6–8]: (1) In scenes where forest fires occur, there may be scenarios similar to fire spots or smoke, such as sunlight refracted in the forest, emitting light, and fog similar to fire points, which may cause misjudgments; (2) when drones are used for patrolling and detecting forest fires, they often maintain a relatively high and safe flying altitude. This results in the flames and smoke of early-stage fires occupying only a small proportion of the captured images, with the majority of the images being taken up by the background. Additionally, environmental factors significantly influence the effectiveness of the detection; (3) when the drone captures images from a high altitude, there may be multiple fire points and smoke in the fire, while traditional target-detection networks only filter each local feature, and cannot make good use of the global features of the entire image, which may lead to a large position deviation during localization.

Typically, object detection networks are categorized into two primary types: single-stage and two-stage networks. As object detection needs to predict the location and category of objects in images, these two methods both introduce a lot of prior information, such as by pre-generating some anchor boxes. In the one-stage model, common ones include SSD [9], DSSD [10], YOLO series [11], etc.; in the two-stage model, common ones include R-CNN [12], Fast R-CNN [13], Faster R-CNN [14], etc. These methods do not directly predict objects, but approximate using anchor boxes or proposal boxes. In the end, they will generate many prediction boxes of various sizes, so in the post-processing stage, non-maximum suppression (NMS) must be used to eliminate these redundant boxes. For aerial fire images shot using UAVs, their backgrounds are complex and the fire spots and smoke clouds are small. Common target-detection models' backbone networks usually use convolutional neural networks (CNNs) or a Vision transformer (ViT) [15]. Convolutional layers in CNNs typically employ convolution kernels of a fixed size, with each neuron

having the capacity to perceive only the local receptive field within the input image. When dealing with forest-fire images, a CNN's global information-extraction performance is poor. In addition, the CNN reduces the number of parameters by sharing convolution kernel parameters across the entire image, and its processing effect is not good for images with multi-scale structures and complex textures, such as forest-fire images. Although ViT can capture the correlation between tokens, it ignores the relationship between token groups, which may be limited when dealing with forest-fire images [16]. At the onset of a fire, the fire points and smoke clouds in the forest-fire images shot using UAVs are small, and compared to the whole picture, the features of the fire points and smoke are more influenced by global features, and using traditional ViT will cause problems such as the incomplete feature extraction of fire points and smoke.

In response to the above question, we propose a new forest-fire- and smoke-image object detection model called FSNet. FSNet transforms non-learnable components such as anchors and NMS into learnable components, eliminating the reliance on prior information and resulting in a simple yet effective end-to-end network. We add an Ebblock module after the backbone to allow each token in the image to interact with tokens and groups, and to generate group proxies, similar to transformers but without the need for complex operations like anchors and NMS. This ensures that each object generates only one prediction box, effectively removing redundant boxes. Through interactions between tokens and groups, Ebblock can effectively extract features of small fires and smoke in forest-fire images by perceiving the global image. Finally, by combining the proposed L_{forest} to select the optimal predicted boxes of the model and calculate their loss, the model continuously learns the feature information in forest-fire images. The experiment was conducted on three public forest-fire-image datasets, including the Flame dataset, the Corsican dataset, and the D-Fire dataset. Experimental results demonstrate that FSNet has achieved outstanding performance in forest-fire image object detection.

The key contributions of this work are summarized as follows:

1. A new forest-fire image object detection model, FSNet, is proposed. The model can effectively perceive the global features of forest-fire images and extract tiny features between fire points and smoke in forest-fire images;
2. YOCO is employed as a data-augmentation strategy to effectively enlarge the dataset without increasing the parameter count. This approach significantly enhances FSNet's capability in learning and recognizing features specific to forest-fire images. Additionally, based on the transformer, an EBblock attention module is proposed, which fuses the outputs between the layers of EBblock using feature pyramids to avoid the feature loss of small targets and ensure the model can fully extract the correlations between various features;
3. A loss function L_{forest} suitable for FSNet is introduced, selecting the optimal predicted boxes of the model and calculating their loss, while continuously allowing the model to learn feature information in forest-fire images;
4. Extensive experiments are conducted on three public forest-fire image datasets: the Flame dataset, the Corsican dataset, and the D-Fire dataset, demonstrating the outstanding performance of FSNet in detecting fire points and smoke in forest-fire images, as well as the effectiveness of each module.

2. Related Work

2.1. Object Detection Based on Deep Learning

As computer vision technology advances, artificial intelligence algorithms are increasingly being applied in the realm of object detection. At present, the field of object detection primarily bifurcates into two distinct approaches: single-stage and two-stage methods.

The one-stage algorithm is a regression-based object detection method, with the YOLO series models being one of the most typical representatives. The preliminary version of the YOLO model, as introduced by Redmond et al. [17], brings object detection to a new stage. The fundamental concept behind it involves utilizing the entire image as

the input for the model, directly pinpointing objects by regressing their positions and categories, which in turn endows the model with robust generalization capabilities. Unlike two-stage algorithms, YOLO does not need to generate candidate regions for feature extraction, classification, and regression but directly utilizes the information contained in the image. YOLOv2 [18] introduces an anchoring mechanism for local predictions and uses Darknet19 as the backbone network to improve the detection accuracy of small targets. YOLOv3 [19] proposes using three different scales of feature maps to enhance Darknet53, thereby improving the detection accuracy of objects to be detected. Both YOLOv4 [20] and YOLOv5 [21] make improvements based on YOLOv3. YOLOv5, proposed by Ultralytics in 2021, is currently one of the most commonly used object detection networks. It uses separate CSP modules in the backbone and neck parts to eliminate the redundancy of gradient information, thereby reducing computational complexity. In general, one-stage algorithms usually require generating anchors and extracting features in the network to predict the classification and position of objects. Anchors need to define prior boxes, which are manually designed to distribute at different positions and sizes in the image. Then, through a CNN network, these anchors are classified and regressed, and the final target-detection result is obtained through NMS. During the manual intervention process, difficulties may arise in fully covering the sizes and shapes of some hard-to-detect targets, leading to prediction errors and missed detections. Additionally, due to the highly similar situations between anchors of different sizes, achieving optimal results might require extensive training and adjustments.

The two-stage algorithm centers on a region-proposal-based approach for object detection, with R-CNN, Fast R-CNN, and Faster R-CNN standing out as its most notable exemplars. Typically, the operation of two-stage detection models is segmented into two phases: initially, they incorporate a region-proposal-generation network within the CNN to produce high-quality candidate boxes for object detection; subsequently, they employ specialized sub-networks to refine and classify these candidate boxes. In 2014, Girshick et al. [12] proposed the R-CNN algorithm. The algorithm extracts features from the image using a sliding window strategy; however, this operation results in a lot of redundant information, leading to the low detection efficiency of the model. Building upon R-CNN, Girshick et al. [13] propose an optimized algorithm—the Fast R-CNN algorithm. This algorithm borrows the idea of spatial pyramid Pooling (SPP) and adds an ROI Pooling layer throughout the network structure to generate uniformly sized feature vectors, thereby increasing detection speed. Based on the previous work, Ren et al. [14] propose the Faster R-CNN algorithm. The model utilizes a region-proposal network (RPN) to generate a large number of anchor boxes, thus improving processing efficiency. Furthermore, the algorithm has a small model size, fast candidate-box generation speed, and does not affect the accuracy of the algorithm. However, the design of these anchor boxes typically needs to be selected based on the distribution of the dataset's targets. Choosing inappropriate anchor boxes may lead to performance degradation. Additionally, since the features of small targets in convolutional neural networks will be reduced to very small sizes, Faster R-CNN faces challenges in detecting small targets, potentially leading to decreased detection performance. In practical forest-fire prevention, UAVs usually patrol at high altitudes to monitor forest fires. In images captured during the early stages of a fire, the proportion of fire points and smoke is extremely small compared to the whole image. Therefore, using Faster R-CNN may not achieve high-precision detection results.

2.2. Forest Fire and Smoke Detection

Forest fires are usually accompanied by a large amount of smoke. The detection task for forest fires mainly focuses on detecting the occurrence of fire spots and smoke [22]. Three primary methodologies exist for detecting fire spots and smoke: (1) manual detection techniques and sensor-based detection; (2) detection approaches utilizing machine learning; and (3) detection strategies grounded in deep learning.

Forest-fire smoke detection primarily employs two basic methods: manual observation and the use of sensors. Manual detection hinges on the physical patrol of forestry personnel within the forest. Once a fire is discovered, they report to the fire department promptly through communication means such as walkie-talkies for emergency rescue. Nevertheless, this approach presents several limitations, such as high error rates, restricted patrol zones, elevated costs, and limited coverage. These factors collectively fall short of the required speed and accuracy standards for effective forest-fire detection [23]. Another method is to use sensors, including optical smoke sensors, ion smoke sensors, etc. However, the effectiveness of sensors is influenced by environmental elements like distance, shading, and angle, which can impact the accuracy of smoke humidity or particle sampling. In addition, the internal structure of sensors may cause serious time delays. Furthermore, sensors are expensive, operationally complex, and have limited applications [24]. Therefore, manual monitoring methods and sensor methods face certain difficulties in meeting the requirements of forest-fire smoke detection. Traditional manual detection and sensor-detection methods have disadvantages such as high costs, delayed response times, and limited applicability.

