Article

Forest Flame Detection in Unmanned Aerial Vehicle Imagery Based on YOLOv5

Haiqing Liu, Heping Hu, Fang Zhou and Huaping Yuan *

Hunan Police Academy, Changsha 410081, China
* Correspondence: yuanhuaping@hnpa.edu.cn; Tel.: +86-731-85623008

Abstract: One of the major responsibilities for forest police is forest fire prevention and forecasting; therefore, accurate and timely fire detection is of great importance and significance. We compared several deep learning networks based on the You Only Look Once (YOLO) framework to detect forest flames with the help of unmanned aerial vehicle (UAV) imagery. We used the open datasets of the Fire Luminosity Airborne-based Machine Learning Evaluation (FLAME) to train the YOLOv5 and its sub-versions, together with YOLOv3 and YOLOv4, under equal conditions. The results show that the YOLOv5n model can achieve a detection speed of 1.4 ms per frame, which is higher than that of all the other models. Furthermore, the algorithm achieves an average accuracy of 91.4%. Although this value is slightly lower than that of YOLOv5s, it achieves a trade-off between high accuracy and real-time. YOLOv5n achieved a good flame detection effect in the different forest scenes we set. It can detect small target flames on the ground, it can detect fires obscured by trees or disturbed by the environment (such as smoke), and it can also accurately distinguish targets that are similar to flames. Our future work will focus on improving the YOLOv5n model so that it can be deployed directly on UAV for truly real-time and high-precision forest flame detection. Our study provides a new solution to the early prevention of forest fires at small scales, helping forest police make timely and correct decisions.

Keywords: forest fire; forest police; flame detection; YOLOv5; UAV imagery; deep learning

check for updates

Citation: Liu, H.; Hu, H.; Zhou, F.; Yuan, H. Forest Flame Detection in Unmanned Aerial Vehicle Imagery Based on YOLOv5. *Fire* **2023**, *6*, 279. https://doi.org/10.3390/fire6070279

Academic Editor: Guoxiong Zhou

Received: 14 June 2023
Revised: 12 July 2023
Accepted: 13 July 2023
Published: 19 July 2023

1. Introduction

Forest fires are a natural disaster; they come on with strong suddenness, cause great destruction, and are extremely difficult to rescue [1]. As the global climate continues to get warmer, forest fires have occurred more frequently in recent years, causing great harm to human safety and the ecological environment. In China, as the major responsible department of the Forest Fire Early Warning and Monitoring Information Center, forest police have paid much attention to forest fire prevention, which is subdivided into forest fire detection, prediction, and forecasting. Forest fires usually spread from the ground to the trunk, and then to the tree canopy, in different degrees [2]. Therefore, flame detection is the key to controlling the spread of forest fires. As a specific application of object detection, flame detection plays an important role in the early detection, prevention, and control of fire sources [3].

The previous studies on flame detection are based on sensors, which are supposed to collect the data of the temperature, smoke concentration, and other information for fire detection. Sensor-based detection systems have a good detection effect indoors, but have obvious defects when they are applied in open forest areas, such as their high cost to set up and maintain [4–6]. Infrared [7] and ultraviolet [8] detectors are susceptible to environmental factors and their detection range is limited. In addition, the sensor-based flame detection system does not issue alarm orders until the monitored environmental parameters reach a certain threshold, resulting in a slow response speed [9].

Flame detection based on image recognition has developed gradually in recent years and has become the mainstream of fire detection research. Researchers have extracted the color features of flame images for detection and transformed the images into another color space [10,11]. A threshold is then set to distinguish whether the pixels are flames or not. In addition to the flame color, the flame edge [12] and texture [13,14] are also used in flame detection. To improve the detection accuracy and accelerate the speed, machine learning algorithms are widely used in flame detection. Negara et al. [15] used the Decision Tree and Bayesian network to detect forest fires. Jain and Sharma [16] proposed a fire detection and prediction system based on machine learning models and meteorological conditions. In addition, there are a number of detection methods using Support Vector Machines [17,18]. Although the above methods have achieved good detection results, all of them rely on the manual selection of flame characteristics. Moreover, due to the complexity of flame features and the interference factors that occur in practical applications, flame-like objects, such as light and sunlight, are difficult to distinguish from flames in forest areas [19].

Recently, deep learning algorithms have been applied in fire detection due to their superior performance in automatic feature extraction [20,21]. Deep learning algorithms can extract deep features to obtain higher detection accuracy [22,23]. Unmanned aerial vehicles (UAV), which are flexible and low-cost to deploy, have emerged as a potential platform for fire detection. UAVs are more flexible than remote sensing satellites in capturing small flames [24–27]. Kinaneva et al. [28] showed the results achieved by the Faster Region-based Convolutional Neural Network (Faster R-CNN) algorithm for detecting smoke and flames in UAV images. The Faster R-CNN, used by Barmpoutis et al. [29], had the second-to-last result among all the tested object detection algorithms, with F1 score rates of 72.7%, 70.6%, and 71.5% for flame, smoke, and fire and smoke, respectively. Other studies introduced the Single Shot Multibox Detector (SSD) algorithm to identify forest fires. Although the results were acceptable, the SSD algorithm was considered to be the least efficient detector [29,30]. Redmon et al. [31], who proposed the You Only Look Once (YOLO) algorithm, realized that the two tasks of classification and positioning in one evaluation could be completed by a single neural network. Redmon and Farhadi [32] proposed YOLOv2 by adopting a new network structure, Darknet19. Alexandrov et al. [33] employed five different techniques to detect forest fire smoke from UAV-based RGB images, three of which were based on deep learning techniques, including Faster R-CNN, SSD, and YOLOv2. Among these detectors, YOLOv2 achieved impressive results, providing the best inference speed (FPS = 6), precision (100%), recall (98.3%), F1 score (99.14%), and accuracy (98.3%). Then, Redmon and Farhadi [26] continued to propose the YOLOv3 algorithm, which adopted multi-scale prediction and a better classification network structure [34] with the same accuracy as the SSD algorithm; however, the detection speed of YOLOv3 was much faster than that of SSD, by three times [35]. Li and Zhao [36] made a comparative analysis of YOLOv3, Region-based Fully Convolutional Networks (R-FCN) [37], Faster-RCNN [38], and SSD [39]. The results showed that the YOLOv3 algorithm was better than the other algorithms in terms of its fire detection accuracy and robustness. Bochkovskiy et al. [40] developed a real-time flame detection system based on the YOLOv4 model, which also achieved good detection results.

Since 2020, YOLOv5 and its sub-versions have been proposed successively. The series of models have been proven to be qualified with the advantages of fast convergence, high precision, and strong customization, even for detecting small objects. Moreover, it also possesses strong real-time processing power and low hardware computing requirements, which means that it is also very portable. However, there are few studies comparing the YOLOv5 models in detecting forest flames, and it is not clear which YOLO is more suitable for working in complex natural conditions and scenarios. The purposes of this study are: (1) to use YOLOv5 and its sub-version to detect forest fires from images captured by UAV; (2) to test and compare the flame detection performance of YOLOv5 under complex environmental conditions. The remaining part of our paper is organized as follows: the data used in our study, the pre-treatment method, the basic framework of the YOLOv5 model, and the evaluation indicators will be described in detail in the Section 2; the Section 3 will compare the performances of various models, including YOLOv3, YOLOv4, YOLOv5, and their variants. In the Section 4, the adaptability and deficiency of the YOLOv5 model in fire detection will be discussed and we will put forward our future research ideas.

2. Materials and Methods

2.1. Data

The data used in our study were derived from the publicly available datasets of FLAME in Northern Arizona University [41]. The data were collected by the Flagstaff (AZ, USA) Fire Department using a Phantom 3 professional drone equipped with a ZenmuseX4S camera (Figure 1). Fire managers conducted the test in a ponderosa pine forest on Observatory Mesa on 16th January 2020. It was a cloudy day without wind, and the temperature was 43 °F (∼6 °C) [41]. As the camera on the UAV did not have an optical zoom function, the flight altitude determined the resolution of the image and the size, number, and orientation of the flames. Therefore, by changing the flight altitude of the UAV, the datasets simulated forest floor fire scenes of small objects and occluded objects and multiple objects under various complex environmental conditions, with a total of 1853 drone aerial images.

Figure 1. The drone and the video camera used in the study.

2.2. Data Pre-Processing

To increase the accuracy of the model training and reduce the training time, we adopted data augmentation methods, including Mosaic and Cutout. Among them, Mosaic augmentation included random clipping, scaling, flipping, and changing the Hue, Saturation, and Value (HSV). To improve the training speed of the model, we also adopted the rectangular inference method.

2.2.1. Data Augmentation

Due to the uneven distribution of small objects in the images, the training is not sufficient. To improve the detection of small objects, we used Mosaic augmentation to make a single imagery from four training images that have been flipped, scaled, HSV changed, and cropped randomly (Figure 2).

Cutout is to randomly select a square area of a fixed size and fill it with all zeros. In order to avoid the influence of the filling zeros on the training, a central normalization operation was performed on the data (Figure 3).

2.2.2. Rectangular Inference

The YOLO algorithm solved the problem of different input image size ratios by directly scaling and filling it to a fixed size, which results in redundant information and generates a lot of meaningless candidate boxes (Figure 4a). To reduce this redundant information, the rectangular inference stipulated that the input image should be a rectangle, the side length of which was divisible by the step size (the default is 32), rather than a square. The rectangular inference minimized padding by scaling the image to a size divisible by the step size and closest to the size required for input (Figure 4b).

