machine learning & knowledge extraction

MDPI

Article

Research on Forest Fire Detection Algorithm Based on Improved YOLOv5

Jianfeng Li [1,*] and Xiaoqin Lian [1,2]

[1] School of Artificial Intelligence, Beijing Technology and Business University, Beijing 100048, China; lianxq@th.btbu.edu.cn
[2] China Light Industry Key Laboratory of Industrial Internet and Big Data, Beijing Technology and Business University, Beijing 100048, China
* Correspondence: lijf@btbu.edu.cn; Tel.: +86-010-81353323

Abstract: Forest fires are one of the world's deadliest natural disasters. Early detection of forest fires can help minimize the damage to ecosystems and forest life. In this paper, we propose an improved fire detection method YOLOv5-IFFDM for YOLOv5. Firstly, the fire and smoke detection accuracy and the network perception accuracy of small targets are improved by adding an attention mechanism to the backbone network. Secondly, the loss function is improved and the SoftPool pyramid pooling structure is used to improve the regression accuracy and detection performance of the model and the robustness of the model. In addition, a random mosaic augmentation technique is used to enhance the data to increase the generalization ability of the model, and re-clustering of flame and smoke detection a priori frames are used to improve the accuracy and speed. Finally, the parameters of the convolutional and normalization layers of the trained model are homogeneously merged to further reduce the model processing load and to improve the detection speed. Experimental results on self-built forest-fire and smoke datasets show that this algorithm has high detection accuracy and fast detection speed, with average accuracy of fire up to 90.5% and smoke up to 84.3%, and detection speed up to 75 FPS (frames per second transmission), which can meet the requirements of real-time and efficient fire detection.

Keywords: forest fire detection; attention mechanism; staged object detection; deep learning

Citation: Li, J.; Lian, X. Research on Forest Fire Detection Algorithm Based on Improved YOLOv5. *Mach. Learn. Knowl. Extr.* **2023**, *5*, 725–745. https://doi.org/10.3390/make5030039

Academic Editors: Guoqing Chao and Xianzhi Wang

Received: 19 May 2023
Revised: 26 June 2023
Accepted: 27 June 2023
Published: 28 June 2023

1. Introduction

Fire has always been a major threat and disaster throughout the world, and early prevention and rapid detection of fire are the most important methods of reducing the serious harm caused by the occurrence and spread of fires [1,2], which is why it is particularly important to be able to warn of fire in a timely and accurate manner. It is well known that forests, grasslands, and wild slopes serve as nature's best "seasoning" system, and they are gaining increased attention due to the benefits they provide for the natural carbon and water cycles as well as for maintaining the balance of an ecosystem and improving its ecology. Although China has a vast land area, the proportion of stored forest, grassland and wild slope resources is tiny. For example, the per capita forest area is less than a quarter of the world average [1,3]. Fires are extremely destructive, not only burning large areas of trees and causing species extinction but also causing soil erosion and threatening people's lives and property [4]. Therefore, inspection technology for fire has received more and more attention from scholars and related departments. Early detection of fires and an accurate grasp of the fire environment and posture allows the command center to take effective suppression measures for fire control and extinguishment. The task of fire detection can be divided into the task of detecting both fire and smoke targets [4]. Fire detection is initially performed using the response values of temperature and smoke

detectors as the detection results. However, the detection of temperature and smoke detectors has a certain lag [5], which is insufficient to fulfill the purpose of early fire prediction, is not resilient enough to environmental changes, and is susceptible to false alarms caused by electromagnetic interference [6].

Computer-vision-based fire detection has made significant progress in recent years [7–9], allowing the detection of both flame targets and smoke targets in images rather than just a single target. In early computer-vision-based fire detection methods, the task is decomposed into two parts, flame detection and smoke detection, using static appearance information, such as color [10], texture [11], and shape [12]. Flames and smoke are exhibited during a fire, and motion information of flames and smoke in time order [13–16] is used to construct discriminative features of flames or smoke targets, and these discriminative features are then used to detect these targets. Most original flame and smoke detection methods rely on hand-crafted features, achieving good results on early single-category datasets with small amounts of data. However, in realistic scenes flames and smoke exhibit features such as color, texture, and shape that are unstable and have some variability, making hand-designed features more arduous [17]. In addition, hand-crafted features rely mainly on a priori information about the target and do not have high abstraction and invariance; thus, their detection accuracy is limited. In addition, most flame and smoke detection methods are proposed for single fire types or fixed scenes, which are not robust, have significantly lower detection accuracy when lighting, scenes, and fire types change, and cannot meet the needs of practical scenarios. There is still a high rate of missed detection for tunnels, forests, dim light, long distances, and small targets, and the performance needs to be further improved to cope with complex real-world scenarios.

FU Tian-ju et al. [18] designed a 12-layer convolutional neural network for forest fire detection that is pre-trained with the ImageNet dataset before training and testing it with a self-built dataset (500 images for training and 100 images for testing). During the training process, a dropout operation is performed on the hidden layer of the network to reduce the probability of overfitting. However, its self-built dataset has only 600 images, and it is still possible to overfit when using iterative training. Moreover, the scenarios in the dataset are single and its robustness is not good in generic scenarios. Additionally, its final output is simply the probability of fire and no fire, and does not contain other information about fire and smoke targets. Frizzi et al. [19] proposed that the classical convolutional neural network combined with convolution and maximum pooling can be used to determine whether the image contains flame or smoke. The algorithm improves the speed of detection by moving a 12 × 12 sliding window over the feature map. Compared with the method of Zhang et al., it contains more datasets, covers a wider range of scenarios and has a lighter and faster network. However, it only includes three categories (flame, smoke, and normal), and it is not well-adapted to scenes with both smoke and fire. Similarly, the final output of this method does not contain the location information of the flame and smoke target, leading to an inaccurate perception of the fire state. Through global and local fire detection of the pictures, the method proposed by Sandler et al. [20] further improves classification precision, but it only uses fire detection results as a basis for fire classification, and cannot identify smoke, leading to a poor performance when smoke is obscuring the flames. Furthermore, it is just a categorization and does not contain the location information of flame and smoke targets.

2. Related Work

2.1. Target Detection Algorithms

The deep learning approach can extract target features from a large number of images and obtain generalized information with better learning ability and adaptability. Target detection is one of the main branches of computer vision based on deep learning, and YOLO, proposed by Redmon et al. [21], is an outstanding example of a target detection algorithm. Using a convolutional neural network, the method extracts and classifies the

features of the input image. The method frames the target on the original image and finds the categories with a good recognition performance but slightly poor accuracy. In order to improve accuracy, Redmon et al. [22] proposed YOLOv2 algorithm, which uses K-means to cluster prior boxes. It improves detection accuracy by splitting the prior frames into three size categories, each of which is further subdivided into three categories, corresponding to three sizes of targets, large and small, respectively. For small targets, however, this algorithm performs poorly. Redmon et al. [23] then proposed the YOLOv3 algorithm. Based on YOLOv2, feature pyramid networks (FPN) [24] were used to improve detection performance by fusing features of different sizes. Detection performance was significantly improved in small target detection. Wang et al. [25] proposed a DSE-YOLO model with good results based on YOLOv3 for strawberry detection. However, the detection of YOLOv3 is not satisfactory for targets with complex features. Bochkovskiy et al. [26] proposed YOLOv4 based on the optimization of the YOLO series in various aspects and achieved better performance indexes in terms of applications. Jocher et al. [27] proposed YOLOv5 based on YOLOv4 with a modified loss function, including a focus structure, adaptive anchor frame calculation at the input, and other optimization methods. Compared with the previous YOLO model, it has a lighter structure and more accurate precision. However, since it uses an anchor frame as the initial preset frame, the computational cost will increase, affecting the model performance. Although YOLOv5 has a relatively good performance in target detection, it needs to be improved in the area of forest fire detection. Part of the fire area is very small and needs to be detected quickly, making it more important to ensure accuracy and real-time forest fire detection. This study proposes a fire detection method, YOLOv5-IFFDM, to address the shortcomings of YOLOv5, and confirms its feasibility for detecting forest fires by experiments.