Early research mainly focus on using traditional image-processing algorithms to analyze the spatiotemporal relationships in image sequences. Ho [25] proposes an integrated machine vision analysis method for monitoring systems, deriving the likely locations of smoke through motion-history detection algorithms, and calculating the area and perimeter of smoke by combining spatial probability density and turbulent phenomena of smoke. Chen et al. [26] conduct an in-depth analysis of early-fire-smoke video segments. They construct a decision tree for smoke recognition based on the distribution pattern of smoke color estimation and dynamic diffusion changes in smoke color for early-fire alarming systems. Gubbi et al. [27] introduced an alternative approach that utilizes wavelet transform to decompose the gradient, mean, and variance of single-frame images in sequences. These decomposed attributes are then merged to depict the characteristics of fire smoke. Following this, support vector machines are applied to ascertain the presence of fire spots or smoke in the images. Yuan et al. [28] introduce an improved fire-smoke-detection method considering both local and global texture features of images. However, accurately measuring local and global boundaries remains a challenge. Setting a unified threshold applicable to the pixel level in this research method is difficult because balancing the differences between close-range and distant smoke is challenging, which may lead to false alarms or missed detections. Töreyin et al. [29] employed a combined background-estimation technique for identifying moving pixel regions in video footage. They suggested the use of a two-dimensional wavelet transform to extract energy levels from high-frequency channels, which aids in determining the existence of fire spots and smoke. However, due to the computational complexity of feature selection and the uncertainty of smoke, these traditional image-processing methods perform poorly in adapting to complex and ever-changing forest environments. Machine learning detection methods also suffer from issues such as incomplete feature extraction and high false-alarm rates.

As computer vision technology rapidly advances, employing deep learning methodologies for forest-fire-image detection during UAV patrols has emerged as a prominent area of research [30]. A CNN framework, grounded in GoogleNet, for fire detection in surveillance videos, along with several convolutional neural network-based algorithms for fire-image detection, opens new avenues for the real-time monitoring of forest-fire imagery. Furthermore, experiments show that YOLOv3 outperforms two-stage object-detection networks like Faster R-CNN in forest-fire-image detection, marking the first application of YOLO series algorithms in this context.

Furthermore, an innovative neural network guided by feature entropy has been introduced for forest-fire detection, designed to equalize the content complexity across various training samples [31]. An approach called SAP is introduced for weakly supervised forest-fire segmentation in UAV imagery, which enhances foreground awareness for distinguishing object categories in images [1]. MMFNet [32] presents a mixed-attention

multiscale convergence coordinated pyramid network and a fast robust NMS for rapid forest-fire-smoke detection. The latest YOLOV8 introduces an anchor-free method as an alternative to traditional anchor methods, avoiding issues related to low anchor accuracy. However, it performs poorly in scenarios with small targets and potential target occlusion in forest-fire images.

Current models for forest-fire-image detection rely on generating anchors or proposals for detection [33]. In this article, an innovative detection model named FSNet is proposed for detecting fire spots and smoke in forest-fire images. This model utilizes the YOCO data-augmentation method to enhance the dataset and improve the model's feature-learning capability. Unlike traditional CNN and transformer methods for extracting image features, the EBblock module is introduced to focus on the correlation between features. Since fire spots and smoke always appear simultaneously in forest-fire images, the EBblock module effectively prevents the misidentification of fire spots or smoke. Additionally, we introduce the L_{forest} loss function for FSNet, selecting the optimal predicted box of the model and calculating its loss while continuously learning feature information in forest-fire images.

3. Methods

3.1. Network Structure of Proposed FSNet

The comprehensive architecture of our proposed forest-fire-image detection model, FSNet, is illustrated in Figure 1. For the input image, data augmentation is first performed using the YOCO method. The image then goes through a convolutional neural network (ResNet101) to extract features, which are serialized into a one-dimensional sequence. This sequence is then fused with positional encoding information and fed into the Ebblock attention module. After passing through four stages of the Ebblock, N prediction boxes are generated. The model calculates the matching loss between these N prediction boxes and all ground truth (GT) boxes. Subsequently, the binary matching loss L_m between the N prediction boxes and GT boxes is computed to select the best-matching prediction box for each object. If a forest-fire image contains one fire spot and one smoke cloud, only two boxes representing them are selected as the foreground, while the remaining $N-2$ boxes are labeled as the background. Finally, calculate the L_H between the predicted box and the label to train the model.

3.2. Data Augmentation Based on YOCO

Common image-data-augmentation methods include horizontal flipping, vertical flipping, color distortion, Gaussian blur, and random erasing. These methods operate at the image level, altering attributes such as viewpoint, color, clarity, etc., to enhance the model's generalization ability and robustness. By imitating human visual perception, these methods retain the global semantic information of images, enabling models to adapt to different scenarios. When humans identify objects, they often utilize local image information, such as specific regions or details within the image. These local cues are referred to as strong natural signals in images and serve as important cues for the human visual system to recognize and understand images.

Patches refer to local regions within an image that carry vital information within the overall context of the image, aiding in enhancing the model's understanding and analytical capabilities of the image content. Similar to ViT segmenting images into non-overlapping patches for the network input, utilizing non-image-level data augmentation offers several advantages: increasing diversity in local regions, enhancing overall image variability, and encouraging the network to extract useful information from partial images. For small targets like fire spots and smoke clouds in forest-fire images captured using UAVs, which may appear in specific regions of the image, we employ the YOCO [34] technique for data augmentation for the forest-fire images. The process is illustrated in Figure 2.

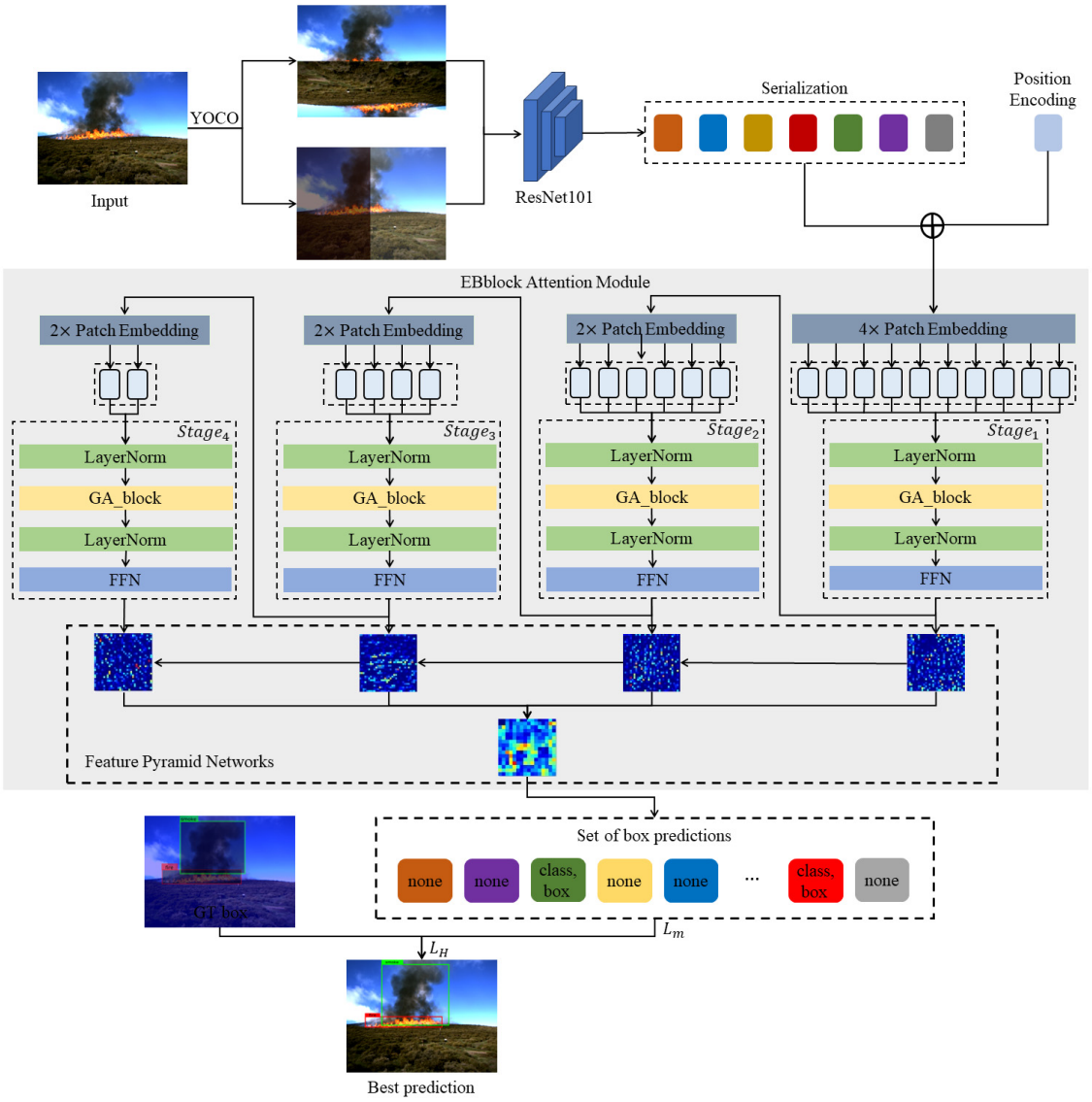

Figure 1. The overall flowchart of the FSNet model.