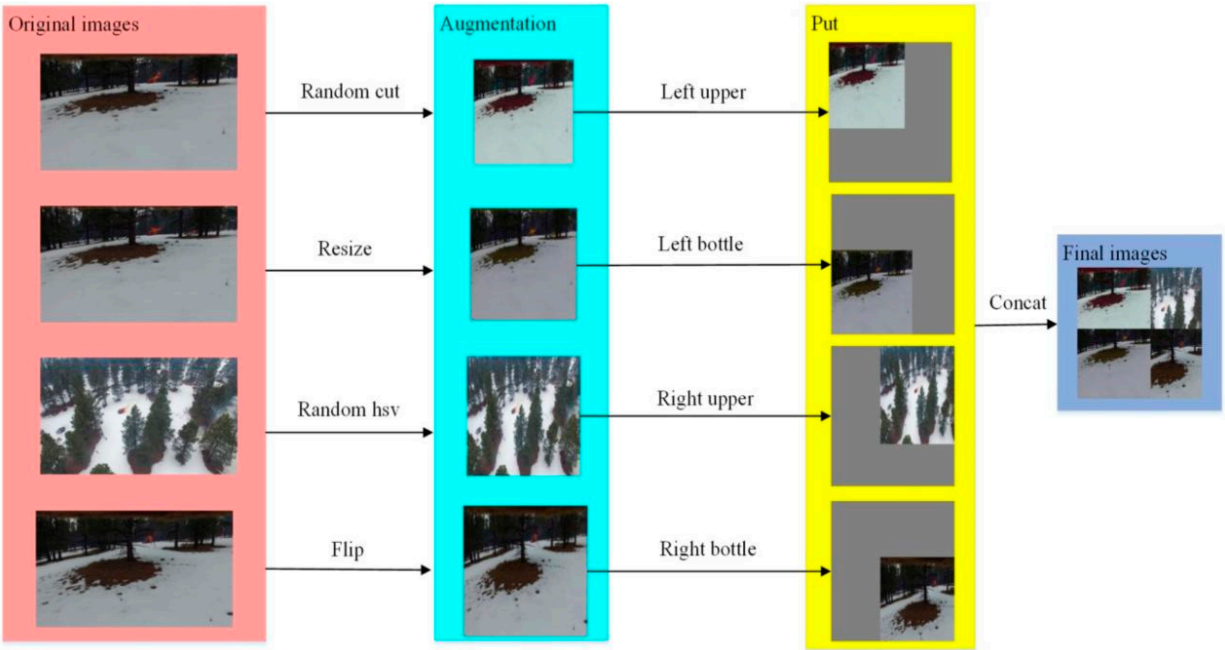


Figure 2. Mosaic augmentation.

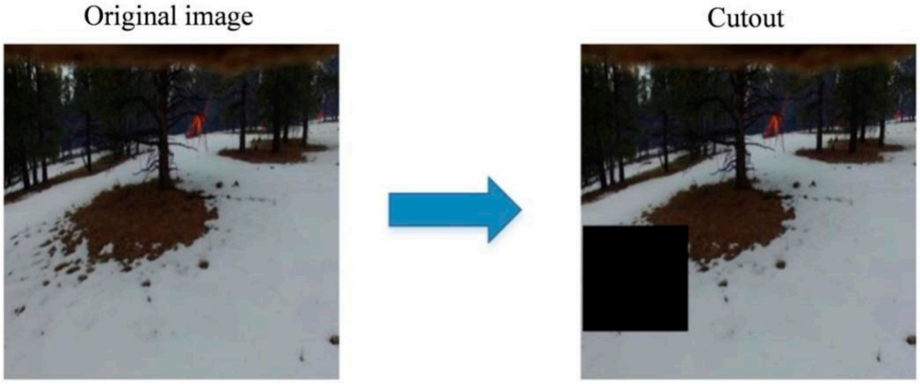


Figure 3. Cutout (filled with all zeros).

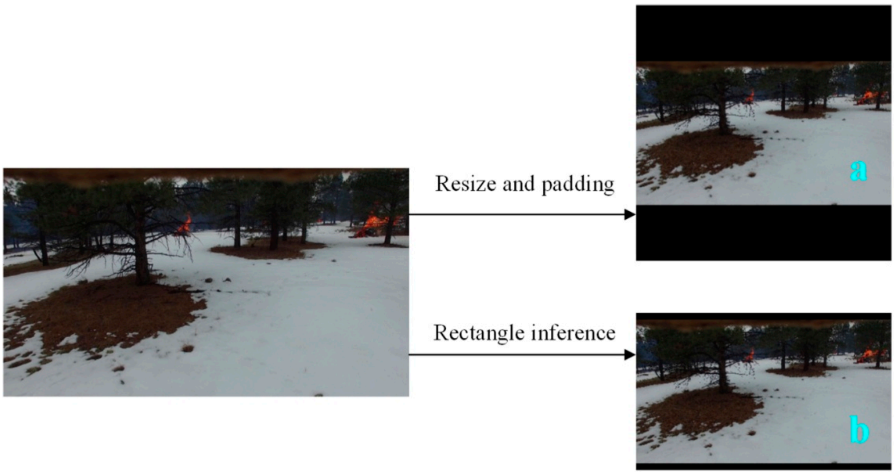


Figure 4. Rectangular inference, where (**a**) is the previous practice, first scale the long side to the object size, and then add 0 to the short side; (**b**) is the practice result of YOLOv5, with the side length to be an integer multiple of the step size (32 by default).

2.2.3. Datasets Division

The performance of the flame detection model was closely connected with the quality of the datasets. The experimental data of this study consisted of 1853 images. These images were divided randomly into training set, validation set, and test set, according to the 8:1:1 proportion [24]. The LabelMe software was used to manually annotate the images and generate XML files containing flame position coordinates to bring them into the model for training. See Table 1 for the number of pictures and flame labels. In addition, to further verify the performance of the flame detection model, we obtained some flame images from the Internet for testing our model, and these images were independent of the training of the model.

Table 1. Datasets division.

Datasets	Number of Pictures	Number of Fire Labels
Training set	1483	3572
Validation set	185	512
Test set	185	381

2.3. *YOLOv5 Network Structure*

YOLOv5 was nearly 90% smaller than YOLOv4, which is very suitable for deployment in resource-constrained embedded devices, such as UAVs, to achieve real-time monitoring. YOLOv5 consisted of three parts: the backbone, the neck, and the detection head (Figure 5). Some of the major components in YOLOv5 are listed below.

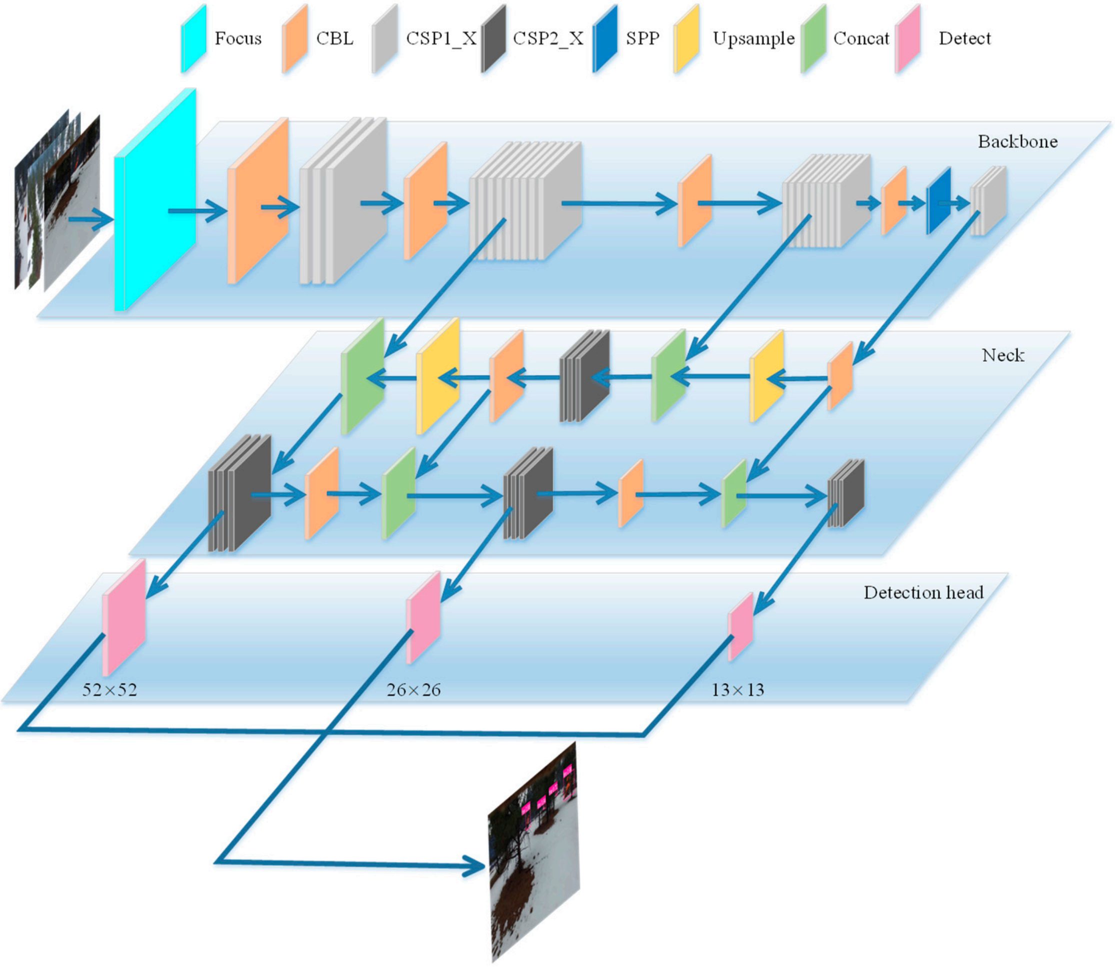


Figure 5. Overall architecture of YOLOv5.