2.2. Methods of This Paper

In order to detect both fire and smoke in fire detection tasks and to ensure the effectiveness of small target detection, the following contributions are made in this paper. (1) Darknet-53 of YOLOv5 is used as the backbone feature extraction network, and the attention mechanism is added to the backbone. The flame and smoke detection accuracy and the perception accuracy of small targets are improved. (2) An improved function is adopted to enhance the loss and gradient weights of high IoU, thus improving the regression accuracy and detection of the model. (3) The model adopts a spatial pyramidal pooling structure and replaces all MaxPool pooling layers with more efficient SoftPool to improve the robustness of the model to spatial layout and object variability. (4) The model is augmented using the stochastic mosaic augmentation technique to enhance the data and the model generalization. (5) The parameters of the convolutional and normalization layers of a trained model are merged to reduce model complexity, improve detection speed, and reduce hardware requirements. (6) We use the K-means algorithm for fire and smoke a priori frame clustering because the a priori frame is often relevant to the detection task, and the appropriate size of the a priori frame improves the accuracy and speed of detection. The overall block diagram of the model is shown in Figure 1.

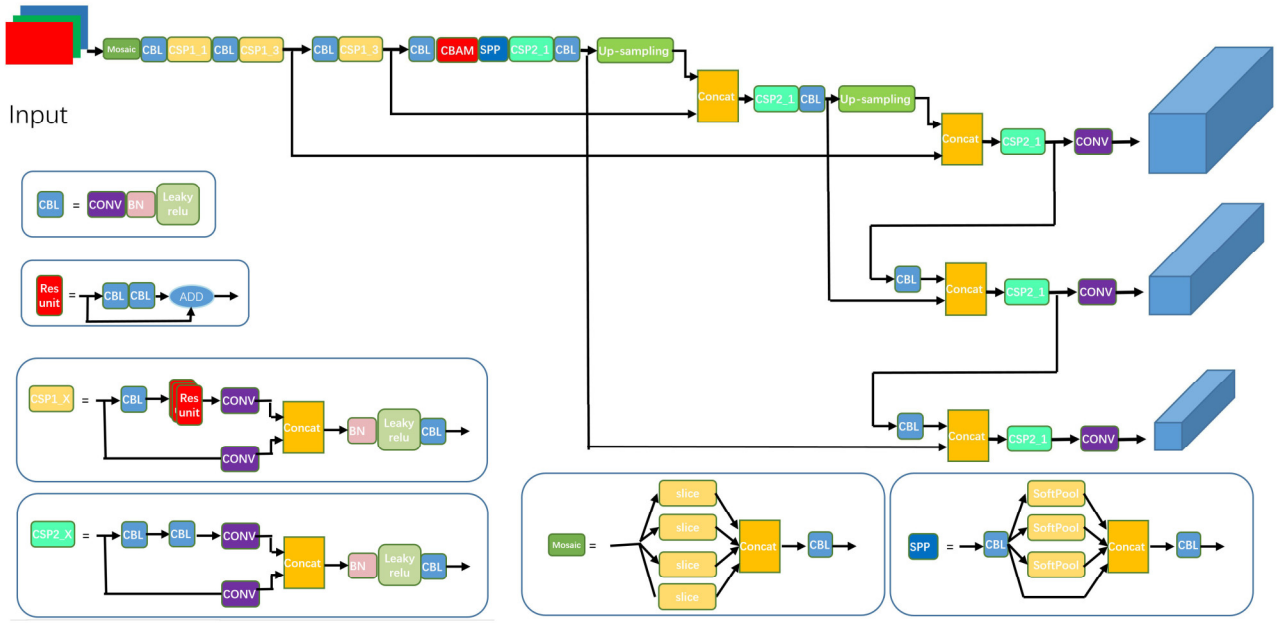


Figure 1. The overall block diagram of the model.

3. Materials and Algorithms

3.1. Feature-Strengthening Structure Based on Attention Mechanism

Attentional mechanisms focus on obtaining local information, which works by imitating the way humans pay attention. It devotes more attentional resources to the target region to obtain more target information and suppresses background information. The main ones are squeeze-and-excitation networks (SE), proposed by Hu et al. [28], and the convolutional block attention module (CBAM) [29], which was advanced by Woo et al. The mechanism used in this paper is CBAM, which is a combination of channel and spatial attention mechanisms. It obtains more information about the channel where the target is located by performing an attention operation on the channel; it performs an attention operation on the space to ensure that the target location information is better obtained and improves the accuracy of target detection. The CBAM attention mechanism is shown in Figure 2.

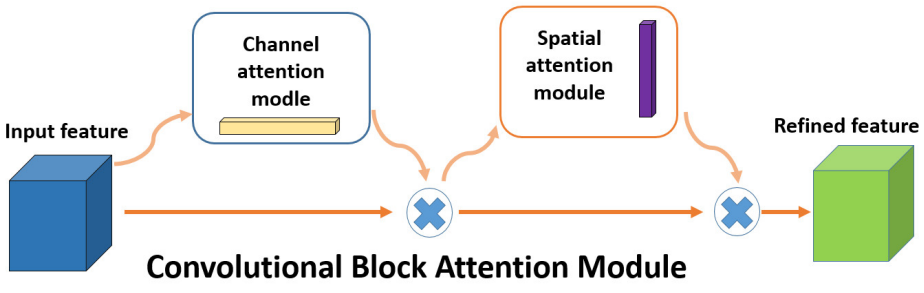


Figure 2. Convolutional Block Attention Module.

It has been shown by Foggia et al. [12] that the feature information of fire differs on different channels, as well as the location information on each feature map. Therefore, CBAM, which combines channel attention and spatial attention mechanisms, can be used to process them, improve the interest of the model for the target, and enhance the effectiveness of acquiring feature targets.

3.2. Improved α-IoU

The CIoU loss function used in YOLOv5 is an improvement of the DIoU loss function. Because the DIoU increases the height and width loss of the prior frame, the prediction accuracy is higher. The Equations (1)–(4) are as follows.

$$\mathrm{CIoU}=\mathrm{IoU}-\frac{\rho^2(b,b^{gt})}{c^2}-\beta U, \tag{1}$$

$$U=\frac{4}{\pi}(\arctan\frac{w^{gt}}{h^{gt}}-\arctan\frac{w}{h})^2, \tag{2}$$

$$\beta=\frac{U}{(1-\mathrm{IoU})+U}, \tag{3}$$

$$L_{\mathrm{CIoU}}=1-\mathrm{CIoU}, \tag{4}$$

The three components in CIoU correspond to the calculation of the IoU, centroid distance, aspect ratio β, and aspect ratio, and the calculation procedure is shown above. Here, w, h and wgt, hgt denote the height and width of the predicted and real frames, respectively. α-IoU increases the loss and gradient weights of high IoU by performing a power operation on the IoU and its penalty term expression, thus improving the regression accuracy of the model. Its Equation (5) is as follows.

$$L_{\alpha\text{-CIoU}}=1-\mathrm{IOU}^2+\frac{\rho^{2\alpha}(b,b^{gt})}{c^{2\alpha}}-(\beta U)^{\alpha}, \tag{5}$$

In this paper, we define the value of α as 3. The power operation focuses more on the high IOU target, which not only enhances the accuracy of the regression but also accelerates the convergence of the network. Therefore, we use the α-IoU loss function for boundary regression in this paper.

3.3. Improved Spatial Pyramidal Pooling Structure

Spatial pyramid pooling is a multi-scale feature fusion pooling method that can preserve object features well [30,31], maintain feature map shape, and output fixed-size features with any feature image size as input. The pooling operation, as one of the most basic algorithms for image processing in the field of deep learning, can reduce the size of the feature map of the model by retaining some features while reducing the computational cost, preventing overfitting, and improving the model's generalization ability. In CNN, the common pooling methods are average pooling and maximum pooling. Average pooling is an averaging operation on the neighborhood feature points, which can preserve the background information well and make the image smoother. However, it will cause feature information loss. Maximum pooling obtains the maximum value of neighboring feature points, which can extract obvious texture feature information well. Thus, it is more suitable for a convolutional neural network to retain prominent features and speed up the model response. It is also easy to ignore some detailed feature information. SoftPool is a variant of a pooling structure with the original pooling layer function while retaining more feature information. The SoftPool method can reduce the risk of losing detailed features during pooling and has good feature retention for small targets. Thus, we propose using SoftPool instead of MaxPool in spatial pyramid pooling to better preserve the detailed features of the fire and enhance its detection [32–34]. The calculation process of SoftPool is shown in Figure 3.