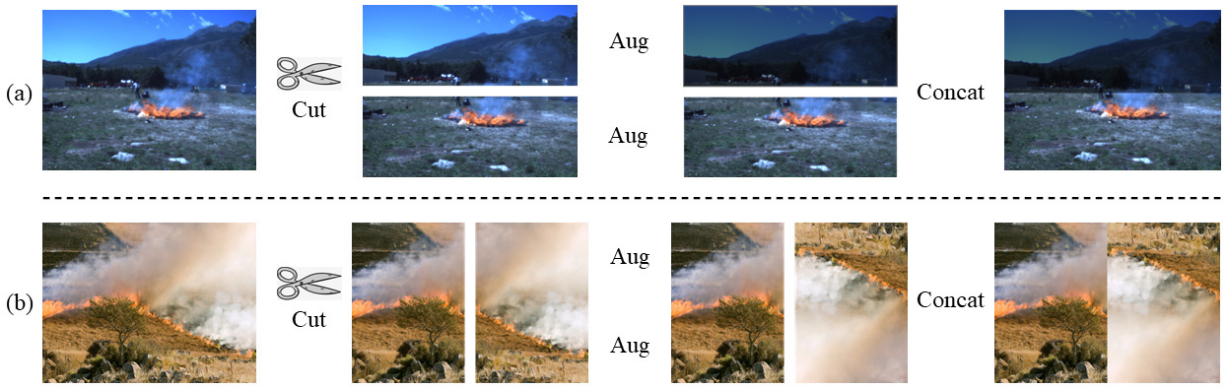

Figure 2. Data augmentation. (**a**) YOCO divides the forest-fire image into two equal parts along the width dimension and applies color jitter enhancement to the upper part of the image and then merges the two parts to create the augmented forest-fire image; (**b**) the forest-fire image is divided into two equal parts along the height dimension. The right part undergoes a horizontal-flipping operation, and then the two parts are merged to form the augmented forest-fire image.

In this context, we represent the model's input image as $X_0 \in R^{C\times H\times W}$. Therefore, the data-augmentation method $A(\cdot)$ can be represented as

$$A(\cdot): \ R^{C\times H\times W} \rightarrow R^{C\times H\times W} \tag{1}$$

$$X = A(X_0) \tag{2}$$

where X represents the image after data augmentation. The YOCO data-augmentation method first divides the forest-fire image into two parts evenly along the width or height dimension, which can be represented by the following formula:

$$[X_1, X_2] = Cut_W(X_0) \tag{3}$$

or

$$[X_1, X_2] = Cut_H(X_0) \tag{4}$$

Here, $A(\cdot)$ can be specifically represented as

$$A(X_0) = A([X_1, X_2]) = concat[a_1(X_1), a_2(X_2)] \tag{5}$$

where $a_1(\cdot)$ and $a_2(\cdot)$ represent two different common image-level data-augmentation methods.

3.3. EBblock Attention Module

EBblock is a transformer-based module that adopts a structure with four stages. The structure of EBblock's stage$_i$ is depicted as each stage of EBblock in Figure 1. In the initial layer of EBblock, image embedding is achieved using two consecutive 3×3 convolutional layers with a stride of 2 and two 3×3 convolutional layers with a stride of 1, resulting in a $4\times$ image embedding. In the subsequent three stages, $2\times$ image embedding is achieved using 3×3 convolutional layers. Each stage contains multiple encoder blocks, each of which includes the GA_block to be introduced below, a Feed-Forward network (FFN), layer normalization, and identity shortcuts. Finally, the results of each stage are fed into a feature pyramid for fusion and prediction-box output.

The GA_block is a crucial component of EBblock used for information interaction and feature fusion within images. The four stages of EBblock and multiple encoder blocks contribute to gradually extracting higher-level features and ultimately generating prediction boxes. Overall, the EBblock model combines the strengths of convolutional neural networks and transformer architecture to effectively handle image tasks. The overall framework of the GA_block is illustrated in Figure 3.

In the GA_block, we replace some components of $Query(Q)$, $Key(K)$, and $Value(V)$ with the aggregated values of the entire group, where the feature x can be represented by Q, K, and V. We partition the Q, K, and V components into 5 segments, denoted as x_n^q, x_n^k, x_n^v representing the segments divided by Q, K, V, respectively. We aggregate four of these segments to generate group representatives Q', K', and V' as follows:

$$Q' = Agg^n\left(x_n^q\right), n \in [1, 2, 3, 4] \tag{6}$$

$$K' = Agg^n\left(x_n^k\right), n \in [1, 2, 3, 4] \tag{7}$$

$$V' = Agg^n(x_n^v), n \in [1, 2, 3, 4] \tag{8}$$

where $Agg(\cdot)$ represents an operation process based on a sliding window. The aggregated feature can be expressed as x_n:

$$x_n = Agg^n(x) \tag{9}$$

Next, attention calculation is performed on the group representatives by inputting Q', K', and V' into the attention function to generate the final output.

$$Attention = \frac{Q'}{\sqrt{d}} Softmax\left(K'^{T} V'\right) \tag{10}$$

where $Softmax(\cdot)$ represents the normalized exponential function, and K'^{T} denotes the transpose of K'.

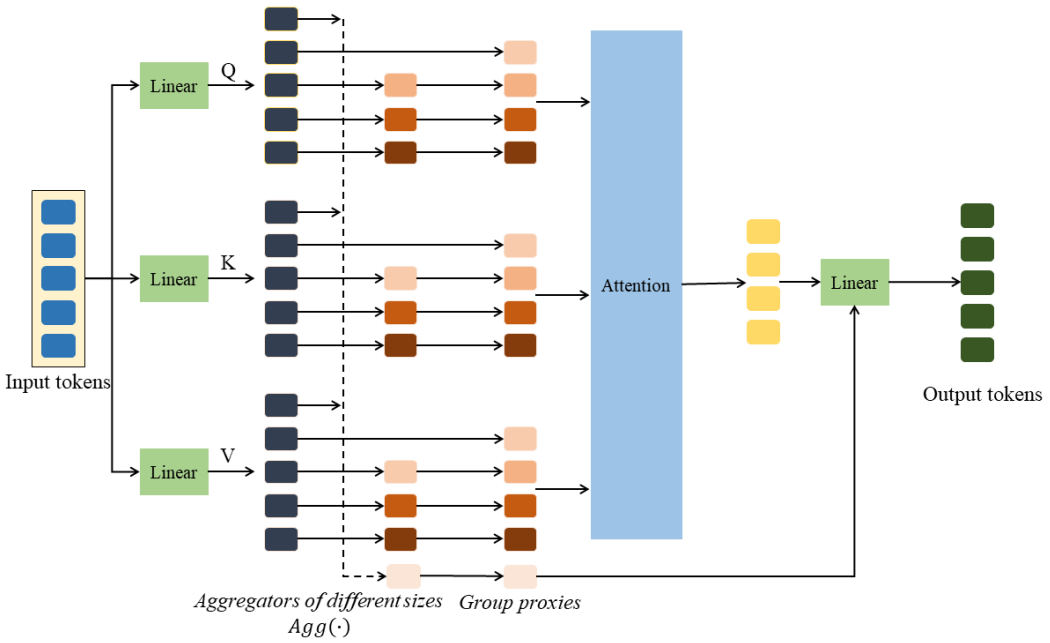

Figure 3. The overall framework of the GA_block.

During the aggregation process, the resolution of the features does not change, which means that the GA_block can extract more fine-grained features for attention calculation without altering the low spatial resolution. This feature-extraction method is superior to methods that rely on smaller feature scales.