2.3.1. Focus

The Focus module (Figure 6) was the first layer of the backbone. Its function was to reduce the amount of calculations and to accelerate the training speed. The Focus module divided the input image (default size is 3 × 640 × 640) into 4 slices, and each slice size is 3 × 320 × 320. Then, the four slices were connected in series to output a 12 × 320 × 320 feature map. After passing through the convolution layer, a feature map of 32 × 320 × 320 was generated. Finally, the results were output to the next layer through the batch normalization (BN) layer and Leaky ReLU layer.

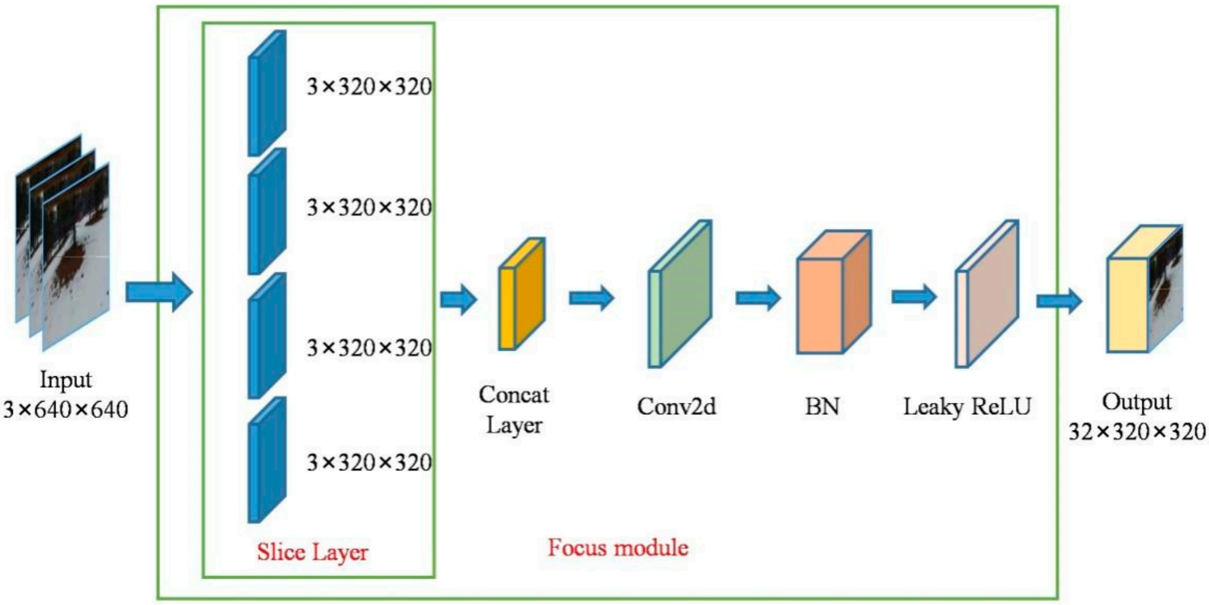


Figure 6. Structure and component of the Focus module.

2.3.2. Conv2d + BN + Leaky ReLU (CBL)

The CBL module (Figure 7) was the second layer of the backbone, which consisted of a convolution layer, BN layer, and Leaky ReLU layer. The CBL module was an important component of the whole network, even though it was the smallest module in YOLOv5.

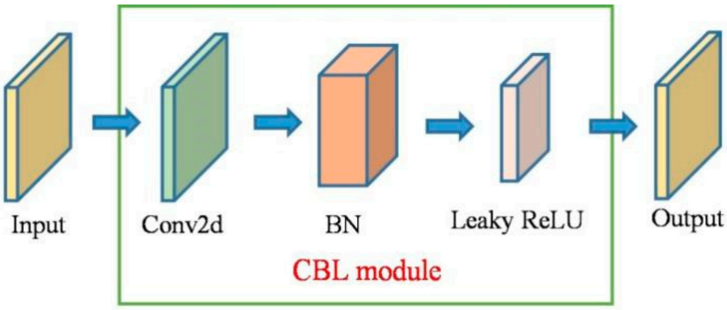


Figure 7. Structure and component of the CBL module.

2.3.3. Cross Stage Partial Networks (CSP1_X)

The CSP1_X module (Figure 8) was used to extract the deep features. In this module, the feature map was divided into two parts, and these modules were combined by a cross-stage hierarchical structure to ensure accuracy and to reduce the amount of calculations. CSP1_X consisted of a CBL, residual component, and so on. The Resunit consisted of two CBL modules. X represented the number of residual components. The output of the residual component was the addition of the output of the two CBL modules and the original input.

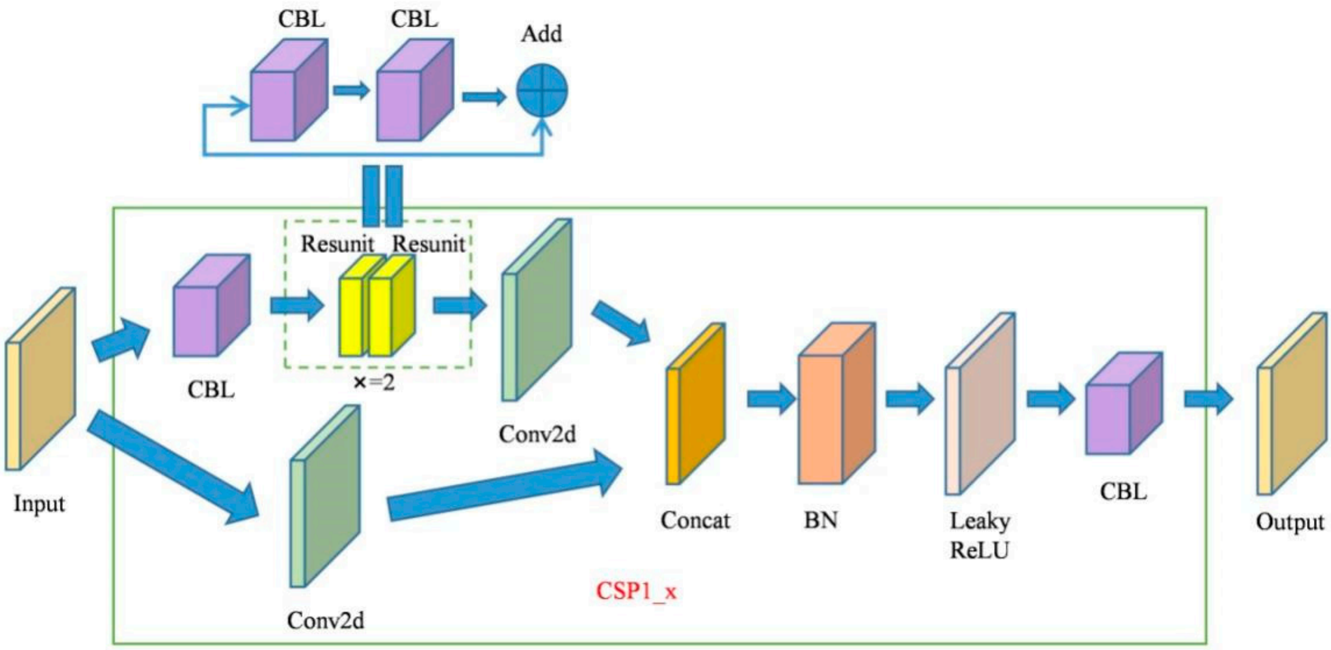


Figure 8. Structure and component of the CSP1_X module.

2.3.4. Spatial Pyramid Pooling (SPP)

The SPP module (Figure 9) transformed the feature map into a feature vector to improve the acceptance domain of the network. SPP convolves the input in the CBL module, and then passed it through three parallel max pooling layers. Finally, the pooled feature map was spliced with the convolved feature map, and the output was obtained by convolving again.

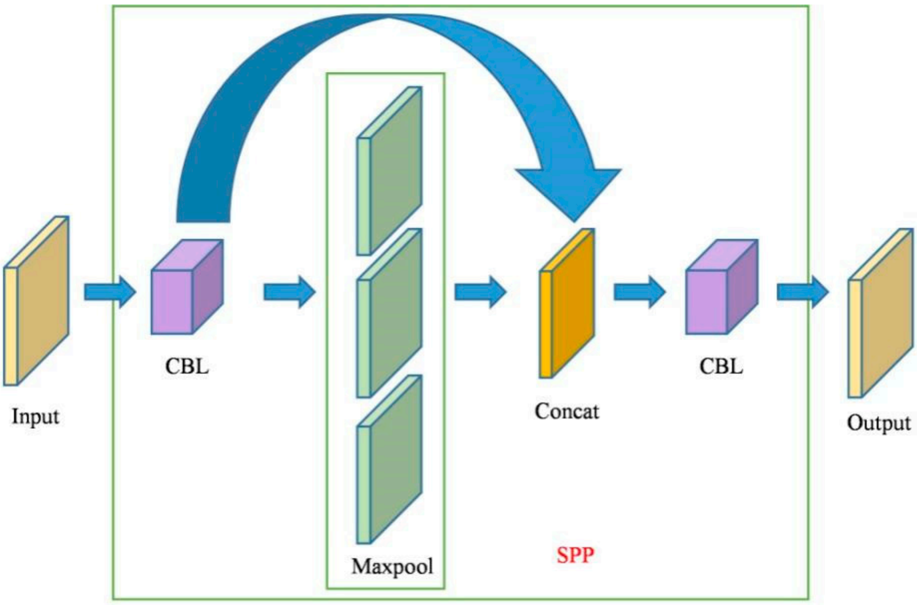


Figure 9. Structure and component of the SPP module.