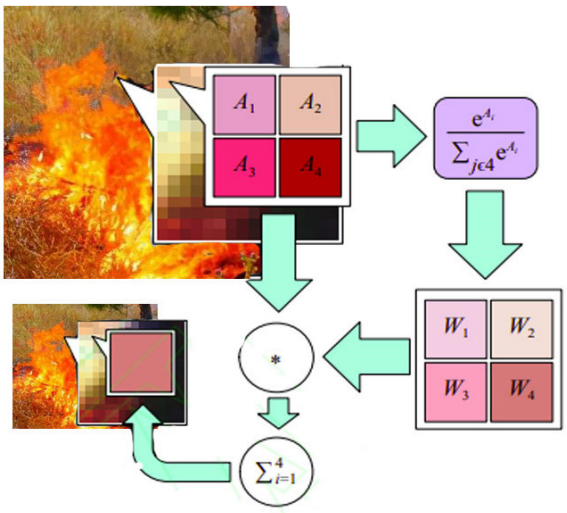


Figure 3. SoftPool calculation flowchart.

The local feature region is defined as m. R is a pooling kernel of size $k \times k$, and the dimension is denoted as $C \times H \times W$, where C denotes the number of channels, H denotes the height of the feature map, and W denotes the width of the feature map. The corresponding feature weights are calculated nonlinearly based on the values of each feature point, and Equation (6) is shown as follows.

$$\omega_i = \frac{e^{a_i}}{\sum_{j \in R} e^{a_j}}, \tag{6}$$

where ω_i denotes the weight, a_i represents the eigenvalue of a point, and e^{a_i} denotes the activation value. Calculating the weights ensures that the feature texture information can be passed and activates the feature values in the region m to be passed backwards. After obtaining the weights, they are summed with the region's feature values m to obtain the output results, as shown in Equation (7).

$$\tilde{a} = \sum_{i \in R} \omega_i \bullet a_i, \tag{7}$$

where $\tilde{a}$ denotes the output value of feature points after SoftPool, obtained by the standard summation of all weighted activations in the kernel neighborhood R, as shown in Figure 4.

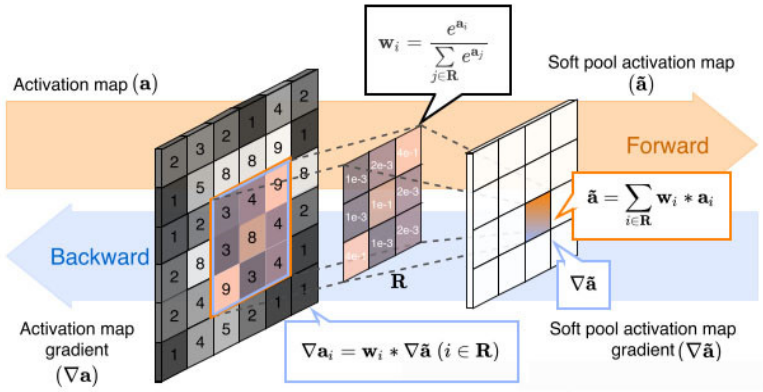


Figure 4. SoftPool transfer process diagram.

3.4. Mosaic Random Data Enhancement Method

Mosaic enhancement technology takes 4 or 6 images, first zooms, pans, flips, performs color gamut transformation, etc., and then performs the stitching operation. Each image has a corresponding target frame. After stitching by the mosaic method, a picture containing 4 or 6 target frames is obtained, which not only greatly enriches the environment in which the target appears but also implicitly increases the pre-trained batch. This is helpful for improving detection accuracy. The implementation process is shown in Figure 5.

(**a**) (**b**)

Figure 5. Mosaic random image processing. (**a**) Processed images; (**b**) Stitched image.

3.5. Merge BN Layers to Convolutional Layers to Improve Network Detection Speed

In the training of deep network models, the BN (Batch Normalization) layer accelerates the convergence of the network and prevents overfitting. It is usually placed after the convolutional layer. The BN layer normalizes the data and can effectively solve the gradient disappearance and gradient explosion problems. Although the BN layer plays a positive role in training, the extra layers of operations in the network's forward inference affect the model's performance and take up more memory or video memory space. BN (Batch Normalization) layer parameters are merged into the convolutional layer to improve model forward inference speed. Therefore, after training, it is merged with the convolutional layer to improve the speed of forward inference.

The principle of merging is as follows.

Given batch data $x_1, x_2, \cdots, x_n$, we perform the normalization operation on them, as shown in Equation (8).

$$\hat{x}_i = \gamma \frac{x_i - \mu}{\sqrt{\sigma^2 + \theta}} + \beta, \tag{8}$$

where μ is the data mean and σ is the data variance. Mean and variance are calculated separately for each channel. γ and β are the parameters that can be learned for each different batch of data: γ is the scaling factor, and β is the translation factor. θ is a very small number, approximately 10^{-6}, which is intended to prevent the denominator from being zero. The above Equation (8) can be transformed as follows Equation (9).

$$\hat{x}_i = \frac{\gamma x_i}{\sqrt{\sigma^2 + \theta}} + \beta - \frac{\gamma \mu}{\sqrt{\sigma^2 + \theta}}, \tag{9}$$

In order to combine BN and convolution, following the normalization Equation (9) above, the image feature map normalization matrix can be written as Equation(10).

$$\begin{pmatrix} \widehat{F}_{1,i,j} \\ \widehat{F}_{2,i,j} \\ \vdots \\ \widehat{F}_{C-1,i,j} \\ \widehat{F}_{C,i,j} \end{pmatrix} = \begin{pmatrix} \frac{\gamma_1}{\sqrt{\widehat{\sigma}_1^{\,2}+\theta}} & 0 & \cdots & & 0 \\ 0 & \frac{\gamma_2}{\sqrt{\widehat{\sigma}_2^{\,2}+\theta}} & & & \\ \vdots & & \ddots & & \\ & & & \frac{\gamma_{C-1}}{\sqrt{\widehat{\sigma}_{C-1}^{\,2}+\theta}} & 0 \\ 0 & 0 & \cdots & 0 & \frac{\gamma_C}{\sqrt{\widehat{\sigma}_C^{\,2}+\theta}} \end{pmatrix} \bullet \begin{pmatrix} F_{1,i,j} \\ F_{2,i,j} \\ \vdots \\ F_{C-1,i,j} \\ F_{C,i,j} \end{pmatrix} + \begin{pmatrix} \beta_1 - \gamma_1 \frac{\widehat{\mu}_1}{\widehat{\sigma}_1^{\,2}+\theta} \\ \beta_2 - \gamma_2 \frac{\widehat{\mu}_2}{\widehat{\sigma}_2^{\,2}+\theta} \\ \vdots \\ \beta_{C-1} - \gamma_{C-1} \frac{\widehat{\mu}_{C-1}}{\widehat{\sigma}_{C-1}^{\,2}+\theta} \\ \beta_C - \gamma_C \frac{\widehat{\mu}_C}{\widehat{\sigma}_C^{\,2}+\theta} \end{pmatrix}, \tag{10}$$

The feature map F, after normalization, is also the result equivalent to a 1 × 1 × C convolution.