EBblock uses deep convolutions to implement the aggregator $Agg(\cdot)$, with these convolutional layers having different kernel sizes. Since the input in EBblock is group representatives, multiple tokens of size $k \times k$ (where k represents the kernel size of $Agg(\cdot)$) can be associated simultaneously to ensure the model can comprehensively and adequately capture image features. By using aggregators with different kernel sizes, group representatives are constructed based on sliding window operations to ensure that each token can blend different-sized groups and information of varying granularity. This process involves employing different kernels at different stages for aggregation, concatenation, and output.

In the final pre-attention branch, no aggregator is used to further diversify the structure and make it an identity mapping. In addition to this branch, a non-attention branch is constructed with an aggregator but no attention. Finally, integration layers with normalization and activation linear projections are used for mixing. To utilize the predictions from the four stages of EBblock simultaneously, the outputs of the four stages are fed into a feature pyramid network (FPN), where they are fused, respectively, to produce N prediction boxes. Here, we set the number of prediction boxes N to 20, which is generally greater than the number of targets in the image.

3.4. Loss Function

To address the N predicted-output boxes, it is necessary to compute the optimal bipartite matching loss between these boxes and the actual ground truth objects. Subsequently, we undertake loss optimization tailored to the specific objects, namely the bounding boxes.

We use y to represent the set of ground truth GT_boxes of real objects, then the set of N prediction boxes can be represented by y':

$$y' = \{y'_i\}_{i=1}^{N} \tag{11}$$

where N is typically greater than the number of objects present in the image. Therefore, we treat y as a set with N elements, filling positions corresponding to non-existent objects with $\varnothing$ (None).

In order to determine the optimal bipartite matching between these two sets, it is essential to identify a permutation of N elements, denoted as $\sigma \in \vartheta_N$, which results in the lowest matching cost:

$$\hat{\sigma} = arg\min_{\sigma \in \vartheta_N} \sum_{i}^{N} L_m\left(y_i, y'_{\sigma_{(i)}}\right) \tag{12}$$

where $L_m\left(y_i, y'_{\sigma_{(i)}}\right)$ denotes the cost of pairwise matching between the actual object y_i and the predicted object at the index $\sigma(i)$.

The calculation of the matching cost incorporates factors such as the accuracy of category prediction and the similarity between the predicted and actual values. The element i in the real object can be represented as y_i:

$$y_i = (z_i, o_i) \tag{13}$$

where z_i signifies the target class label (typically $\varnothing$), and $o_i \in [0,1]^4$ specifies the actual center coordinates, height, and width, all relative to the overall size of the image. For the prediction located at index $\sigma(i)$, the class probability for z_i is defined as $\hat{p}_{\sigma(i)}(z_i)$, and the corresponding predicted value is represented as $\hat{o}_{\sigma(i)}$. Therefore, the matching cost $L_m\left(y_i, y'_{\sigma_{(i)}}\right)$ between the predicted and actual values can be expressed as

$$L_m\left(y_i, y'_{\sigma_{(i)}}\right) = -\lessdot_{\{z_i \neq \varnothing\}} \hat{p}_{\sigma(i)}(z_i) + \lessdot_{\{z_i \neq \varnothing\}} L_{forest}\left(o_i, \hat{o}_{\sigma(i)}\right) \tag{14}$$

Prior to determining the loss function, it is crucial to compute the Hungarian loss for each pair of matched elements. The loss function, denoted as $L_H(y, y')$, encompasses a linear combination of the negative log-likelihood for class predictions and the bounding-box loss, which is to be defined subsequently:

$$L_H(y, y') = \sum_{i=1}^{N}\left[-log\hat{p}_{\hat{\sigma}(i)}(z_i) + \lessdot_{\{o_i \neq \varnothing\}} L_{forest}(o_i, \hat{o}_{\hat{\sigma}}(i))\right] \tag{15}$$

where $\hat{\sigma}$ is the optimal assignment calculated in Equation (12). In calculating the matching cost, we use the probability $\hat{p}_{\hat{\sigma}(i)}(z_i)$ instead of the log probability, which allows the category prediction term to be unified with $L_{forest}(\cdot,\cdot)$, resulting in better model performance.

The second component of both the matching cost and the Hungarian loss is represented by $L_{forest}(\cdot)$, which serves to assess the accuracy of the bounding-box matches. Unlike many detectors, we use an IoU loss and impose a loss on $\hat{o}$:

$$L_{forest}\left(o_{\sigma(i)}, \hat{o}_i\right) = \lambda_{iou} L_{iou}\left(o_{\sigma(i)}, \hat{o}_i\right) + \lambda_{L1}\|o_{\sigma(i)} - \hat{o}_i\|_1 \tag{16}$$

where $\lambda_{iou}, \lambda_{L1} \in R$ are hyperparameters, L_{iou} is the generalized IoU, calculated as

$$L_{iou}\left(o_{\sigma(i)}, \hat{o}_i\right) = 1 - \left(\frac{\left|o_{\sigma(i)} \cap \hat{o}_i\right|}{\left|o_{\sigma(i)} \cup \hat{o}_i\right|} - \frac{\left|B\left(o_{\sigma(i)}, \hat{o}_i\right) \backslash o_{\sigma(i)} \cup \hat{o}_i\right|}{\left|B\left(o_{\sigma(i)}, \hat{o}_i\right)\right|}\right) \tag{17}$$

where $|\cdot|$ denotes the "area", and the intersection over union (IoU) of the predicted bounding-box coordinates is used as shorthand for the box itself. The area for the IoU is calculated through the minimum/maximum values of the linear functions of $o_{\sigma(i)}$ and $\hat{o}_i$, making the loss sufficiently robust for stochastic gradients. $B\left(o_{\sigma(i)}, \hat{o}_i\right)$ represents the smallest box containing both $o_{\sigma(i)}$ and o_i.

4. Experiments and Results

4.1. Dataset

In relation to our proposed FFNet model, we conducted experiments on three publicly available datasets—the Flame dataset, the Corsican dataset, and the D-Fire dataset. The Flame dataset, obtained from drone surveillance at high altitudes, encompasses various scenarios of forest-fire occurrences, including instances where fire points are small and partially obscured by vegetation, as well as different manifestations of forest fires across seasons. Moreover, this dataset includes images captured from multiple perspectives including top-down, oblique, and horizontal views. The Corsican dataset focuses on forest-fire scenes with closer distances and more prominent fire points, enabling the clear visualization of fire details in the images. The D-Fire dataset comprises 21,000 images related to smoke and flames. From this dataset, we selected 5843 images of forest fires for our experiments, covering various scenes captured from both distant and close distances and providing a rich data foundation for our experimentation. These three datasets cover a range of scenarios that may occur during forest fires, making them representative of such incidents. Below are partial images from two of the datasets, as shown in Figure 4.

Figure 4. Example images of the Flame, Corsican, and D-Fire datasets.

4.2. Evaluation Metrics

For an objective assessment of FSNet's capability in detecting fire points and smoke in forest-fire images captured using UAVs, we employ three key metrics: Recall rate (R), mean average precision (mAP), and frames per second (FPS) to evaluate the efficacy of various fire-detection methodologies. The metrics of precision (P) and Recall (R) are defined as follows:

$$P = \frac{TP}{TP + FP} \tag{18}$$

$$R = \frac{TP}{TP + FN} \tag{19}$$

where True positives (TPs) represent the number of fire points or smoke clouds correctly detected using the FSNet model, false positives (FP) represent the number of instances where FSNet incorrectly detects fire points or smoke, and false negatives (FNs) represent the number of instances where FSNet fails to detect fire points or smoke.

Average precision (AP) can be obtained from the relationship between P and R:

$$AP = \sum_{i=1}^{n-1} (r_{i+1} - r_i) P_i(r_{i+1}) \tag{20}$$

where $P_i(r)$ represents the maximum precision in the interval $[r, 1]$. Here, we use mAP as the standard to measure FSNet's ability to detect fire points and smoke, which can be expressed as

$$mAP = \frac{\sum_{i-1}^{k} AP_i}{k} \tag{21}$$

In this formula, k signifies the total count of distinct categories within forest-fire images, encompassing fire points, smoke, and the background. Additionally, AP_i denotes the average precision (AP) value corresponding to class i.

FPS (frames per second) is utilized as a metric to gauge the processing speed of FSNet when handling forest-fire images. FPS can be understood as the refresh rate of forest-fire images, indicating how many frames of forest-fire images FSNet can process per second. Assuming FSNet takes b seconds to detect one forest-fire image, the formula to calculate FPS is

$$FPS = 1/b \tag{22}$$

4.3. Implementation Details

The experiments were performed on a system running Ubuntu 18.04, equipped with an RTX NVIDIA 3090 24G GPU. The software environment included PyTorch 1.8.0 and Python 3.9. The settings for the model-training hyperparameters were as follows: a batch size of 16, an initial learning rate of 0.01, a weight decay set at 0.0005, and the use of the AdamW optimizer in conjunction with a cosine learning-rate schedule.