2.3.5. CSP2_X

The CPS2_X module (Figure 10) was the first layer of the neck. CPS1_X had a similar structure to CPS2_X, except that X in CPS1_X represented the number of residual components. The number of CBLs in CSP2_X should be twice as many as X.

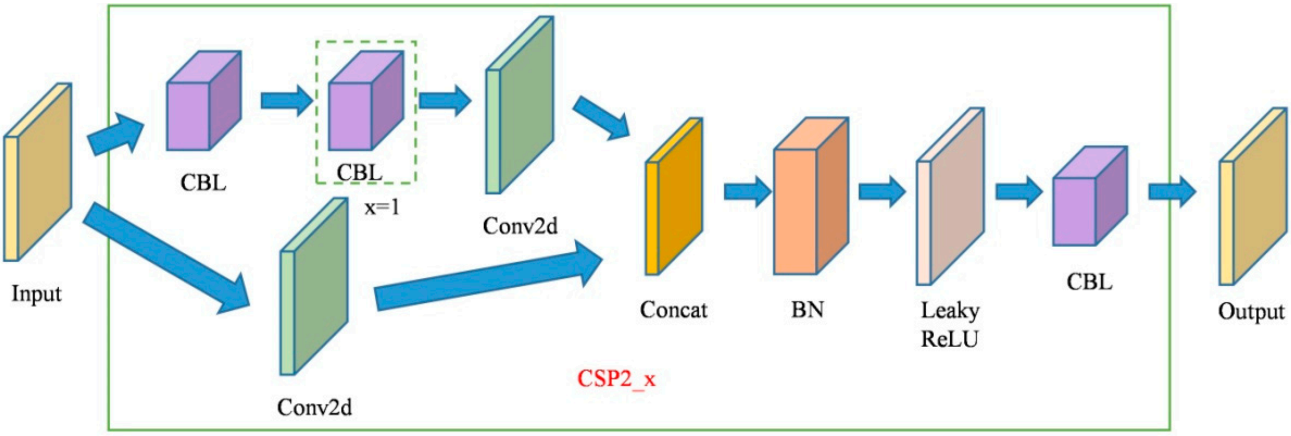


Figure 10. Structure and component of the CSP2_X module.

2.4. Model Training

2.4.1. Training Platform and Settings

The experiment was deployed on a computer with an i9-7920X CPU and a GeForce RTX 2080Ti GPU. We used the PyTorch deep learning framework for modeling, and adopted the CUDA, Cudnn, and OpenCV libraries to train and to test the forest fire detection model.

The pre-training weights provided by the YOLO developers during training were used in the YOLO models (V3, V4, and V5). Among them, YOLOv3 and YOLOv4 trained 2000 Iterations and YOLOv5 trained 20 epochs. To achieve the best training performance, the YOLO series configuration files could be modified with hyper-parameters. The batch size of all YOLO models was set to 16 and the pixel size was set to 608 × 608. During the training, the data augmentation technology in Section 2.2.1 was used. The augmentation coefficient of image saturation and exposure was 1.5 and the hue setting was randomly enhanced with a coefficient of 0.05.

2.4.2. Loss Function Design

The difference between the predicted value and the real value was described by the loss function in the object detection task. The total loss (Ltotal) (Equation (1)) in YOLOv5 included bounding box loss (LGIoU), confidence loss (Lconf), and classification loss (Lcla). The confidence loss and classification loss were calculated using the method of cross entropy, as shown in Equations (2) and (3) [35]. Bounding box loss (LGIoU) was calculated by Equations (4)–(6) [24,31].

$$\mathrm{L_{total}} = \mathrm{L_{GIoU}} + \mathrm{L_{conf}} + \mathrm{L_{cla}} \tag{1}$$

$$\mathrm{L_{conf}} = \lambda_{\mathrm{obj}} \sum_{\mathrm{i=0}}^{\mathrm{S^2}} \sum_{\mathrm{j=0}}^{\mathrm{B}} \mathrm{I_{ij}^{obj}} \left[\text{-}\hat{\mathrm{C}}_{\mathrm{i}}\mathrm{lnC_i}\text{-}(1\text{ - }\hat{\mathrm{C}}_{\mathrm{i}})\ \mathrm{ln}(1\text{ - }\mathrm{C_i})\right] + \lambda_{\mathrm{nobj}} \sum_{\mathrm{i=0}}^{\mathrm{S^2}} \sum_{\mathrm{j=0}}^{\mathrm{B}} \mathrm{I_{ij}^{nobj}} \left[\text{-}\hat{\mathrm{C}}_{\mathrm{i}}\mathrm{lnC_i}\text{ - }(1\ -\hat{\mathrm{C}}_{\mathrm{i}})\ \mathrm{ln}(1\text{-}\mathrm{C_i})\right] \tag{2}$$

$$\mathrm{L_{cla}} = \sum_{\mathrm{i=0}}^{\mathrm{S^2}} \sum_{\mathrm{j=0}}^{\mathrm{B}} \sum_{\mathrm{c\in cla}} \mathrm{I_{ij}^{obj}}\left[\text{-}\hat{\mathrm{p}}_{\mathrm{i}}(\mathrm{c})\ \mathrm{ln}(\mathrm{p_i}(\mathrm{c})) - (1\text{-}\hat{\mathrm{p}}_{\mathrm{i}}(\mathrm{c}))\ \mathrm{ln}(1\text{-}\mathrm{p_i}(\mathrm{c}))\right] \tag{3}$$

where $\mathrm{S^2}$ is the number of grids; B is the number of boundary boxes in each grid; $\mathrm{I_{i,j}^{obj}}$ is 1 when the object exists in the boundary box, and 0 in the other case. The $\mathrm{I_{ij}^{nobj}}$ rule is reversed; λ_{obj} and λ_{nobj} are the weight coefficients with or without objects. $\mathrm{C_i}$, $\hat{\mathrm{C}}_{\mathrm{i}}$ are the confidence values for the predicted object and the actual object; C is the object category predicted by the bounding box; $\mathrm{p_i(c)}$ is the prediction probability of c when the ith grid detects the object; $\hat{\mathrm{p}}_{\mathrm{i}}(\mathrm{c})$ is the actual probability that the ith grid detects an object that belongs to C.

The similarity between the predicted bounding box and ground-truth bounding box is measured by Intersection over Union (IoU), which is the metric to detect object accuracy (Figure 11a). IoU is a normalized index, and its value range is [0, 1] (Equation (4)).

However, when the two bounding boxes do not overlap or overlap in different directions, the calculation of IoU fails. Therefore, the generalized IoU (GIoU) [31] is used as the bounding box loss function in our study (Figure 11b). Its calculation forum is shown in Equation (5).

$$\mathrm{IoU} = \frac{\mathrm{B_{pre}} \cap \mathrm{B_{gt}}}{\mathrm{B_{pre}} \cup \mathrm{B_{gt}}} \tag{4}$$

$$\mathrm{GIoU} = \mathrm{IoU}\text{-}\frac{\left|\mathrm{C} - \left(\mathrm{B_{pre}} \cup \mathrm{B_{gt}}\right)\right|}{|\mathrm{C}|} \tag{5}$$

$$\mathrm{L_{GIoU}} = \sum_{i=0}^{S^2} \sum_{j=0}^{B} [1 - \mathrm{GIoU}] \tag{6}$$

where $\mathrm{B_{pre}}$ is the predicted bounding box and $\mathrm{B_{gt}}$ is the ground truth bounding box, C is the smallest rectangular bounding box containing $\mathrm{B_{pre}}$ and $\mathrm{B_{gt}}$, and the value range of GIoU is $[-1, 1]$.