Write

$$\mathbf{W}_{BN} = \begin{pmatrix} \frac{\gamma_1}{\sqrt{\widehat{\sigma}_1^{\,2}+\theta}} & 0 & \cdots & & 0 \\ 0 & \frac{\gamma_2}{\sqrt{\widehat{\sigma}_2^{\,2}+\theta}} & & & \\ \vdots & & \ddots & & \\ & & & \frac{\gamma_{C-1}}{\sqrt{\widehat{\sigma}_{C-1}^{\,2}+\theta}} & 0 \\ 0 & 0 & \cdots & 0 & \frac{\gamma_C}{\sqrt{\widehat{\sigma}_C^{\,2}+\theta}} \end{pmatrix}, \tag{11}$$

$$\mathbf{b}_{BN} = \begin{pmatrix} \beta_1 - \gamma_1 \frac{\widehat{\mu}_1}{\widehat{\sigma}_1^{\,2}+\theta} \\ \beta_2 - \gamma_2 \frac{\widehat{\mu}_2}{\widehat{\sigma}_2^{\,2}+\theta} \\ \vdots \\ \beta_{C-1} - \gamma_{C-1} \frac{\widehat{\mu}_{C-1}}{\widehat{\sigma}_{C-1}^{\,2}+\theta} \\ \beta_C - \gamma_C \frac{\widehat{\mu}_C}{\widehat{\sigma}_C^{\,2}+\theta} \end{pmatrix}, \tag{12}$$

$\mathbf{W}_{BN}$ is the matrix of C × C and $\mathbf{b}_{BN}$ is the matrix of C × 1. Therefore, the BN process can be written as Equation (13):

$$\widehat{\mathbf{F}} = \mathbf{W}_{BN}\mathbf{F} + \mathbf{b}_{BN}, \tag{13}$$

In addition, $\mathbf{F} = conv(\mathbf{X}) = \mathbf{W}_{conv} \bullet \mathbf{X} + \mathbf{b}_{conv}$

So:

$$\hat{\mathbf{F}} = \mathbf{W}_{BN}(\mathbf{W}_{conv} \bullet \mathbf{X} + \mathbf{b}_{conv}) + \mathbf{b}_{BN} = \mathbf{W}_{BN}\mathbf{W}_{conv} \bullet \mathbf{X} + \mathbf{W}_{BN}\mathbf{b}_{conv} + \mathbf{b}_{BN}, \quad (14)$$

Therefore $\mathbf{W}_{new} = \mathbf{W}_{BN}\mathbf{W}_{conv}$, $\mathbf{b}_{new} = \mathbf{W}_{BN}\mathbf{b}_{conv} + \mathbf{b}_{BN}$, and the fusion of the convolutional and BN layers is achieved.

3.6. Re-Clustering Anchor

The YOLOv5 network still uses the anchor boxes mechanism to preset three initial boxes with different areas and aspect ratio sizes for each feature point. According to the K-means clustering algorithm, these are the a priori boxes. For the target detection algorithm, a suitable set of prior frames will reduce the tuning of the network to obtain the prediction frames faster. Flame and smoke detection can only be achieved with high accuracy and speed when the a priori frame of YOLOv5 is optimized to fit the dataset and scenarios of the detection task [35–39]. In this paper, we decided to analyze the true boxes of labeled flames and smoke by an improved K-means clustering algorithm so that the dimensions of the a priori boxes better match the dimensions of the forest flame and smoke dataset used in this paper. The number of clusters K is the number of a priori boxes to be selected, and the width and height of the central box of the clusters give the width and height of the a priori boxes.

4. Experiment

4.1. Experimental Environment

To implement the training and testing of the proposed method and other target detection methods in this paper, an 11th Gen Intel(R) Core(TM) i5-11260H@ 2.60 GHz CPU with 8 GB of RAM is used as the experimental device and an NVIDIA GeForce RTX 3090 24G GPU as the graphics device. The training parameters are as follows: the Adam optimizer is used, and a total of 100 iteration cycles are set. We performed 1–50 cycles of freeze training on the backbone network with an initial learning rate of 0.001, weight decay of 0.0005, and batches of 8. 51–100 cycles were trained for thawing, with an initial learning rate of 0.0001, a weight decay of 0.0005, and a batch size of 2. The experiment conditions in this paper are shown in Table 1.

Table 1. Experimental conditions.

Experimental Environment	Details
Operating System	Windows 10
CPU	11th Gen Intel(R) Core(TM) i5-11260H@ 2.60 GHz
Deep Learning Framework	PyTorch 1.12.1
Programming Language	Python 3.9
GPU	NVIDIA GeForce RTX 3090
CUDA Version	CUDA 10.2
Initial learning rate	0.001
Epochs	200
batch-size	8
Img-size	640 × 640

4.2. Dataset

The dataset used is the fire and smoke dataset (http://www.yongxu.org/databases.html, accessed on 18 February 2023) as well as images downloaded from the web and taken in reality. The acquired images are annotated by the dataset annotation tool LabelMe. The VOC2007 dataset format is used as the dataset format. The dataset includes a total of 13,688 images of flames and smoke from various scenes, some of which are shown in Figure 6. Details of the dataset are shown in Table 2.

(**a**)

(**b**)

(**c**)

(**d**)

Figure 6. Partial images of the dataset. (**a**,**b**) Typical forest fire and (**c**,**d**) forest fire from the UAV view. (**a**) Typical canopy forest fire; (**b**) Typical surface forest fire; (c) Canopy Forest fire from UAV view; (**d**) Surface forest fire from the UAV view.

Table 2. Details of dataset.

Dataset	Train	Val	Test
forest	11,090	1238	1360
forest fire	6892	679	978
forest smoke	873	128	212
forest fire and smoke	3325	431	170

Dataset Widening

a) Flip to Increase the Width

Image flip simulation can be used to obtain images collected from different flight directions due to the limited dataset material. In this way, we use image flipping to obtain the image as captured when flying from different directions, thus achieving the purpose of data widening [40]. The image flip effect is shown in Figure 7.

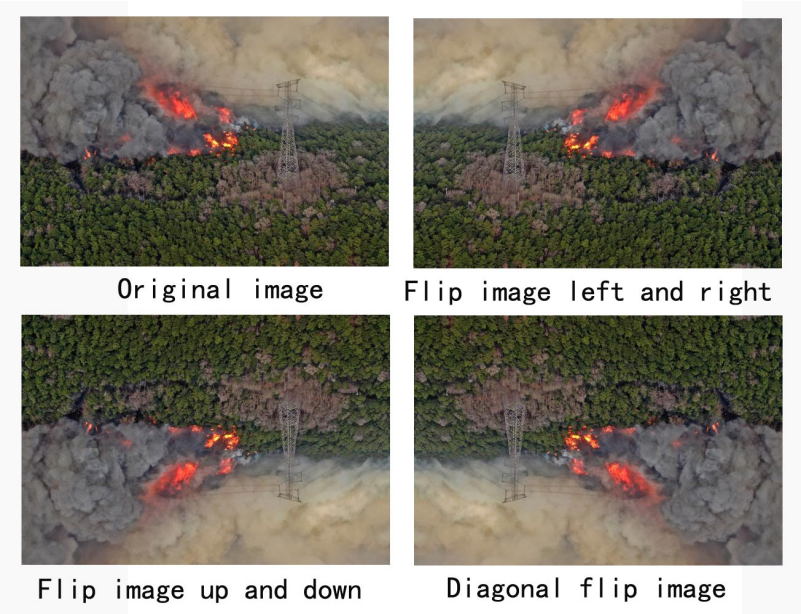


Figure 7. Flip effect example.

b) Mixing and Widening

Since the acquired image scenes are limited, to enrich the scenes, the images are augmented by the method of blending backgrounds. Firstly, a forest image is randomly selected as the background. Secondly, a fire image is obtained by keying out the flames on the image and then fusing the flames with the background image using an image fusion technique [41].

c) Image Mosaic Stitching Enhancement Technology

When processing the dataset with the mosaic stitching enhancement technique, 4 or 6 images are randomly selected to be stitched together into one image. By using this method, the pre-trained batch is implicitly increased, and the complexity of the image is increased, which affects the accuracy of image detection.

In addition, the dataset was partitioned in the ratio of 9:1 for training and testing, respectively. The training dataset part is further divided into training and validation sets with a ratio of 9:1 to prevent the occurrence of overfitting during the training process.