In this study, each of the three datasets was divided individually, allocating 20% of the data for the test set and 80% for the training set. Specifically, the Flame dataset comprises a total of 2003 forest-fire images, out of which 1602 were utilized for training and 401 for testing. The Corsican dataset contains 1135 forest-fire images, with 908 designated for training and 227 for testing. Similarly, the D-Fire dataset includes 5843 forest-fire images, with 4674 assigned for training and 1169 for testing purposes.

4.4. Comparison with Other Target-Detection Algorithms

We trained the FSNet and the latest object detection networks, including MS-DETR [35], YOLOX [36], DETR [37], and YOLOV7 [38], on the Flame, Corsican, and D-Fire datasets, respectively. The experimental results are detailed in Tables 1–3.

Table 1. Mean average precision (*mAP*), Recall (*R*), and frames per second (*FPS*) evaluation metrics on the Flame test set.

Model	*mAP*	*R*	*FPS*
MS-DETR	93.6%	88.3%	82.3
YOLOX	86.8%	83.6%	79.7
DETR	79.9%	74.1%	67.6
YOLOV7	88.4%	85.5%	89.3
FSNet	**97.2%**	**93.9%**	**91.2**

Table 2. Mean average precision (*mAP*), Recall (*R*), and frames per second (*FPS*) evaluation metrics on the Corsican test set.

Model	*mAP*	*R*	*FPS*
MS-DETR	86.7%	85.8%	80.0
YOLOX	85.2%	83.4%	78.8
DETR	80.6%	80.1%	64.3
YOLOV7	84.7%	84.9%	85.2
FSNet	**87.5%**	**87.3%**	**90.7**

Table 3. Mean average precision (*MAP*), Recall (*R*), and frames per second (*FPS*) evaluation metrics on the D-Fire test set.

Model	*mAP*	*R*	*FPS*
MS-DETR	87.5%	86.2%	82.4
YOLOX	86.2%	83.6%	78.3
DETR	84.4%	79.9%	65.6
YOLOV7	85.6%	82.1%	89.6
FSNet	**94.3%**	**90.8%**	**92.6**

In the experimental results, YOLOX, and YOLOV7 belong to the YOLO series of algorithms, which generate multiple prediction boxes and then select the optimal ones through non-maximum suppression (NMS). On the other hand, MS-DETR and DETR are end-to-end models that only generate a single prediction box. The FSNet model adopts a prediction-box-generation approach similar to MS-DETR and DETR instead of using the multiple-prediction-box method of the YOLO series. However, FSNet effectively utilizes information interaction and feature fusion in the backbone network EBblock, which leads to an outstanding performance in forest-fire-image object detection.

Due to the small size and occlusion of fire point objects in the Flame dataset, the detection difficulty is high. FSNet can leverage token-to-token, token-to-group, and group-to-group correlations during feature extraction. For small fire point targets, this model performs much better than other models. FSNet's *mAP* is 3.6%, 10.4%, 17.3%, and 8.8% higher than MS-DETR, YOLOX, DETR, and YOLOV7, respectively. Its *Recall* is 1.5%, 3.6%, 13.2%, and 1.5% higher than MS-DETR, YOLOX, DETR, and YOLOV7, respectively. Its *FPS* is 8.9, 11.5, 23.6, and 1.9 higher than MS-DETR, YOLOX, DETR, and YOLOV7, respectively. The results of the FSNet model in detecting small targets or occluded fire points are quite impressive compared to other networks. For the Corsican dataset and D-Fire dataset, which have larger fire point targets, the detection difficulty is not as high as in the Flame dataset. However, FSNet still shows improvement compared to the other four networks in fire point detection.

Tables 1–3 present the experimental results of each model on the Flame, Corsican, and D-Fire dataset. To provide a more intuitive comparison of the detection performance of each model on these datasets, we created visualizations of the detection results for some forest-fire images. The visualized results are shown in Figures 5–7.

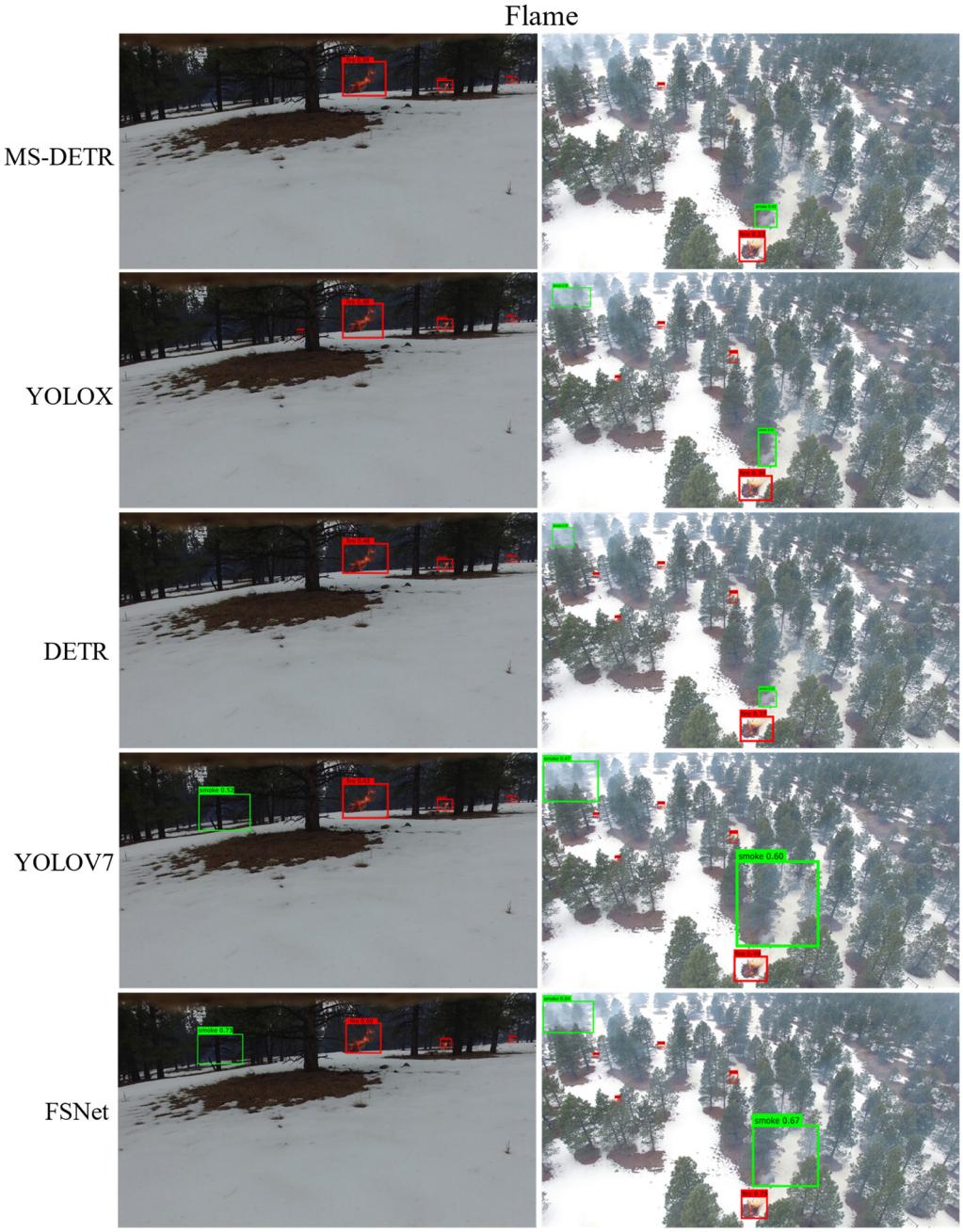

Figure 5. The visualized results of smoke recognition by various models on the Flame dataset. The number in the green detection box indicates the probability of detecting smoke, and the red detection box indicates the probability of detecting fire.

Figure 5 shows the visualized results of smoke recognition by various models on the Flame dataset. The Flame dataset primarily consists of images captured using UAVs at relatively high or far distances of forest-fire scenes. These images commonly exhibit issues such as small smoke or fire point targets and occlusions, which are typical scenarios encountered during actual drone patrols. From Figure 5, it is evident that MS-DETR and YOLOX have more missed detections for small targets of smoke and fire points at far distances, failing to meet the requirements of drone patrols for fire detection. DETR performs reasonably well in identifying fire points at far distances, but its detection performance for smoke is mediocre, especially for relatively sparse smoke, which tends to occur at the early stages of a fire. The timely detection of such sparse smoke can effectively prevent large-scale fires. Compared to YOLOV7, both YOLOV7 and the FSNet model are capable of detecting sparse smoke

and small fire points. However, YOLOV7 exhibits an inferior performance in detecting the contours of smoke and fire points compared to FSNet. In forest-fire prevention, detecting the contours of smoke and fire points is crucial for assessing the scale of a fire and allocating corresponding manpower and resources. However, the contours of smoke and fire points detected by YOLOV7 are evidently larger than the actual situation, potentially leading to a waste of manpower or resources.