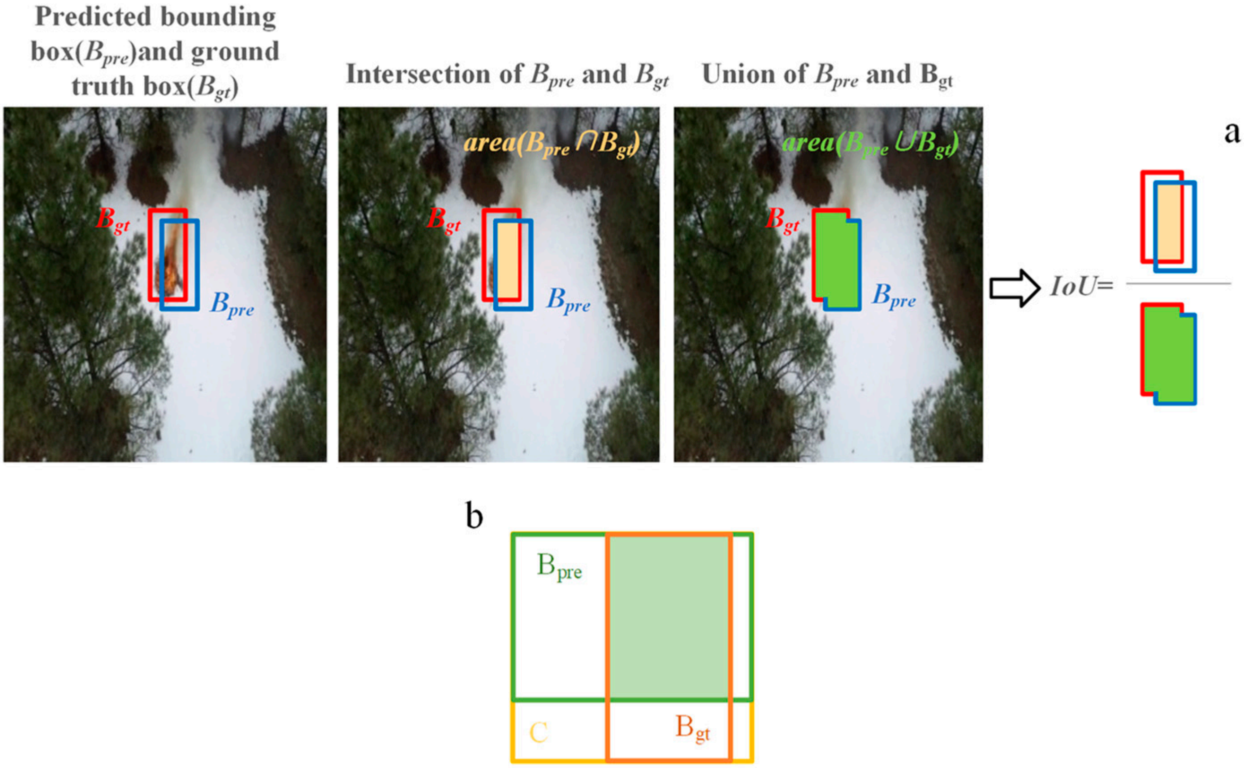

Predicted bounding box(B_{pre})and ground truth box(B_{gt})
Intersection of B_{pre} and B_{gt}
Union of B_{pre} and B_{gt}
a
$area(B_{pre} \cap B_{gt})$
$area(B_{pre} \cup B_{gt})$
B_{gt}
B_{pre}
$IoU=$
b
B_{pre}
C
B_{gt}

Figure 11. Flame detection evaluation with two IoUs: (**a**) IoU and (**b**) GIoU.

2.4.3. Evaluation Metrics

To evaluate the performance of the flame detection model, the evaluation metrics adopted in this study included Precision, Recall, F1-score, Floating Point Operations (Flops), and Mean Average Precision (mAP) (Equations (7)–(10)). Among them, Precision and Recall were calculated according to the confusion matrix, the Average Precision (AP) was used to calculate the area under the Precision-Recall (PR) curve, and mAP was used to calculate the average area under the PR curve of all categories. As we only set up the fire category in this paper, the AP was the mAP. The F1-score was the harmonic mean of Precision and Recall. The formula was as follows:

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{7}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{8}$$

$$\text{mAP} = \frac{1}{\text{C}} \sum\nolimits_{\text{k}=1}^{\text{N}} \text{Precision}(\text{k}) \Delta \text{Recall}(\text{k}) \tag{9}$$

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{10}$$

where TP (True Positive) is the correct detection of fire, FN (False Negative) is not detected from the fire image, FP (False Positive) is the number of false detections of non-fire objects. C is the number of categories; k is the threshold of the IoU. Precision(k) indicates the Precision when IoU is k. The same applies to Recall(k).

3. Results

3.1. Training Result of YOLOv5 Model

To improve the detection performance of the YOLOv5 model, we used pre-training model weights for transfer learning. These pre-training weights were trained on training datasets. The model consisted of five different sub-versions of successively increasing sizes and complexities: YOLOv5n (nano), YOLOv5s (small), YOLOv5m (medium), YOLOv5l (large), and YOLOv5x (extra-large). The five models were based on the YOLOv5 architecture, and their differences lie in the feature extraction module and network convolution kernel. These variants offered different trade-offs between speed and accuracy to accommodate different computing power and real-time requirements. We focused on YOLOv5s and YOLOv5n, which occupied less resources and had lower model complexity.

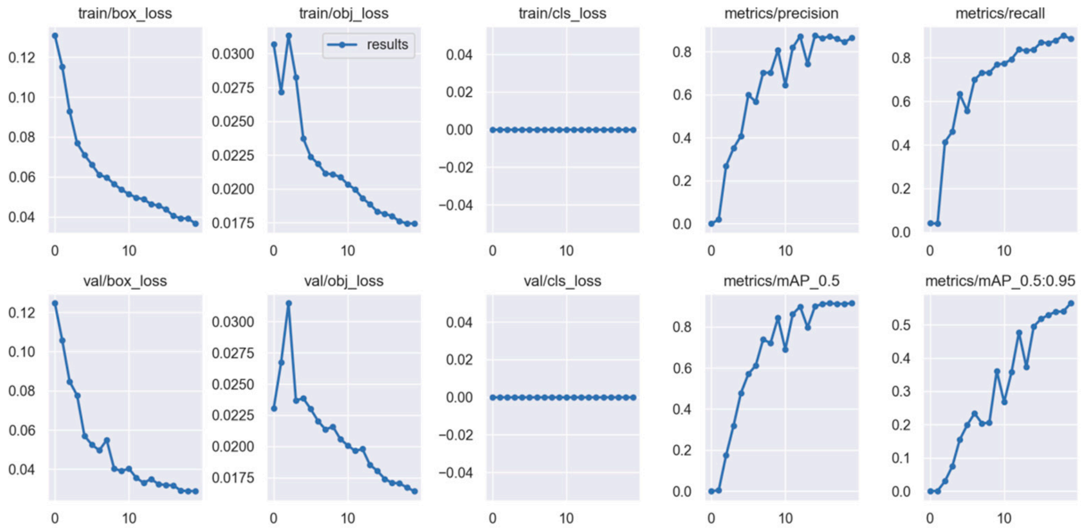
Figure 12 shows the performance of the fine-tuned YOLOv5n and YOLOv5s models after the training process. The loss of the two models showed an overall downward trend. When the IoU was 0.5 after 20 epochs, the Precision and Recall of the model reached more than 80% stably, which indicated that the two YOLOv5 models possessed quick convergence and considerable accuracy.

3.2. Comprehensive Comparison of YOLO Models

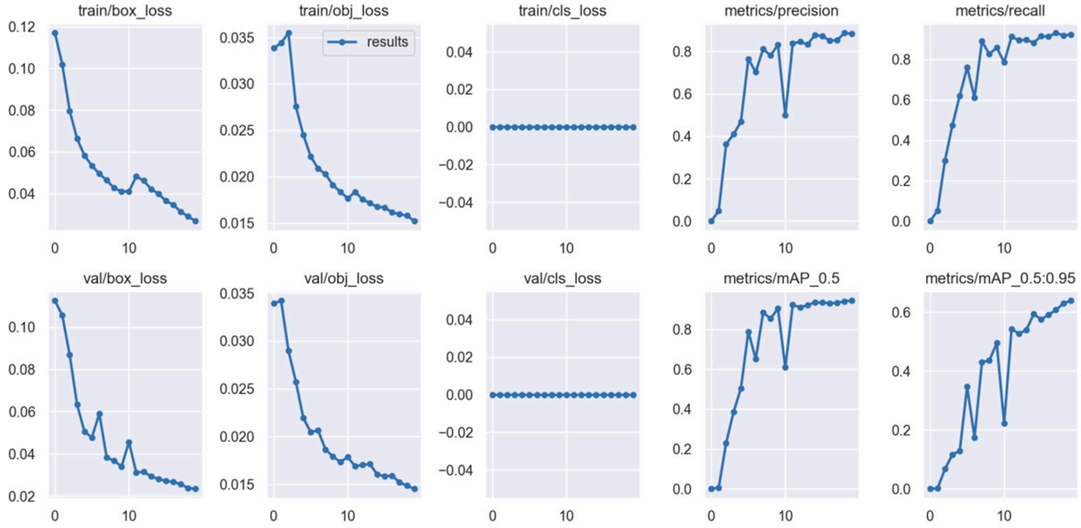
For comparison, we evaluated the performance of previous versions of YOLOv3 and YOLOv4 and their tiny models. Figure 13 shows the mapping relationship between the Precision and Recall of the flame detection in the training process. The larger the area under the PR curve was, the better the detection effect of the model was. Obviously, the detection effect of YOLOv3_tiny and YOLOv4_tiny was significantly worse than the other models. Thus, these two models might not be suitable for forest fire detection under complex conditions. However, the detection effects of the other models remained almost the same.

More detailed evaluation metrics of the nine YOLO models are listed in Table 2. From the F1-score, by integrating the Precision and Recall, the YOLOv5 models were superior to the YOLOv3 and YOLOv4 models on the whole. As another important metric, Flops represented the floating point operations and was used to measure the complexity of the model. The larger the value of this metric, the more the computing resources were consumed. Giga (G), as the unit of Flops, stood for one billion. By comparing the Flops, we found that YOLOv5m, YOLOv5l, and YOLOv5x were not our first choices due to their high computational overhead and complexity.

Although YOLOv5s was slightly superior to YOLOv5n in its mAP, Precision, and Recall, YOLOv5n was superior to YOLOv5s in its model complexity and computation speed. YOLOv5n could achieve a detection speed of 1.4ms per frame, which was much faster than the other YOLOv5 models. The flame detection results in Table 3 show that YOLOv5n offers a trade-off in detection accuracy, speed, and complexity, so YOLOv5n was more suitable for deployment on UAVs than the other YOLOs.



(**a**) Yolov5n



(**b**) YOLOv5s

Figure 12. Training results of YOLOv5n (**a**) and YOLOv5s (**b**) using training and validation set, respectively.