4.3. Evaluation Indicators

To evaluate the YOLOv5-IFFDM network as well as other network performance, we use the following performance metrics as comprehensive evaluation metrics [42]: Precision, recall, F1 score, and mean average precision (mAP). We use size to measure the model's size and FPS to measure the model in real time. Some of its Formulas (15)–(21) are defined as follows.

$$P = \frac{TP}{TP + FP}, \tag{15}$$

$$R = \frac{TP}{TP + FN}, \tag{16}$$

$$F1 = 2 \times \frac{P \times R}{P + R}, \tag{17}$$

$$AP_i = \int_0^i P_i(R_i)dR_i\,, \tag{18}$$

$$mAP = \frac{1}{n}\sum_{i=1}^{n} AP_i\,, \tag{19}$$

$$Time = Pre-process + Inference + NMS\,, \tag{20}$$

$$FPS = \frac{1}{Time}\,, \tag{21}$$

where TP denotes true positive samples, FP denotes false positive samples, and FN denotes false negative samples. The $F1$ score is the summed average of the precision and recall rates. The higher the *F1* score, the higher the precision and recall. AP indicates the performance of each target object class. The mAP metric is the average of the mean accuracy, which is used to measure the overall detection accuracy of the target detection algorithm model. *Time* represents the time spent for the whole training, consisting of three parts. FPS indicates how many forest fire images can be processed in one second.

Taking forest fire detection as an example, *TP* indicates how many forest fire images are correctly predicted, i.e., the detection object and the model detection result are both forest fires. *FP* means the detection object is a non-forest fire, and the model detection result is a non-forest fire. *FN* indicates that the detection object is a forest fire, and the model detection result is a non-forest fire. *NMS*, also known as non-maximum suppression, is the post-frame processing time.

The experimental procedure is as follows. First, the original YOLOv5 detection model is trained using a self-built forest fire classification dataset and evaluated using a test set. The CBAM attention mechanism is then added for experimental comparison. Again, the MaxPool layer in YOLOv5 was replaced with SoftPool for evaluation. The fourth experiment is to modify the loss function to α-IoU based on experiment III. The fifth experiment is to add mosaic data enhancement for experimental comparison based on experiment IV. Following is a table that shows the experimental test results for the sixth and seventh experiments, which add the frame re-clustering and convolutional layer merging again in turn. The experimental results are shown in Table 3.

Table 3. Model experimental results.

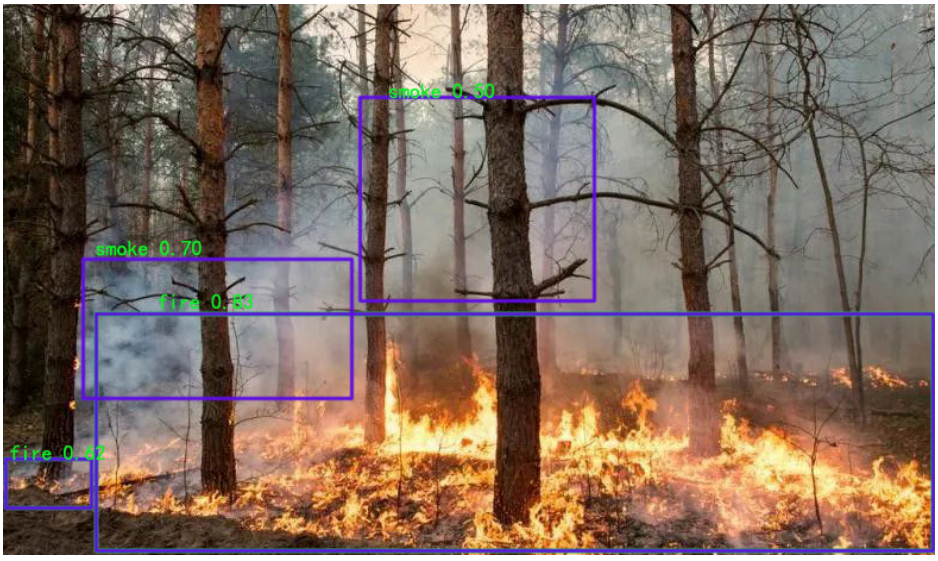
Model	Forest Fire	Forest Smoke	FPS	Time
YOLOv5	0.820	0.790	59	16.9
YOLOv5 + CBAM	0.852	0.805	62	16.1
YOLOv5 + CBAM + SoftPool	0.870	0.823	63	15.9
YOLOv5 + CBAM + SoftPool + α-IoU	0.885	0.822	63	15.9
YOLOv5 + CBAM + SoftPool + α-IoU + Mosaic	0.896	0.835	63	15.9
YOLOv5 + CBAM + SoftPool + α-IoU + Mosaic + Re-clustering class anchor	0.906	0.842	62	16.1
YOLOv5 + CBAM + SoftPool + α-IoU + Mosaic + Re-clustering class anchor + Convolution, BN merge (YOLOv5-IFFDM, ours)	0.905	0.843	75	13.3

4.4. Detection Performance Analysis

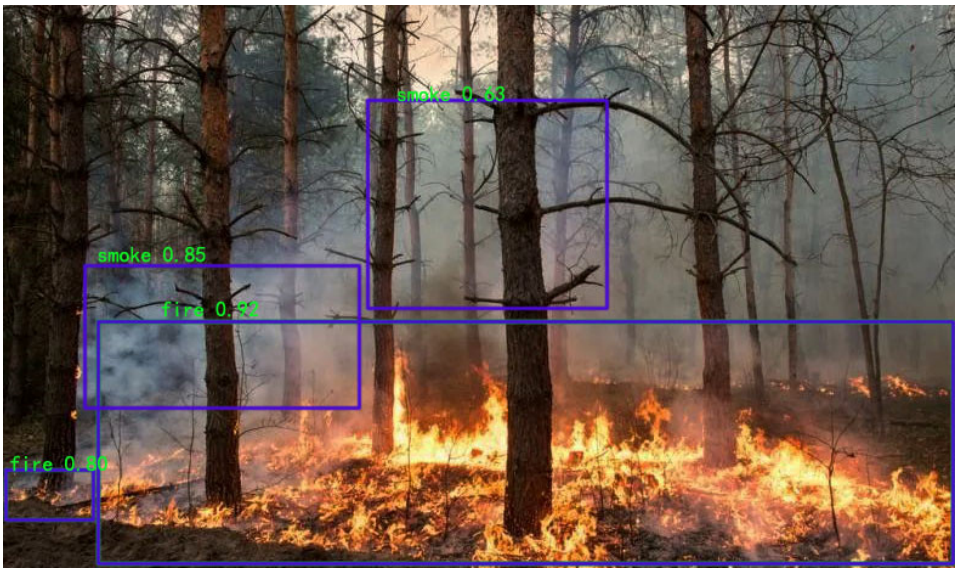
From the above results in Table 3, we found that YOLOv5, one of the more advanced single-stage target detection models, has a good mAP@0.5 for forest fire classification and recognition, but there is still room for improvement. YOLOv5 has a detection accuracy of 0.820 and 0.729 for forest fire and forest smoke, respectively, which is slightly lower, but there is still room for improvement. In addition, the FPS value of YOLOv5 is 59, and the detection time is 16.9 ms, which is slow in real-time detection. In experiment II, we added the attention mechanism CBAM in YOLOv5, and both forest fire and forest smoke were improved by 3.2% and 1.5%, respectively, showing the effectiveness of the CBAM attention mechanism in the detection process.

Experiment III further improves detection accuracy by replacing the spatial pyramidal MaxPool pooling structure with SoftPool. In experiment IV, the loss function in YOLOv5 is changed to α-IoU, the forest fire accuracy is improved, and the forest smoke detection accuracy is basically unchanged. The detection accuracy of forest fires and forest smoke was improved after adding dataset mosaic preprocessing and re-clustering anchors to experiments V and VI. The merging of the convolution and BN layers was performed in experiment VII, and the detection accuracy did not improve, but the detection speed increased substantially, indicating that the prediction speed is indeed affected when the convolution and BN layers are separated, as expected. The model's accuracy in this paper is 90.5% and 84.3% for forest fire and forest smoke, respectively. Compared with the YOLOv5 model, the mAP of this model is improved by 8.5% and 5.3%, respectively, suggesting that the model has better results in forest fire classification detection.