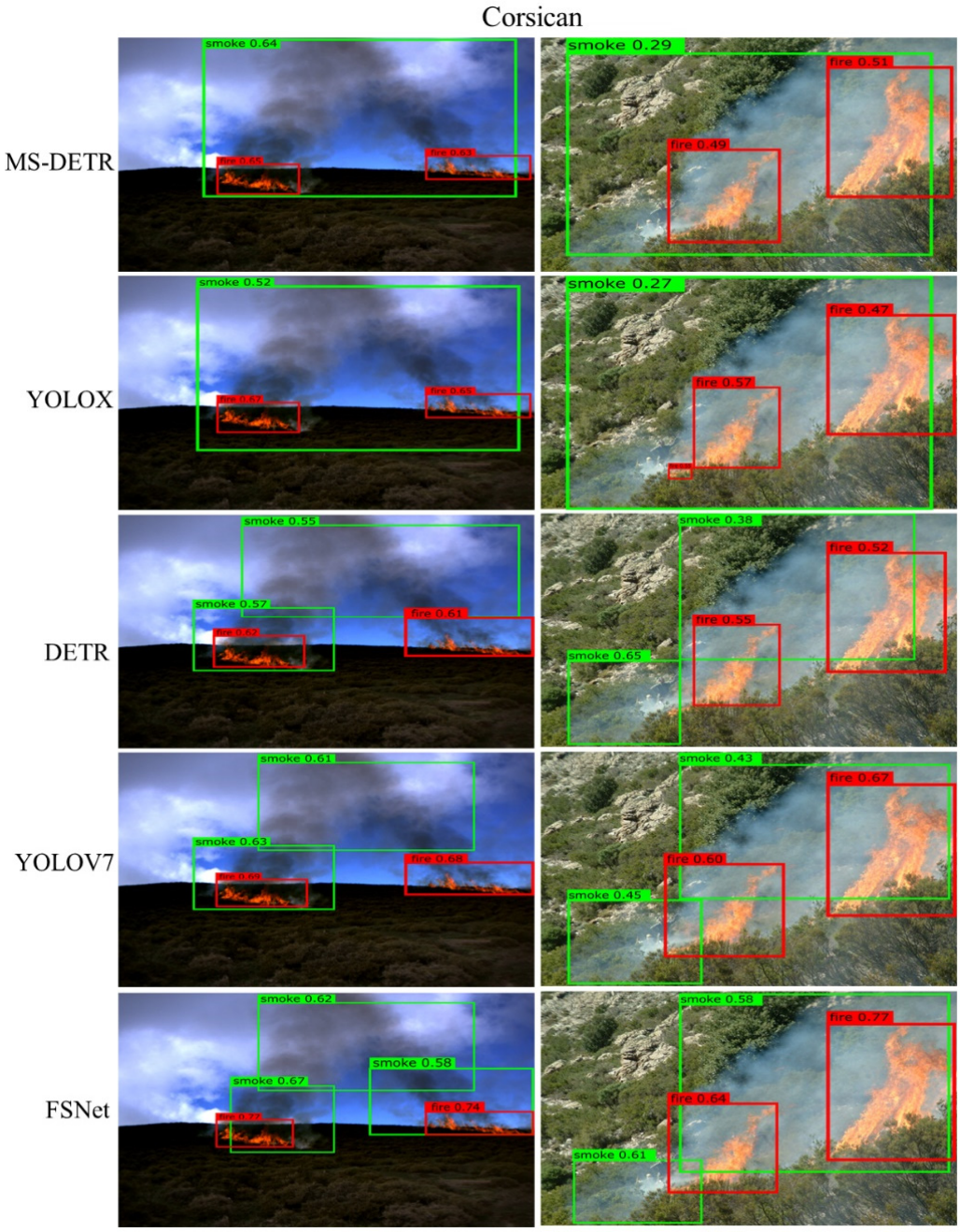

Figure 6. The visualized results of smoke recognition by various models on the Corsican dataset. The number in the green detection box indicates the probability of detecting smoke, and the red detection box indicates the probability of detecting fire.

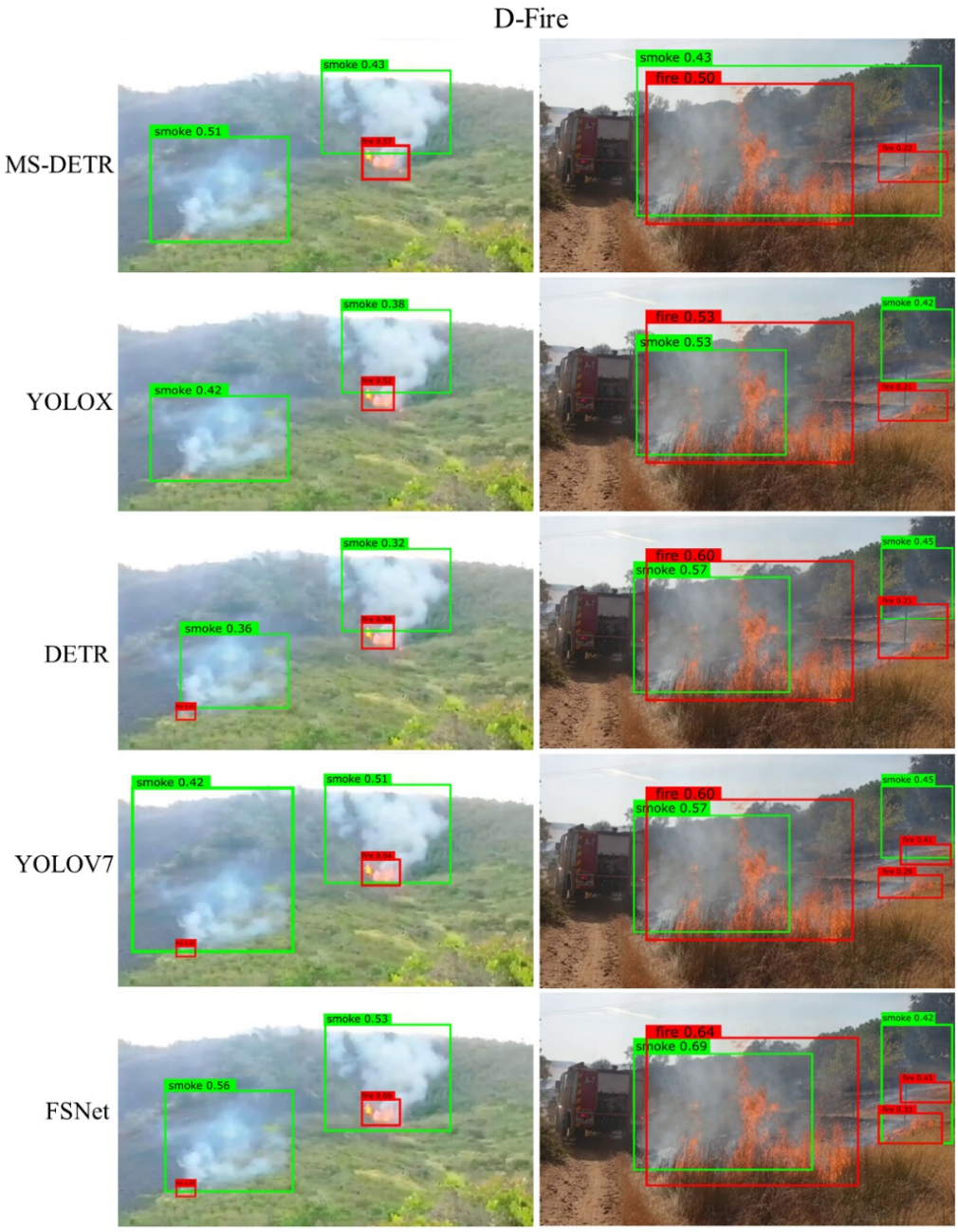

Figure 7. The visualized results of smoke recognition using various models on the D-Fire dataset. The number in the green detection box indicates the probability of detecting smoke, and the red detection box indicates the probability of detecting fire.

Figure 6 presents the experimental visualization results of the models on the Corsican dataset. Images in the Corsican dataset generally exhibit characteristics such as large smoke and fire point ranges. The MS-DETR, YOLOX, DETR, YOLOV7, and FSNet models can all detect smoke and fire points in the images relatively accurately. However, from the visualized detection results of each fire point and smoke cloud, it is evident that FSNet achieves higher precision in detecting fire points and smoke compared to the other four models. In fire scenes captured in the Corsican dataset, which often represent the mid-stage of a fire where the fire has spread to some extent, smoke and fire points captured using UAVs at high altitudes tend to have larger ranges. At this stage, common detection networks trained to detect smoke and fire points do not face significant challenges. For forest workers, the focus is on determining the specific location and assessing the affected

area based on the size and position of the fire points. Additionally, assessing the direction of the smoke helps predict the next steps in the fire's development. Therefore, simply detecting smoke and fire points is insufficient to meet the requirements at this stage. Our proposed FSNet model accurately locates smoke and fire points in such scenarios. It excels in detecting smoke by identifying different clusters of smoke, unlike MS-DETR and YOLOX, which detect smoke as a single entity. This capability of FSNet is crucial as it enables a more detailed analysis of the fire situation, aiding forest workers in making informed decisions based on the specific characteristics of the smoke and fire points detected.