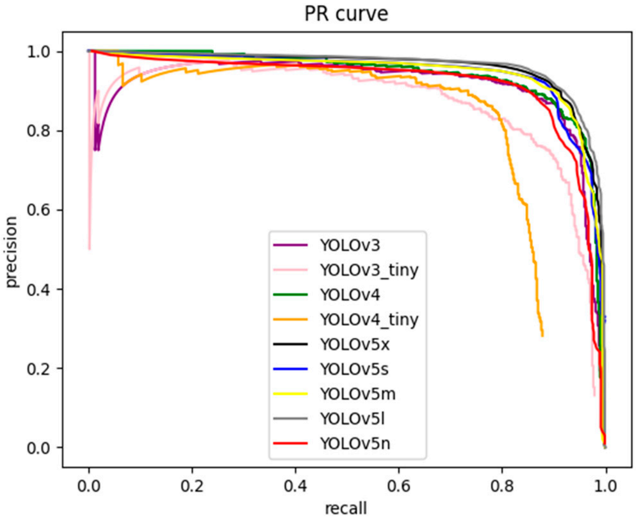
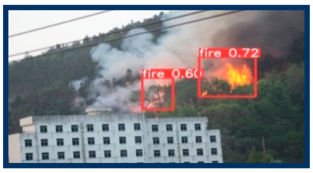
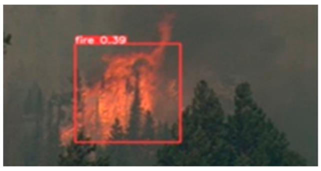
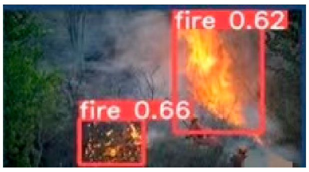
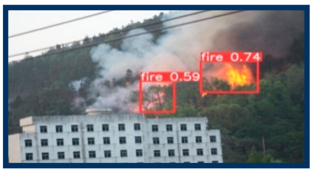
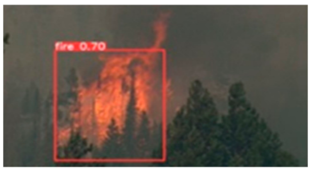
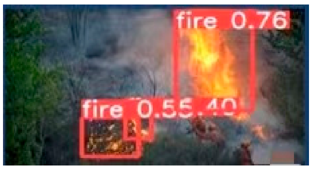
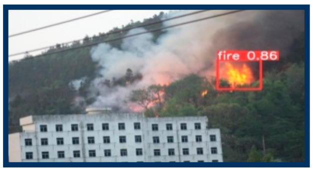
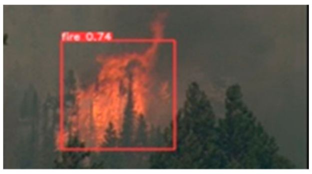
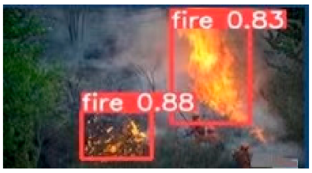
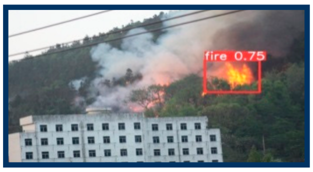
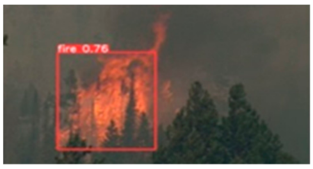
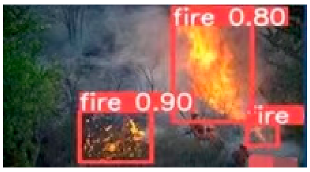
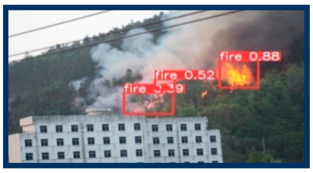
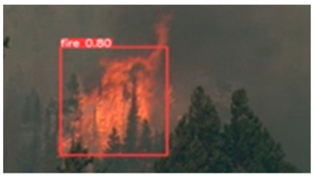
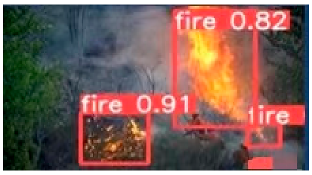


Figure 13. PR curves of 9 YOLO models.

Table 2. Comparison of evaluation metrics of each model.

Model	mAP (%)	Precision (%)	Recall (%)	Training Time	Flops (G)	F1-Score	Speed (ms/frame)
YOLOv3-tiny	89.33	79	90	1 h 18 m 36 s	5.448	0.84	
YOLOv3	94.06	84	93	2 h 11 m 24 s	65.304	0.88	
YOLOv4-tiny	80.47	86	79	1 h 56 m 24 s	6.787	0.82	
YOLOv4	94.54	70	81	6 h 52 m 12 s	127.232	0.76	
YOLOv5n	91.4	85.9	88.9	1 h 33 m 47 s	4.2	0.88	1.4
YOLOv5s	94.4	88.5	92.4	1 h 33 m 32 s	15.8	0.90	2.2
YOLOv5m	94.4	90.6	89.6	1 h 35 m 35 s	47.9	0.90	5.1
YOLOv5l	96.3	91	91.9	1 h 32 m 4 s	107.8	0.91	8.5
YOLOv5x	95.7	89.8	92	1 h 38 m 59 s	204	0.91	14.1

Table 3. Flame detection results for the YOLOv5 models.

Model	Detection Results		
YOLOv5n			
YOLOv5s			
YOLOv5m			
YOLOv5l			
YOLOv5x			

3.3. Flame Detection in Different Scenarios

Figure 14 shows the flame detection results of the YOLOv5n model in different scenarios. It can be seen from the images that YOLOv5n could simultaneously locate multiple flames taken by an UAV from high altitude, even if these flames were small objects (Figure 14a). Moreover, flames obscured by trees could be identified with a certain degree of confidence (Figure 14b). It also possessed a higher detection rate for the flame image affected by environmental interference (mainly smoke) (Figure 14c) or the flame image captured at low altitude (Figure 14d).

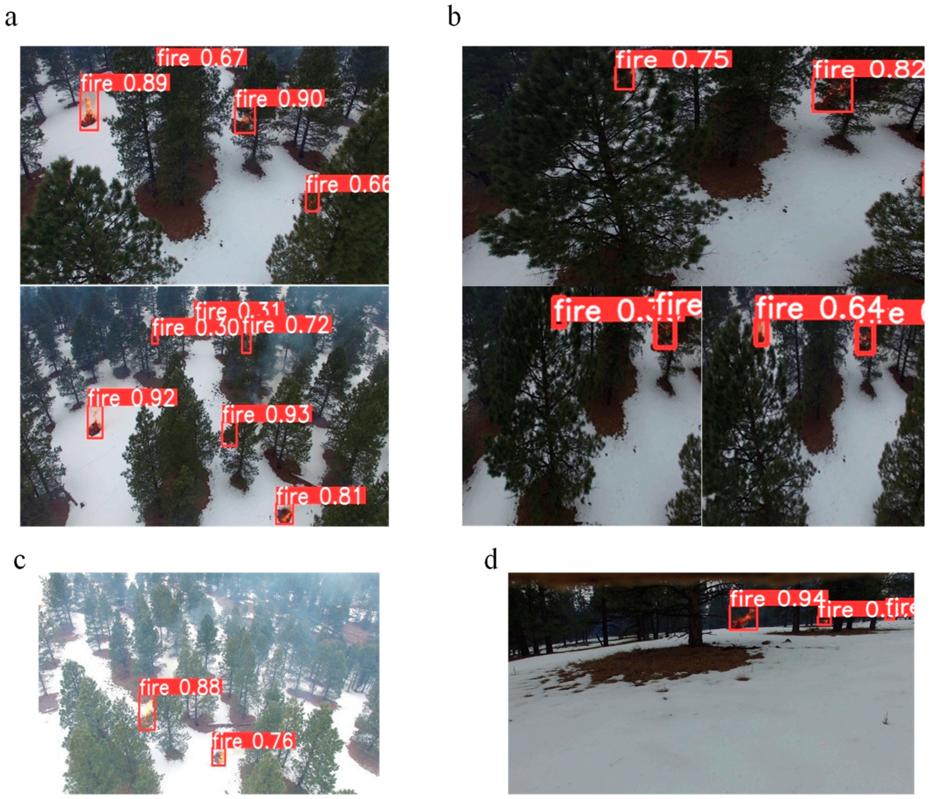


Figure 14. Flame detection diagrams by using YOLOv5n model in different scenarios based on FLAME datasets, including: (**a**) multiple small object flames taken from high altitude, (**b**) flames obscured by trees, (**c**) flames disturbed by environment (mainly smoke), (**d**) a close-up shot of the flames. Rectangles delineate the location and size of the object, with numbers representing confidence.

We further validated the performance and applicability of the YOLOv5n detector by using flame images obtained from the Internet, which were independent of the FLAME datasets on which the model was trained. The results in Figure 15 show that our model could identify flames that were significantly different from the training set (Figure 15a–c). Moreover, our model was reliable and it did not misjudge flame-like targets (Figure 15d,e).