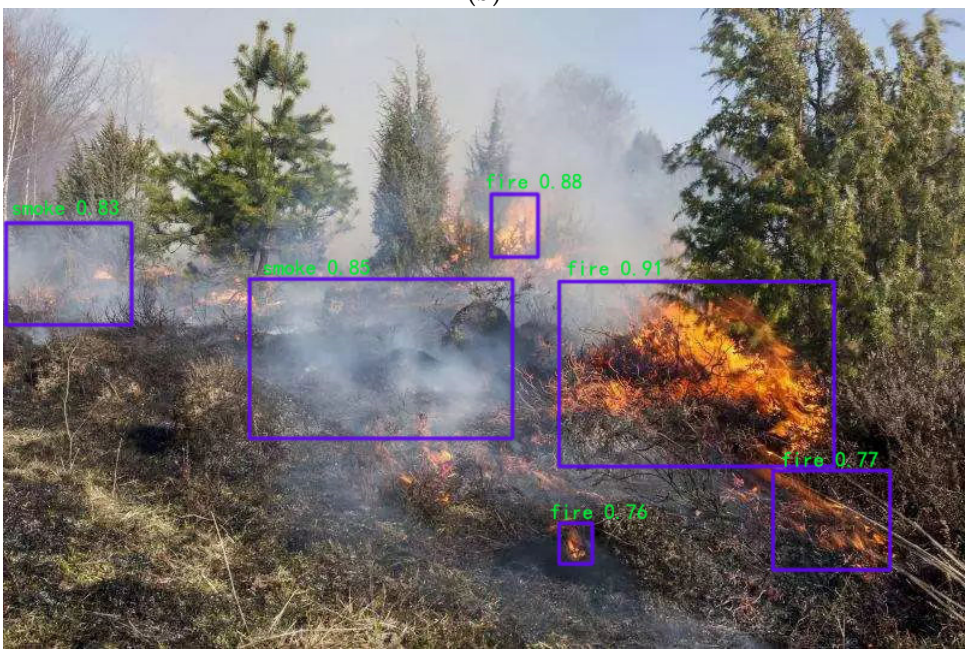
The results of the YOLOv5 model and the YOLOv5-IFFDM forest fire classification and detection model are shown in Figures 8 and 9. Figure 8 shows the detection of a sample of forest fire training images with typicality, and Figure 9 shows the detection of forest fire image samples from the UAV camera view. From the detection result, the YOLOv5 model detects that the rectangular box of the forest fire is not located in the same place as the real box. As a result, the missing detection problem will occur, and the detection performance will be poor. YOLOv5-IFFDM model detection aligns more with the real forest fire target frame. The model has better detection of forest fire types and fewer false detections of leakage. Experiments show that the model YOLOv5-IFFDM proposed in this paper is more suitable for forest fire classification detection.



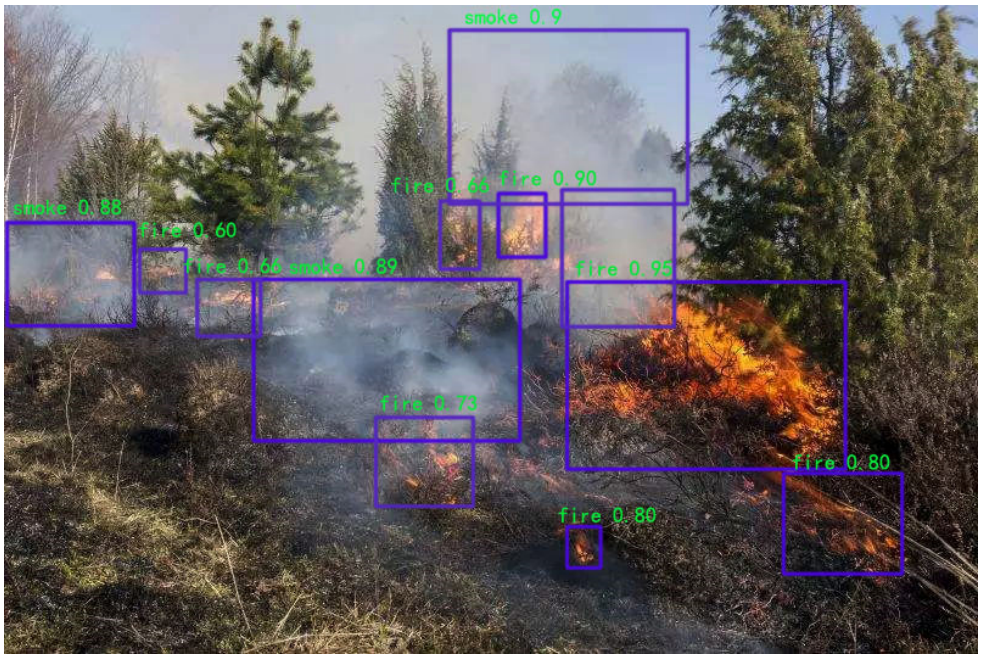
(**a**)


smoke 0.63
smoke 0.85
fire 0.72
fire 0.80

(**b**)

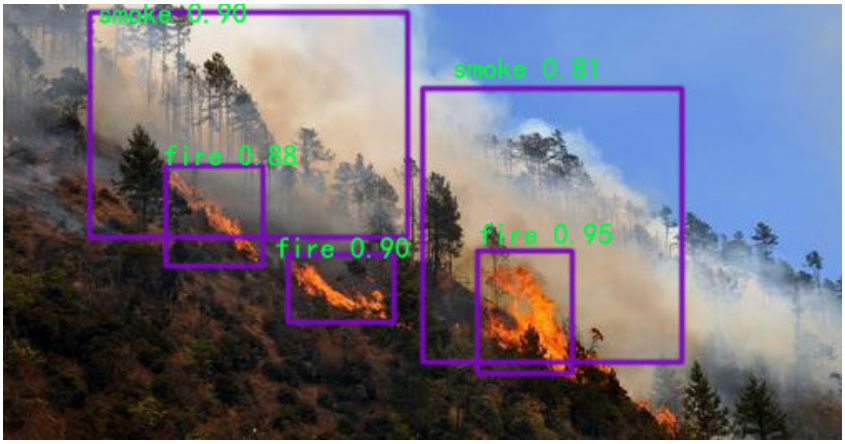

smoke 0.83
fire 0.88
smoke 0.85
fire 0.91
fire 0.77
fire 0.76

(**c**)

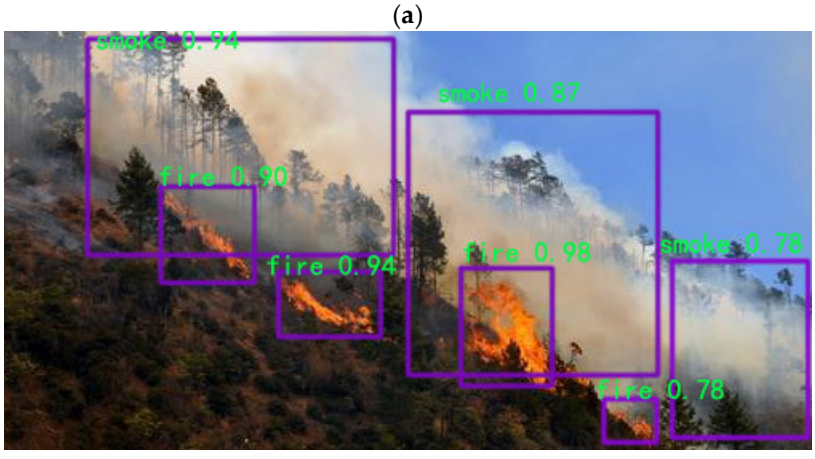


(**d**)

Figure 8. The detection results of the YOLOv5 and YOLOv5-IFFDM models for forest fire and smoke image. (**a**,**b**) Surface fire and smoke, (**c**,**d**) Canopy fire and smoke. (**a**) Surface fire and smoke detection using YOLOv5; (**b**) Surface fire and smoke detection using YOLOv5-IFFDM; (**c**) Canopy fire and smoke using YOLOv5; (**d**) Canopy fire and smoke using YOLOv5-IFFDM.