Figure 7 illustrates the visualization of the experimental results of the models on the D-Fire dataset. The D-Fire dataset contains both long-range and close-range forest-fire smoke data, as well as scenes with backgrounds similar to smoke or fire points. The two visualized images selected have certain representativeness. In the left column of Figure 7, there is fog in the mountainous background, which may affect smoke detection. In the right column of Figure 7, the fire occurred during autumn dusk, and the color of the entire background is similar to that of the fire points, which may affect the detection of fire points. From the visualization of the entire experimental results, it is evident that YOLOV7 is greatly disturbed when detecting smoke, while MS-DETR, YOLOX, and DETR are disturbed by dry grass in the background when detecting fire points. This leads to the lower detection accuracy of these models. However, the EBblock added to the FSNet model fully utilizes the group information between pixels, effectively avoiding the risk of background confusion in images with significant distractions. From the detection results in Figure 7, it is clear that FSNet achieves higher detection accuracy than the other four models.

4.5. Ablation Experiments

To validate the effectiveness of the data-augmentation method YOCO, the attention module EBblock, and the proposed loss function in the FSNet model, we conducted ablation experiments. The experimental results are summarized in Table 4.

Table 4. Effectiveness of the components.

Backbone	YOCO	EBblock	L_{forest}	Flame *mAP*	Corsican *mAP*	D-Fire *mAP*
√	-	-	-	79.9%	80.6%	84.4%
√	√	-	-	82.6%	81.8%	87.3%
√	-	√	-	88.5%	84.3%	90.4%
√	-	-	√	83.7%	83.5%	85.9%
√	√	√	-	90.6%	86.8%	92.5%
√	√	-	√	85.9%	83.9%	88.3%
√	-	√	√	95.7%	87.2%	93.7%
√	√	√	√	**97.2%**	**87.5%**	**94.3%**

After incorporating the data-augmentation method YOCO, the attention module EBblock, and L_{forest} into our model, we conducted experiments on the Flame, Corsican, and D-Fire dataset. When only YOCO or L_{forest} is added, there is a slight improvement in the experimental results on all three datasets, but the improvement is not significant. However, when EBblock is added alone, there is a more noticeable improvement in the experimental results on all three datasets. Furthermore, when YOCO, L_{forest}, and EBblock are all added, the model achieves *mAP* scores of 97.2%, 87.5%, and 94.3% on the Flame, Corsican, and D-Fire dataset, respectively. This indicates that the addition of YOCO, L_{forest}, and EBblock has a synergistic effect on the object detection of forest-fire images, and all three components contribute to improving the accuracy of object detection.

5. Discussion

Forest fires have profound negative impacts on ecosystems, human health, and socio-economic aspects [39]. Effective fire prevention and control are crucial for maintaining

ecological balance, achieving the sustainable development of human society, and ensuring the normal functioning of ecosystems. Despite advancements in detection technology, detecting forest-fire smoke remains challenging [40,41]. Further research is therefore needed in forest-fire smoke detection.

While Unmanned Aerial Vehicle (UAV) patrols have become the primary method for forest-fire prevention, existing detection technologies often struggle to cope with the complexity of forest-fire images captured from high altitudes [3–5]. For example, Rahman et al. [42] utilized the SSD model, leveraging texture and color information, to equip UAVs with high-speed and high-accuracy capabilities. Challenges include remote locations, small fire spots, light-colored smoke targets, and complex background environments. FSNet incorporates innovative components to enhance detection accuracy and robustness. The YOCO data-augmentation method enriches the dataset, improving the model's ability to identify specific features of forest-fire images. Additionally, the EBblock attention module based on the transformer framework facilitates comprehensive feature extraction by fostering patch annotations and inter-group correlations.

Compared to earlier research methods, FSNet offers significant advantages. Traditional manual detection and sensor-based methods suffer from high costs, delayed response times, and limited applicability [43]. Similarly, earlier research primarily relied on traditional image-processing algorithms, which often struggled to adapt to the complex and evolving forest environment. Ho [25] proposed an integrated machine vision analysis method for monitoring systems. Pérez-Porras et al. [44] used machine learning techniques and SVM, RF, MLP, and Logistic regression (LR) algorithms in detecting forest fires. Although machine learning detection methods have improved, challenges such as incomplete feature extraction and high false-alarm rates persist. In contrast, FSNet utilizes deep learning methods tailored specifically for forest-fire detection during UAV patrols. Unlike previous methods, FSNet integrates the YOCO data-augmentation method and eblock attention module to enhance feature learning and correlation extraction. By effectively preventing the misidentification of fire spots and smoke, FSNet outperforms traditional CNN and transformer methods. While recent advancements, such as MMFNet [32] and YOLOV8, introduce innovative techniques, they still rely on generating anchors or proposals for detection, which may lead to issues related to low anchor accuracy and occlusion in forest-fire images. In contrast, FSNet's innovative approach eliminates the need for anchors or proposals, providing a more efficient and accurate detection mechanism.

However, despite the progress made, FSNet is not without limitations. To validate the effectiveness and applicability of this method, further comparisons with recent research findings and evaluations under different environmental conditions and forest types are necessary. Additionally, studying its computational efficiency is crucial for deployment on resource-constrained UAVs. Future research directions include integrating the FSNet model with actual UAV patrol systems to optimize its performance and stability and further enhance its applicability. Furthermore, improvements to the model can be made to better adapt to different types of forest environments and fire characteristics, making it more versatile and adaptable.

6. Conclusions

In conclusion, this study validates FSNet as a highly effective model for detecting and locating fire spots and smoke in forest-fire images, offering substantial improvements over existing models. The incorporation of YOCO data augmentation, the innovative EBblock attention module, and the L_{forest} loss function within FSNet's architecture are key to its exceptional performance. The comprehensive experimental results, including the mAP scores, Recall rates, and FPS values across three datasets, provide strong evidence of FSNet's potential in practical applications such as drone-based forest-fire surveillance. This work opens the door for the implementation of FSNet in real-world scenarios, enhancing forest-fire management and prevention efforts.

Author Contributions: Conceptualization, D.W. (Donghua Wu); methodology, Z.Q.; software, J.W.; validation, D.W. (Dongyang Wu); formal analysis, D.W. (Dongyang Wu); investigation, D.W. (Donghua Wu); resources, D.W. (Donghua Wu); data curation, J.W.; writing—original draft preparation, D.W. (Donghua Wu); writing—review and editing, D.W. (Donghua Wu); visualization, Z.Q.; supervision, Z.Q.; project administration, Z.Q.; funding acquisition, J.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Postgraduate Research & Practice Innovation Program of Jiangsu Province, grant number KYCX22_1105.

Data Availability Statement: The Flame dataset is available at https://ieee-dataport.org/open-access/flame-dataset-aerial-imagery-pile-burn-detection-using-drones-uavs (accessed on 8 September 2023). The Corsican dataset is available at http://cfdb.univ-corse.fr/index.php?menu=1 (accessed on 10 May 2023). The D-Fire dataset is available at https://github.com/gaiasd/DFireDataset (accessed on 10 May 2023).