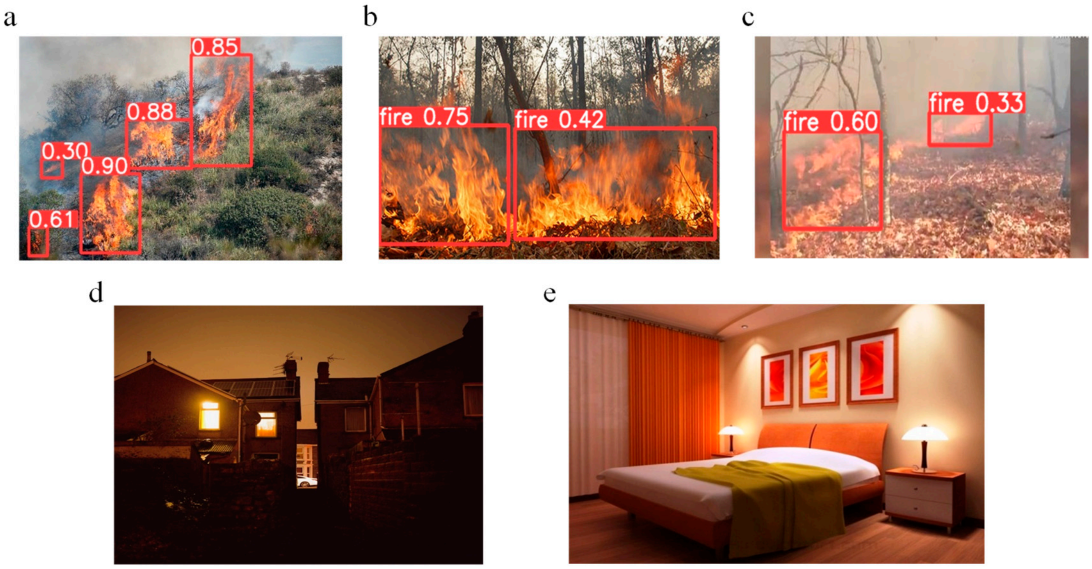


Figure 15. Flame detection with YOLOv5n model based on images from the Internet, including: (**a**) forest canopy flames, (**b**,**c**) woodland ground flames, (**d**,**e**) flame-like targets.

4. Discussion and Conclusions

Satellite remote sensing is good at detecting large-scale forest fires [42–46]. But at a small scale, UAVs equipped with ordinary optical cameras may be able to monitor forest fires in real-time [47,48]. In this study, YOLOv5 achieved the fast and accurate detection of flame objects in the UAV imagery, aiming to provide a feasible solution for the early prevention of forest fires.

YOLOv5 is the classic version of the YOLO architecture, which possess improved detection accuracy and inference speed compared to the earlier versions of YOLO [49]. In previous studies, machine learning algorithms could only extract shallow features for fire detection [50,51], while YOLOv5 can realize the depth extraction of features [52]. Moreover, YOLOv5 is more suitable for deployment in resource-constrained embedded devices [53,54]. Therefore, this study evaluated the application of YOLOv5 in fire detection based on UAV imagery. The results showed that the detection effect of YOLOv5 was better than that of YOLOv3 and YOLOv4. Some previous studies had taken the same view [55,56].

In the YOLOv5 network, we compared the performance of the YOLOv5s and YOLOv5n models with low computing resource requirements. The results showed that the mAP of the YOLOv5s and YOLOv5n models was 94.4% and 91.4%, respectively, and the detection speed was 2.2 ms and 1.4 ms per frame, respectively. It had been reported that YOLOv5s showed an excellent performance in forest fire detection [2]. However, by balancing the limited computing resources of UAVs against the need for real-time, fast, and accurate fire detection, we concluded that YOLOv5n was the best among the other models.

In contrast to other fixed objects, flames are a special object; they are diverse in size, shape, texture, and color [2]. In addition, forest fires are also affected by environmental interference or occlusion by trees. Therefore, forest fire detection has more restrictions and higher requirements of detection algorithms and training data than other types of object detection. The drone images we used came from the FLAME datasets, which are publicly available in Northern Arizona University. The datasets contained forest fire images of multiple objects, small objects, and sheltered objects in various complex situations, which was conducive to the adaptation of the detection model to different detection scenarios [57]. We will further enrich the datasets so that the detection network remains sensitive to different types of flame [58,59]. The forest fire images of different environments and disturbances will be brought into the model training to improve the generalization of the model. We will also try to improve the marking strategy of flames, as the quality of the training data affects the detection performance.

In the follow-up study, we will improve the accuracy of the YOLOv5n model within its original size or we will try to compress the model size of YOLOv5s without reducing its accuracy. Moreover, it is necessary to improve the data augmentation algorithm as the number of datasets limits the learning and extraction features. Most importantly, as the data are not analyzed on the UAV itself, we need to lightweight the detection network and deploy it on UAVs to achieve the real-time and high-precision detection of small flames in complex forest conditions.

Deploying a UAV solution could incur additional costs due to the necessity of performing flights in the forest for monitoring the conditions, especially in remote natural forests. In addition, the limited UAV memory is still a shortcoming [25–27]. But we believe that our study will provide a good attempt for the early prevention of forest fires.

Author Contributions: Writing—original draft preparation, H.L.; writing—review and editing, H.H.; visualization, F.Z.; supervision, H.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (grant number [22175062, 21671064]), the Natural Science Foundation of Hunan Province of China (grant number [2020JJ5446, 2021JJ30232]), the Scientific Research Fund of Hunan Provincial Education Department (grant number [19B448, 19B183, 20B209]), and the High-level-talent Initiation Research Fund of Hunan Police Academy (grant number [2021KYQD16]).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing not applicable.

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

References

1. Ghorbanzadeh, O.; Blaschke, T.; Gholamnia, K.; Aryal, J. Forest Fire Susceptibility and Risk Mapping Using Social/Infrastructural Vulnerability and Environmental Variables. *Fire* **2019**, *2*, 50. [CrossRef]
2. Xu, R.; Lin, H.; Lu, K.; Cao, L.; Liu, Y. A Forest Fire Detection System Based on Ensemble Learning. *Forests* **2021**, *12*, 217. [CrossRef]
3. Sharma, A.; Kumar, H.; Mittal, K.; Kauhsal, S.; Kaushal, M.; Gupta, D.; Narula, A. IoT and deep learning-inspired multi-model framework for monitoring Active Fire Locations in Agricultural Activities. *Comput. Electr. Eng.* **2021**, *93*, 107216. [CrossRef]
4. Mohindru, P.; Singh, R. Multi-sensor based forest fire detection system. *Int. J. Soft Comput. Eng.* **2013**, *3*, T707.
5. Choi, K.; Hong, S.; Kim, D.S.; Choi, D. An Experimental Study on the Optimum Installation of Fire Detector for Early Stage Fire Detecting in Rack-Type Warehouses. *World Acad. Sci. Eng. Technol. Int. J. Mech. Aerosp. Ind. Mechatron. Manuf. Eng.* **2017**, *11*, 757–764.
6. Rachman, F.Z.; Yanti, N.; Hadiyanto, H.; Suhaedi, S.; Hidayati, Q.; Widagda, M.E.P.; Saputra, B.A. Design of the early fire detection based fuzzy logic using multisensor. *IOP Conf. Ser. Mater. Sci. Eng.* **2020**, *732*, 012039. [CrossRef]
7. Nemalidinne, S.M.; Gupta, D. Nonsubsampled contourlet domain visible and infrared image fusion framework for fire detection using pulse coupled neural network and spatial fuzzy clustering. *Fire Saf. J.* **2018**, *101*, 84–101. [CrossRef]
8. Dong, Z.; Huang, D.G.; Zhang, D.Y. Research of an Automatic Forest Fire Detection System Based on Cooperative Perception. *Appl. Mech. Mater.* **2011**, *48-49*, 916–919. [CrossRef]
9. Hackner, A.; Oberpriller, H.; Ohnesorge, A.; Hechtenberg, V.; Müller, G. Heterogeneous sensor arrays: Merging cameras and gas sensors into innovative fire detection systems. *Sens. Actuators B Chem.* **2016**, *231*, 497–505. [CrossRef]
10. Kirani, Y.; Dey, D. Detection of Fire Regions from a Video image frames using YCbCr Color Model. *Int. J. Recent Technol. Eng. (IJRTE)* **2019**, *8*, 6082–6086.
11. Senthil, M. Implications of Color Models in Image Processing for Fire Detection. *Int. J. Comput. Appl.* **2018**, *179*, 38–41. [CrossRef]
12. Hu, G.L.; Jiang, X. Early Fire Detection of Large Space Combining Thresholding with Edge Detection Techniques. *Appl. Mech. Mater.* **2010**, *44–47*, 2060–2064. [CrossRef]
13. Gupta, A.; Bokde, N.; Marathe, D.; Kishore. A Novel approach for Video based Fire Detection system using Spatial and Texture analysis. *Indian J. Sci. Technol.* **2018**, *11*, 1–17. [CrossRef]
14. Prema, C.E.; 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. [CrossRef]
15. Negara, B.S.; Kurniawan, R.; A Nazri, M.Z.; Abdullah, S.N.H.S.; Saputra, R.W.; Ismanto, A. Riau Forest Fire Prediction using Supervised Machine Learning. *J. Phys. Conf. Ser.* **2020**, *1566*, 012002. [CrossRef]
16. Jain, T.; Sharma, N. Forest Fire Prediction using Machine Learning Models based on DC, Wind and RH. *Int. J. Recent Technol. Eng.* **2020**, *8*, 7–8. [CrossRef]
17. Chanthiya, P.; Kalaivani, V. Forest fire detection on LANDSAT images using support vector machine. *Concurr. Comput. Pract. Exp.* **2021**, *33*, e6280. [CrossRef]
18. Qiu, J.; Wang, H.; Shen, W.; Zhang, Y.; Su, H.; Li, M. Quantifying Forest Fire and Post-Fire Vegetation Recovery in the Daxin'anling Area of Northeastern China Using Landsat Time-Series Data and Machine Learning. *Remote Sens.* **2021**, *13*, 792. [CrossRef]
19. Abid, F. A Survey of Machine Learning Algorithms Based Forest Fires Prediction and Detection Systems. *Fire Technol.* **2020**, *57*, 559–590. [CrossRef]
20. Pratapa, A.; Doron, M.; Caicedo, J.C. Image-based cell phenotyping with deep learning. *Curr. Opin. Chem. Biol.* **2021**, *65*, 9–17. [CrossRef]
21. Ma, P.; Lau, C.P.; Yu, N.; Li, A.; Liu, P.; Wang, Q.; Sheng, J. Image-based nutrient estimation for Chinese dishes using deep learning. *Food Res. Int.* **2021**, *147*, 110437. [CrossRef]
22. 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*, 496–503. [CrossRef]
23. Pham, V.; Weindorf, D.C.; Dang, T. Soil profile analysis using interactive visualizations, machine learning, and deep learning. *Comput. Electron. Agric.* **2021**, *191*, 106539. [CrossRef]
24. Jiao, Z.; Zhang, Y.; Mu, L.; Xin, J.; Jiao, S.; Liu, H.; Liu, D. A YOLOv3-based Learning Strategy for Real-time UAV-based Forest Fire Detection. In Proceedings of the 2020 Chinese Control And Decision Conference (CCDC), Hefei, China, 22–24 August 2020; pp. 4963–4967. [CrossRef]
25. Li, D.; Sun, X.; Elkhouchlaa, H.; Jia, Y.; Yao, Z.; Lin, P.; Li, J.; Lu, H. Fast detection and location of longan fruits using UAV images. *Comput. Electron. Agric.* **2021**, *190*, 106465. [CrossRef]
26. Sarwar, F.; Griffin, A.; Rehman, S.U.; Pasang, T. Detecting sheep in UAV images. *Comput. Electron. Agric.* **2021**, *187*, 106219. [CrossRef]