(**a**)



(**b**)

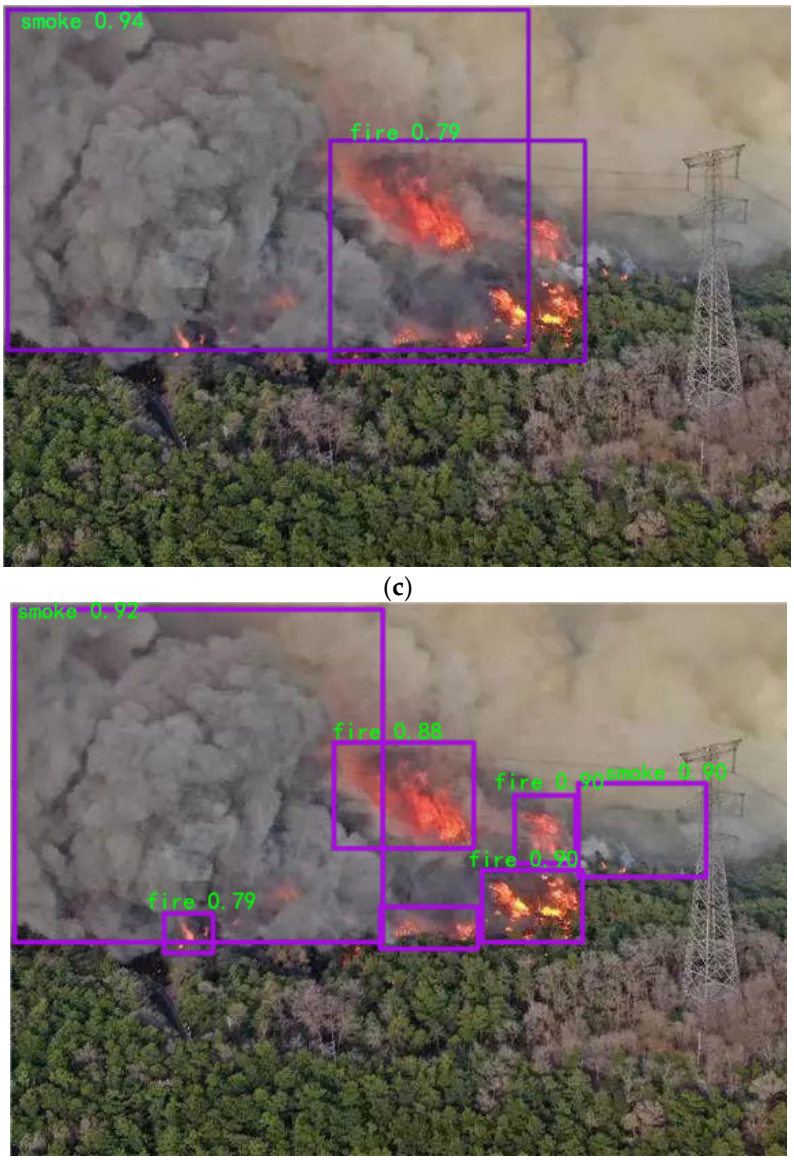


Figure 9. The detection results of the YOLOv5 and YOLOv5-IFFDM models for forest fire and smoke from the UAV camera view. (**a**) Fire and smoke detection using YOLOv5; (**b**) Fire and smoke detection using YOLOv5-IFFDM; (**c**) Fire and smoke detection using YOLOv5; (**d**) Fire and smoke detection using YOLOv5-IFFDM.

5. Discussion

Forests are a valuable ecological resource on Earth, and forest fires pose a serious threat to forests and the Earth's ecology, and can have far-reaching effects. Over the past fifty years, the area of forests burned by forest fires has increased by as much as 10 times per year. If forest fires are not detected and extinguished in time, they may cause serious ecological damage. With the continuous development and maturity of target detection technologies, it is of practical importance to use them to detect and identify forest fires and take timely and appropriate measures to extinguish them.

Therefore, based on the above reasons, forest fire detection techniques need further research and development to improve detection accuracy. Through experiments, it is found that the YOLOv5 neural network model has good detection performance in identifying forest fires. However, in the early stage of fire, its detection accuracy is low, and it is difficult to detect the fire in time and accurately. The main reason is the low number of pixels in the picture and the small scope of the fire. Therefore, in order to improve the

accuracy and speed of detection of forest fires, we introduce the CBAM attention mechanism into the network to improve the accuracy of fire detection. We improve the loss function to improve the training and inference of the detection algorithm. We use a pyramid pooling structure to improve the robustness of the model. We also employ random mosaic enhancement techniques, merging convolutional and normalization layers, and re-clustering prior frames to improve detection accuracy and speed.

To further validate the specific performance of our model for forest fire and smoke detection, we compare it with methods commonly used in the literature in recent years. These detection methods are relatively classical and open-source. The performance of different detection methods can only be compared if they are tested on the same dataset; references [33,37–39,42,43] and others cannot be compared for specific performance because they are not open source. We put the classical and open-source methods to the test on our self-built forest fire dataset. The test results are shown in Table 4. These results show that our model has more advantages in terms of speed and accuracy.

Table 4. Comparison of different methods.

Model	Number of Model Parameters (M)	mAP (%)	mAP@0.5 (%)	FPS
Faster-RCNN [42]	107.98	84.32	67.02	13
SSD [44]	90.58	85.28	68.86	52
YOLO V3 [23]	234.71	84.51	70.28	45
YOLO V4 [26]	243.92	86.02	71.06	47
YOLO V4-tiny [45]	22.58	77.60	62.53	**90**
YOLO V5m [27]	21.2	86.05	71.23	59
YOLO V5-IFFDM(ours)	**20.3**	**86.92**	**71.66**	**75**

The forest fire detection model proposed in this paper has been improved compared with YOLOv5 regarding detection accuracy and detection speed, but the model still has room for improvement. For example, in low illumination, some small fire areas may be missed, and some fire points may only be a few dozen pixels in size when taken from a high altitude and distance, which may result in their misdetection as well. Further research is needed to improve the detection accuracy and speed up the detection of small fire spots.

The experimental results show that the forest fire detection model proposed in this paper has good application prospects. The model can be fitted to drones or watchtowers for forest image acquisition to detect quickly if a forest fire occurs. Compared with the traditional method of arranging sensors to detect forest fires or manual inspection, this method has a larger detection range, lower cost, better, more timely and more efficient detection, and has a positive effect on protecting forests from the threat of forest fires.

In addition, other types of data can be considered to train the neural model, for example, satellite image data and remote sensing data, so that the detection range of forest fires can be greatly improved. The high precision satellite data can better guarantee the timely detection of forest fires, which is of great significance for forest protection. Using satellite image data or remote sensing data for fire detection will be our further research direction.

6. Conclusions and Outlook

This paper focuses on improving the feature extraction network to address the shortcomings of YOLOv5 in fire detection applications by improving the attention mechanism, prior frame, loss function, convolutional layer and BN layer merging to make YOLOv5 more efficient in fire detection. The algorithm has experimented with a self-designed and labeled fire-detection dataset. The experimental results show that the improved method

in this paper has better fire-detection robustness than the original YOLOv5 detection algorithm, both in terms of accuracy and speed. Based on the dataset, the detection algorithm achieved 91.6% accuracy, 83.2% recall, 84.5% mAP, and an average detection speed of 13.3 ms. In the future, in addition to optimizing and improving the network, we will expand the existing fire dataset to increase sample diversity and improve the sample quality of the training set, as well as expand the existing fire dataset.

Author Contributions: Conceptualization, J.L. and X.L.; methodology, J.L. and X.L.; software, J.L.; validation, J.L.; formal analysis, J.L.; data curation, J.L.; writing—original draft preparation, J.L.; writing—review and editing, J.L. and X.L.; visualization, J.L.; supervision, X.L.; project administration, J.L. and X.L.; funding acquisition, X.L. All authors have read and agreed to the published version of the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by 2020 Beijing Higher Education "Undergraduate Teaching Reform and Innovation Project " (Project No. 173), 2020 Teaching Reform Research Project of the Teaching Instruction Committee of Electronic Information (Project No. 2020-YB-09), 2020 Education Teaching Reform Research Project of Beijing Technology and Business University (Project No. jg205105), 2021 Talent Training Quality Construction Project of Beijing Technology and Business University (Project No. 9008021060).

Data Availability Statement: The data used to support the findings of this study are available from the corresponding author upon request.

Acknowledgments: The authors would like to thank the School of Artificial Intelligence, Beijing Technology and Business University for assistance with simulation verifications related to this work.