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

References

1. Wang, J.; Wang, Y.; Liu, L.; Yin, H.; Ye, N.; Xu, C. Weakly Supervised Forest Fire Segmentation in UAV Imagery Based on Foreground-Aware Pooling and Context-Aware Loss. *Remote Sens.* **2023**, *15*, 3606. [CrossRef]
2. Jin, L.; Yu, Y.; Zhou, J.; Bai, D.; Lin, H.; Zhou, H. SWVR: A Lightweight Deep Learning Algorithm for Forest Fire Detection and Recognition. *Forests* **2024**, *15*, 204. [CrossRef]
3. Abdikan, S.; Bayik, C.; Sekertekin, A.; Bektas Balcik, F.; Karimzadeh, S.; Matsuoka, M.; Balik Sanli, F. Burned Area Detection Using Multi-Sensor SAR, Optical, and Thermal Data in Mediterranean Pine Forest. *Forests* **2022**, *13*, 347. [CrossRef]
4. Qarallah, B.; Othman, Y.A.; Al-Ajlouni, M.; Alheyari, H.A.; Qoqazeh, B.A. Assessment of Small-Extent Forest Fires in Semi-Arid Environment in Jordan Using Sentinel-2 and Landsat Sensors Data. *Forests* **2022**, *14*, 41. [CrossRef]
5. Shin, J.; Seo, W.; Kim, T.; Park, J.; Woo, C. Using UAV Multispectral Images for Classification of Forest Burn Severity—A Case Study of the 2019 Gangneung Forest Fire. *Forests* **2019**, *10*, 1025. [CrossRef]
6. Yuan, C.; Liu, Z.; Zhang, Y. UAV-Based Forest Fire Detection and Tracking Using Image Processing Techniques. In Proceedings of the International Conference on Unmanned Aircraft Systems (ICUAS), Denver, CO, USA, 9–12 June 2015; pp. 639–643.
7. Gao, Y.; Cao, H.; Cai, W.; Zhou, G. Pixel-Level Road Crack Detection in UAV Remote Sensing Images Based on ARD-Unet. *Measurement* **2023**, *219*, 113252. [CrossRef]
8. Zhan, J.; Hu, Y.; Zhou, G.; Wang, Y.; Cai, W.; Li, L. A High-Precision Forest Fire Smoke Detection Approach Based on ARGNet. *Comput. Electron. Agric.* **2022**, *196*, 106874. [CrossRef]
9. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. Ssd: Single Shot Multibox Detector. In Proceedings of the Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Proceedings, Part I 14. Springer: Berlin/Heidelberg, Germany, 2016; pp. 21–37.
10. Fu, C.-Y.; Liu, W.; Ranga, A.; Tyagi, A.; Berg, A.C. Dssd: Deconvolutional Single Shot Detector. *arXiv* **2017**, arXiv:1701.06659.
11. Jiang, P.; Ergu, D.; Liu, F.; Cai, Y.; Ma, B. A Review of Yolo Algorithm Developments. *Procedia Comput. Sci.* **2022**, *199*, 1066–1073. [CrossRef]
12. Bharati, P.; Pramanik, A. Deep Learning Techniques—R-CNN to Mask R-CNN: A Survey. *Comput. Intell. Pattern Recognit. Proc. CIPR 2019* **2020**, 657–668.
13. Girshick, R. Fast R-Cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
14. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-Cnn: Towards Real-Time Object Detection with Region Proposal Networks. *Adv. Neural Inf. Process Syst.* **2015**, *28*, 2969239–2969250. [CrossRef] [PubMed]
15. Han, K.; Wang, Y.; Chen, H.; Chen, X.; Guo, J.; Liu, Z.; Tang, Y.; Xiao, A.; Xu, C.; Xu, Y. A Survey on Vision Transformer. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 87–110. [CrossRef] [PubMed]
16. Liu, S.; Wang, L.; Yue, W. An Efficient Medical Image Classification Network Based on Multi-Branch CNN, Token Grouping Transformer and Mixer MLP. *Appl. Soft Comput.* **2024**, *153*, 111323. [CrossRef]
17. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. 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; pp. 779–788.
18. 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.
19. Redmon, J.; Farhadi, A. Yolov3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
20. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. Yolov4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934.
21. Jocher, G.; Chaurasia, A.; Stoken, A.; Borovec, J.; Kwon, Y.; Michael, K.; Fang, J.; Wong, C.; Yifu, Z.; Montes, D. Ultralytics/Yolov5: V6. 2-Yolov5 Classification Models, Apple M1, Reproducibility, Clearml and Deci. Ai Integrations. *Zenodo* **2022**.

22. Smith, A.K.; Dragicevic, S. An Agent-Based Model to Represent Space-Time Propagation of Forest-Fire Smoke. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2018**, *4*, 207–212. [CrossRef]
23. Sun, H.; Yang, Y.; Tang, X.; Sun, J. A Joint Source Channel Adaptive Communication System Design for the Fire Environment. In Proceedings of the Chinese Automation Congress (CAC), Jinan, China, 20–22 October 2017; pp. 1598–1602.
24. Fonollosa, J.; Solórzano, A.; Marco, S. Chemical Sensor Systems and Associated Algorithms for Fire Detection: A Review. *Sensors* **2018**, *18*, 553. [CrossRef]
25. Ho, C.-C. Machine Vision-Based Real-Time Early Flame and Smoke Detection. *Meas. Sci. Technol.* **2009**, *20*, 045502. [CrossRef]
26. Chen, T.-H.; Yin, Y.-H.; Huang, S.-F.; Ye, Y.-T. The Smoke Detection for Early Fire-Alarming System Base on Video Processing. In Proceedings of the International Conference on Intelligent Information Hiding and Multimedia, Pasadena, CA, USA, 18–20 December 2006; pp. 427–430.
27. Gubbi, J.; Marusic, S.; Palaniswami, M. Smoke Detection in Video Using Wavelets and Support Vector Machines. *Fire Saf. J.* **2009**, *44*, 1110–1115. [CrossRef]
28. Yuan, F.; Xia, X.; Shi, J. Holistic Learning-Based High-Order Feature Descriptor for Smoke Recognition. *Int. J. Wavelets Multiresolut. Inf. Process* **2019**, *17*, 1940005. [CrossRef]
29. Töreyin, B.U.; Dedeoğlu, Y.; Cetin, A.E. Wavelet Based Real-Time Smoke Detection in Video. In Proceedings of the 13th European Signal Processing Conference, Antalya, Turkey, 4–8 September 2005; pp. 1–4.
30. Kinaneva, D.; Hristov, G.; Raychev, J.; Zahariev, P. Early Forest Fire Detection Using Drones and Artificial Intelligence. In Proceedings of the 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia, 20–24 May 2019; pp. 1060–1065.
31. Guan, Z.; Min, F.; He, W.; Fang, W.; Lu, T. Forest Fire Detection via Feature Entropy Guided Neural Network. *Entropy* **2022**, *24*, 128. [CrossRef] [PubMed]
32. Zhang, L.; Lu, C.; Xu, H.; Chen, A.; Li, L.; Zhou, G. MMFNet: Forest Fire Smoke Detection Using Multiscale Convergence Coordinated Pyramid Network with Mixed Attention and Fast-Robust NMS. *IEEE Internet Things J.* **2023**, *10*, 18168–18180. [CrossRef]
33. Yang, H.; Wang, J.; Wang, J. Efficient Detection of Forest Fire Smoke in UAV Aerial Imagery Based on an Improved Yolov5 Model and Transfer Learning. *Remote Sens.* **2023**, *15*, 5527. [CrossRef]
34. Han, J.; Fang, P.; Li, W.; Hong, J.; Armin, M.A.; Reid, I.; Petersson, L.; Li, H. You Only Cut Once: Boosting Data Augmentation with a Single Cut. In Proceedings of the International Conference on Machine Learning, Baltimore, MA, USA, 17–23 July 2022; pp. 8196–8212.
35. Zhao, C.; Sun, Y.; Wang, W.; Chen, Q.; Ding, E.; Yang, Y.; Wang, J. MS-DETR: Efficient DETR Training with Mixed Supervision. *arXiv* **2024**, arXiv:2401.03989.
36. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. Yolox: Exceeding Yolo Series in 2021. *arXiv* **2021**, arXiv:2107.08430.
37. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-End Object Detection with Transformers. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 213–229.
38. 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.
39. Sayad, Y.O.; Mousannif, H.; Al Moatassime, H. Predictive Modeling of Wildfires: A New Dataset and Machine Learning Approach. *Fire Saf. J.* **2019**, *104*, 130–146. [CrossRef]
40. Santos, S.M.B.d.; Bento-Gonçalves, A.; Vieira, A. Research on Wildfires and Remote Sensing in the Last Three Decades: A Bibliometric Analysis. *Forests* **2021**, *12*, 604. [CrossRef]
41. Xu, R.; Lin, H.; Lu, K.; Cao, L.; Liu, Y. A Forest Fire Detection System Based on Ensemble Learning. *Forests* **2021**, *12*, 217. [CrossRef]
42. Rahman, E.U.; Khan, M.A.; Algarni, F.; Zhang, Y.; Irfan Uddin, M.; Ullah, I.; Ahmad, H.I. Computer Vision-Based Wildfire Smoke Detection Using UAVs. *Math. Probl. Eng.* **2021**, *2021*, 9977939. [CrossRef]
43. Wu, L.; Chen, L.; Hao, X. Multi-Sensor Data Fusion Algorithm for Indoor Fire Early Warning Based on BP Neural Network. *Information* **2021**, *12*, 59. [CrossRef]
44. Pérez-Porras, F.-J.; Triviño-Tarradas, P.; Cima-Rodríguez, C.; Meroño-de-Larriva, J.-E.; García-Ferrer, A.; Mesas-Carrascosa, F.-J. Machine Learning Methods and Synthetic Data Generation to Predict Large Wildfires. *Sensors* **2021**, *21*, 3694. [CrossRef] [PubMed]