27. Pandey, A.; Jain, K. An intelligent system for crop identification and classification from UAV images using conjugated dense convolutional neural network. *Comput. Electron. Agric.* **2022**, *192*, 106543. [CrossRef]
28. Kinaneva, D.; Hristov, G.; Raychev, J.; Zahariev, P. Application of Artificial Intelligence in UAV platforms for Early Forest Fire Detection. In Proceedings of the 27th National Conference with International Participation (TELECOM), Sofia, Bulgaria, 30–31 October 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 50–53. [CrossRef]
29. Barmpoutis, P.; Stathaki, T.; Dimitropoulos, K.; Grammalidis, N. Early Fire Detection Based on Aerial 360-Degree Sensors, Deep Convolution Neural Networks and Exploitation of Fire Dynamic Textures. *Remote Sens.* **2020**, *12*, 3177. [CrossRef]
30. Qin, Y.-Y.; Cao, J.-T.; Ji, X.-F. Fire Detection Method Based on Depthwise Separable Convolution and YOLOv3. *Int. J. Autom. Comput.* **2021**, *18*, 300–310. [CrossRef]
31. Redmon, J.; Divvala, S.K.; Girshick, R.B.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
32. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6517–6525.
33. Alexandrov, D.; Pertseva, E.; Berman, I.; Pantiukhin, I.; Kapitonov, A. Analysis of Machine Learning Methods for Wildfire Security Monitoring with an Unmanned Aerial Vehicles. In Proceedings of the 24th conference of open innovations association (FRUCT), Moscow, Russia, 8–12 April 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 3–9. [CrossRef]
34. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
35. Abdusalomov, A.; Baratov, N.; Kutlimuratov, A.; Whangbo, T.K. An Improvement of the Fire Detection and Classification Method Using YOLOv3 for Surveillance Systems. *Sensors* **2021**, *21*, 6519. [CrossRef]
36. Li, P.; Zhao, W. Image fire detection algorithms based on convolutional neural networks. *Case Stud. Therm. Eng.* **2020**, *19*, 100625. [CrossRef]
37. Dai, J.; Li, Y.; He, K.; Sun, J. R-FCN: Object Detection via Region-based Fully Convolutional Networks. *arXiv* **2016**, arXiv:1605.06409.
38. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *39*, 1137–1149. [CrossRef]
39. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.E.; Fu, C.; Berg, A.C. SSD: Single Shot MultiBox Detector. In Proceedings of the ECCV, Amsterdam, The Netherlands, 11–14 October 2016.
40. Bochkovskiy, A.; Wang, C.; Liao, H.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934.
41. Shamsoshoara, A.; Afghah, F.; Razi, A.; Zheng, L.; Fulé, P.Z.; Blasch, E. Aerial imagery pile burn detection using deep learning: The FLAME dataset. *Comput. Netw.* **2021**, *193*, 108001. [CrossRef]
42. Lee, W.; Kim, S.; Lee, Y.-T.; Lee, H.-W.; Choi, M. Deep neural networks for wild fire detection with unmanned aerial vehicle. In Proceedings of the 2017 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, USA, 8–10 January 2017; pp. 252–253. [CrossRef]
43. Szpakowski, D.M.; Jensen, J.L.R. A Review of the Applications of Remote Sensing in Fire Ecology. *Remote Sens.* **2019**, *11*, 2638. [CrossRef]
44. Barmpoutis, P.; Papaioannou, P.; Dimitropoulos, K.; Grammalidis, N. A Review on Early Forest Fire Detection Systems Using Optical Remote Sensing. *Sensors* **2020**, *20*, 6442. [CrossRef]
45. Farhadi, H.; Mokhtarzade, M.; Ebadi, H.; Beirami, B.A. Rapid and automatic burned area detection using sentinel-2 time-series images in google earth engine cloud platform: A case study over the Andika and Behbahan Regions, Iran. *Environ. Monit. Assess.* **2022**, *194*, 369. [CrossRef]
46. Liu, J.; Maeda, E.E.; Wang, D.; Heiskanen, J. Sensitivity of Spectral Indices on Burned Area Detection using Landsat Time Series in Savannas of Southern Burkina Faso. *Remote Sens.* **2021**, *13*, 2492. [CrossRef]
47. Kanga, S.; Singh, S.K. Forest Fire Simulation Modeling using Remote Sensing & GIS. *Int. J. Adv. Res. Comput. Sci.* **2017**, *8*, 326–332.
48. Yuan, C.; Liu, Z.; Zhang, Y. Aerial Images-Based Forest Fire Detection for Firefighting Using Optical Remote Sensing Techniques and Unmanned Aerial Vehicles. *J. Intell. Robot. Syst.* **2017**, *88*, 635–654. [CrossRef]
49. Yang, G.; Feng, W.; Jin, J.; Lei, Q.; Li, X.; Gui, G.; Wang, W. Face Mask Recognition System with YOLOV5 Based on Image Recognition. In Proceedings of the 2020 IEEE 6th International Conference on Computer and Communications (ICCC), Chengdu, China, 11–14 December 2020; pp. 1398–1404.
50. Peng, Y.; Wang, Y. Real-time forest smoke detection using hand-designed features and deep learning. *Comput. Electron. Agric.* **2019**, *167*, 105029. [CrossRef]
51. Pham, B.T.; Jaafari, A.; Avand, M.; Al-Ansari, N.; Dinh Du, T.; Yen, H.P.H.; Phong, T.V.; Nguyen, D.H.; Le, H.V.; Mafi-Gholami, D.; et al. Performance Evaluation of Machine Learning Methods for Forest Fire Modeling and Prediction. *Symmetry* **2020**, *12*, 1022. [CrossRef]
52. Fang, Y.; Guo, X.; Chen, K.; Zhou, Z.; Ye, Q. Accurate and automated detection of surface knots on sawn timbers using YOLO-V5 model. *Bioresources* **2021**, *16*, 5390. [CrossRef]
53. Zhao, J.; Zhang, X.; Yan, J.; Qiu, X.; Yao, X.; Tian, Y.; Zhu, Y.; Cao, W. A Wheat Spike Detection Method in UAV Images Based on Improved YOLOv5. *Remote Sens.* **2021**, *13*, 3095. [CrossRef]

54. Zhan, W.; Sun, C.; Wang, M.; She, J.; Zhang, Y.; Zhang, Z.; Sun, Y. An improved Yolov5 real-time detection method for small objects captured by UAV. *Soft Comput.* **2022**, *26*, 361–373. [CrossRef]
55. Kuznetsova, A.; Maleva, T.; Soloviev, V. Detecting Apples in Orchards Using YOLOv3 and YOLOv5 in General and Close-Up Images. In Proceedings of the International Symposium on Neural Networks, Cairo, Egypt, 4–6 December 2020; Springer: Cham, Switzerland, 2020; pp. 233–243.
56. Nepal, U.; Eslamiat, H. Comparing YOLOv3, YOLOv4 and YOLOv5 for Autonomous Landing Spot Detection in Faulty UAVs. *Sensors* **2022**, *22*, 464. [CrossRef]
57. Chen, B.-H.; Shi, L.-F.; Ke, X. A Robust Moving Object Detection in Multi-Scenario Big Data for Video Surveillance. *IEEE Trans. Circuits Syst. Video Technol.* **2018**, *29*, 982–995. [CrossRef]
58. Shorten, C.; Khoshgoftaar, T.M. A survey on Image Data Augmentation for Deep Learning. *J. Big Data* **2019**, *6*, 60. [CrossRef]
59. Zoph, B.; Cubuk, E.D.; Ghiasi, G.; Lin, T.-Y.; Shlens, J.; Le, Q.V. Learning Data Augmentation Strategies for Object Detection. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2020; pp. 566–583. [CrossRef]

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