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

References

1. Wen, X. Research on Key Techniques and Methods of Forest Resources Second Class Survey. Master's Thesis, Nanjing Forestry University, Nanjing, China, 2017.
2. Wang, M.W.; Mao, W.; Dou, Z.; Li, Y. Fire recognition based on multi-Channel convolutional neural Network. *Fire Technol.* **2018**, *54*, 531–554.
3. Dong, X. Research on Forest Fire Detection System Based on FY3 Remote Sensing Images. Master's Thesis, Harbin Engineering University, Harbin, China, 2018.
4. Yan, Y. Application of Visual Information Network Foundation Platform in Forest Fire Prevention. *For. Sci. Technol. Inf.* **2019**, *51*, 18–21
5. Xiang, X.B. The Research of Smoke Detection Algorithm on Video. Master's Thesis, Zhejiang University, Han Zhou, China, 2017.
6. Xiao, X.; Kong, F.Z.; Liu, J.H. Monitoring Video Fire Detection Algorithm Based on Dynamic Characteristics and Static Characteristics. *Comput. Sci.* **2019**, *46*, 284–286.
7. Chen, T.H.; Wu, P.H.; Chiou, Y.C. An early fire-detection method based on image processing. In Proceedings of the 2004 International Conference on Image Processing, ICIP'04, Singapore, 24–27 October 2004.
8. Chen, J.; He, Y.; Wang, J. Multi-feature fusion based fast video flame detection. *Build. Environ.* **2010**, *45*, 1113–1122.
9. Celik, T.; Demirel, H.; Ozkarmanli, H. Automatic fire detection in video sequences. *Fire Saf. J.* **2006**, *6*, 233–240.
10. Li, Z.; Mihaylova, L.S.; Isupova, O.; Rossi, L. Autonomous flame detection in videos with a dirichlet process Gaussian mixture color model. *IEEE Trans. Ind. Inform.* **2017**, *14*, 1146–1154.
11. Emmy, P.C.; Vinsley, S.S.; Suresh, S. Efficient flame detection based on static and dynamic texture analysis in forest fire detection. *Fire Technol.* **2018**, *54*, 255–288.
12. Foggia, P.; Saggese, A.; Vento, M. Real-time fire detection for video-surveillance applications using a combination of experts based on color, shape, and motion. *IEEE Trans. Circuits Syst. Video Technol.* **2015**, *25*, 1545–1556.
13. Han, X.F.; Jin, J.S.; Wang, M.J.; Jiang, W.; Gao, L.; Xiao, L.P. Video fire detection based on Gaussian mixture model and multi-color features. *Signal Image Video Process.* **2017**, *11*, 1419–1425.
14. Jian, W. Research on Fire Detection Method Based on Video Smoke Motion Detection. Master's Thesis, Nanchang Aviation University, Nanchang, China, 2018.
15. Dimitropoulos, K.; Barmpoutis, P.; Grammalidis, N. Higher order linear dynamical systems for smoke detection in video surveillance applications. *IEEE Trans. Circuits Syst. Video Technol.* **2017**, *27*, 1143–1154.
16. Wang, S.; He, Y.; Yang, H.; Wang, K.; Wang, J. Video smoke detection using shape, color and dynamic features. *J. Intell. Fuzzy Syst.* **2017**, *33*, 305–313.

17. Appana, D.K.; Islam, R.; Khan, S.A.; Kim, J.M. A Video based smoke detection using smoke flow pattern and spatial-temporal energy analyses for alarm systems. *Inf. Sci.* **2017**, *418*, 91–101.
18. Fu, T.-j.; Zheng, C.-e.; Tian, Y.; Qiu, Q.-m.; Lin, S.-j. Forest Fire Recognition Based on Deep Convolutional Neural Network Under Complex Background. *Comput. Mod.* **2016**, *0*, 5257.
19. Frizzi, S.; Kaabi, R.; Bouchouicha, M.; Ginoux, J.M.; Moreau, E.; Fnaiech, F. Convolutional neural network for video fire and smoke detection. In Proceedings of the IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society, Florence, Italy, 23–26 October 2016; IEEE: New York, NY, USA, 2016; pp. 877–882.
20. Sandler, M.; Howard, A.; Zhu, M. Mobilenetv2: Inverted Residuals and Linear Bottlenecks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.
21. 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.
22. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 6517–6525.
23. Redmon, J.; Farhadi, A. YOLOv3: An Incremental improvement. In *Computer Vision and Pattern Recognition*; Springer: Berlin/Heidelberg, Germany, 2018.
24. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition. Honolulu, HI, USA, 21–26 July 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 936–944.
25. Wang, Y.; Yan, G.; Meng, Q.; Yao, T.; Han, J.; Zhang, B. DSE-YOLO: Detail semantics enhancement YOLO for multi-stage strawberry detection. *Comput. Electron. Agric.* **2022**, *198*, 107057.
26. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. YOLOv4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
27. Jocher, G. YOLOv5. Available online: https://github.com/ultralytics/yolov5 (accessed on 6 January 2023).
28. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 7132–7141.
29. Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; Hu, Q. ECA-Net: Efficient channel attention for deep convolutional neural networks. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 11531–11539.
30. Lu, K.; Xu, R.; Li, J.; Lv, Y.; Lin, H.; Liu, Y. A Vision-Based Detection and Spatial Localization Scheme for Forest Fire Inspection from UAV. *Forests* **2022**, *13*, 383.
31. Pan, J.; Ou, X.; Xu, L. A Collaborative Region Detection and Grading Framework for Forest Fire Smoke Using Weakly Supervised Fine Segmentation and Lightweight Faster-RCNN. *Forests* **2021**, *12*, 768.
32. Jiao, Z.; Zhang, Y.; Xin, J.; Mu, L.; Yi, Y.; Liu, H.; Liu, D. A Deep Learning Based Forest Fire Detection Approach Using UAV and YOLOv3. In Proceedings of the 1st International Conference on Industrial Artificial Intelligence (IAI), Shenyang, China, 23–27 July 2019; pp. 1–5.
33. 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.
34. Bouguettaya, A.; Zarzour, H.; Taberkit, A.M.; Kechida, A. A Review on Early Wildfire Detection from Unmanned Aerial Vehicles Using Deep Learning-Based Computer Vision Algorithms. *Signal Process* **2022**, *190*, 108309.
35. Almalki, F.; Soufiene, B.; Alsamhi, S.; Sakli, H. A Low-Cost Platform for Environmental Smart Farming Monitoring System Based on IoT and UAVs. *Sustainability* **2021**, *13*, 5908.
36. Hu, Y.; Zhan, J.; Zhou, G.; Chen, A.; Cai, W.; Guo, K.; Hu, Y.; Li, L. Fast Forest fire smoke detection using MVMNet. *Knowl.-Based Syst.* **2022**, *241*, 108219.
37. Wahyono; Harjoko, A.; Dharmawan, A.; Adhinata, F.D.; Kosala, G.; Jo, K.-H.G. Real-Time Forest Fire Detection Framework Based on Artificial Intelligence Using Color Probability Model and Motion Feature Analysis. *Fire* **2022**, *5*, 23.
38. Guede-Fernández, F.; Martins, L.; de Almeida, R.V.; Gamboa, H.; Vieira, P. A Deep Learning Based Object Identification System for Forest Fire Detection. *Fire* **2021**, *4*, 75.
39. Benzekri, W.; El Moussati, A.; Moussaoui, O.; Berrajaa, M. Early Forest Fire Detection System using Wireless Sensor Network and Deep Learning. *Int. J. Adv. Comput. Sci. Appl.* **2020**, *11*, 5.
40. Cao, Y.; Yang, F.; Tang, Q.; Lu, X. An Attention Enhanced Bidirectional LSTM for Early Forest Fire Smoke Recognition. *IEEE Access* **2019**, *7*, 154732–154742.
41. 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.
42. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Trans. Anal. Mach. Intell.* **2017**, *39*, 1137–1149.
43. Wang, Y.; Hua, C.; Ding, W.; Wu, R. Real-time detection of flame and smoke using an improved YOLOv4 network. *SIViP* **2022**, *16*, 1109–1116.

44. 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; Lecture Notes in Computer Science: 2016; Volume 9905, Springer, Cham. DOI: https://doi.org/10.1007/978-3-319-46448-0_2
45. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. YOLO V4-Tiny. Available online: https://github.com/WongKinYiu/ScaledY-OLOv4/tree/yolov4-tiny (accessed on 18 January 2023).