forests

MDPI

Article

Research on Forest Flame Detection Algorithm Based on a Lightweight Neural Network

Yixin Chen, Ting Wang * and Haifeng Lin

College of Information Science and Technology & College of Artificial Intelligence, Nanjing Forestry University, Nanjing 210037, China; cyx10405@njfu.edu.cn (Y.C.); haifeng.lin@njfu.edu.cn (H.L.)
* Correspondence: chunchun1010@163.com; Tel.: +86-25-8542-7827

Abstract: To solve the problem of the poor performance of a flame detection algorithm in a complex forest background, such as poor detection performance, insensitivity to small targets, and excessive computational load, there is an urgent need for a lightweight, high-accuracy, real-time detection system. This paper introduces a lightweight object-detection algorithm called GS-YOLOv5s, which is based on the YOLOv5s baseline model and incorporates a multi-scale feature fusion knowledge distillation architecture. Firstly, the ghost shuffle convolution bottleneck is applied to obtain richer gradient information through branching. Secondly, the WIoU loss function is used to address the issues of GIoU related to model optimization, slow convergence, and inaccurate regression. Finally, a knowledge distillation algorithm based on feature fusion is employed to further improve its accuracy. Experimental results based on the dataset show that compared to the YOLOv5s baseline model, the proposed algorithm reduces the number of parameters and floating-point operations by approximately 26% and 36%, respectively. Moreover, it achieved a 3.1% improvement in $mAP_{0.5}$ compared to YOLOv5s. The experiments demonstrate that GS-YOLOv5s, based on multi-scale feature fusion, not only enhances detection accuracy but also meets the requirements of lightweight and real-time detection in forest fire detection, commendably improving the practicality of flame-detection algorithms.

Keywords: forest flame detection; inter-stage local network; loss function; feature fusion; knowledge distillation

Citation: Chen, Y.; Wang, T.; Lin, H. Research on Forest Flame Detection Algorithm Based on a Lightweight Neural Network. *Forests* **2023**, *14*, 2377. https://doi.org/10.3390/f14122377

Academic Editor: George P. Petropoulos

Received: 5 November 2023
Revised: 27 November 2023
Accepted: 30 November 2023
Published: 5 December 2023

1. Introduction

A forest is an important resource and environmental condition for human beings. It shoulders many functions, such as windbreak and sand fixation, water conservation, soil and water conservation, climate regulation, environment beautification, oxygen release, air purification, and noise reduction. Forest fires have always been a major direct threat to forests, and they are also the focus of forestry monitoring.

However, the traditional forest fire monitoring is mainly through manual inspection, which has obvious shortcomings, such as low efficiency, high cost, and difficulties in achieving all-weather duty. With the development of sensor technology, most fire-detection systems rely on sensors [1–4], such as smoke and temperature sensors. However, limitations, such as the restricted installation range, high costs, and inability to provide crucial visual information constrain the use of sensors in fire detection.

With the development of computer vision technology, image processing techniques have become widely applied in fire detection. Image detection has the advantages of a short detection time, high accuracy, and flexible installation, so it can be fitted on UAV-equipped drones for real-time detection. As the altitude of the UAV is constantly changing, the size of the flame image captured is constantly changing. When flying at high altitudes, a single image may contain multiple small targets. Sometimes the image contains complex background information, including the detection of objects that are obscured and the presence of objects that are easily misdetected. These problems will lead to a lower accuracy of

object detection. Object detection is one of the fundamental tasks in computer vision image analysis and can be broadly categorized into traditional object-detection methods and deep learning-based object detection methods. Traditional object-detection methods involve selecting regions that may contain objects, followed by feature extraction from the chosen regions, such as SIFT, Harr, and HOG, and then detecting and classifying the extracted features. In contrast, deep learning-based object-detection methods, as a newer approach, can be divided into two-stage object-detection methods and one-stage object-detection methods. Compared to traditional methods, deep learning-based methods have higher accuracy and recall rates. Two-stage methods initially identify candidate regions and then classify the objects within those regions while also determining their position. Typical two-stage models include R-CNN [5] (Regions with CNN feature), Faster R-CNN [6], R-FCN, and Cascade R-CNN [7]. While two-stage detectors provide accurate results, they may not meet real-time requirements in practical applications. Hence, one-stage object-detection methods have been developed. One-stage algorithms do not independently extract candidate regions but directly provide object category probabilities and position coordinates from input images in a single stage, resulting in faster detection speeds. Prominent one-stage detectors include EfficientNet [8], EfficientDet [9], SSD [10], and the YOLO [11] series.

With the advancement of deep learning technology, an increasing number of researchers are employing deep learning for forest fire detection, where convolutional neural networks (CNNs) find widespread application in forest fire recognition and localization. Ding [12] proposed an improved flame recognition color space (IFCS) based on chaos theory and k-medoids particle swarm optimization algorithm. The multi-layer algorithm developed by Mondal [13] et al. takes color-based cues for detection into account, combining three filtering stages, "centroid analysis", "histogram analysis", and "variance analysis", to successfully detect fires. Huang [14], proposed a lightweight forest-fire-detection method using a YOLO-based dehazing algorithm. They obtained haze-free images using a dark channel prior before dehazing and improved YOLOX-L through techniques, such as GhostNet, depthwise separable convolution, and SENet [15], applying it to haze-free image-based forest fire detection. Sun [16] employed a lightweight backbone network called Squeeze and Excitation-GhostNet (SE-GhostNet) for feature extraction, making it easier to distinguish a forest fire from smoke within the background while significantly reducing model parameters. Zhou [17] introduced a cosine annealing algorithm, label smoothing, and multi-scale training to improve the detection accuracy of the model. Lu [18] proposed a multi-task learning-based forest-fire-detection model (MTL-FFDet), which contains three tasks (the detection task, the segmentation task, and the classification task) and shares the feature extraction module. Additionally, Huang [19] introduced an improved early forest-fire-smoke-detection model based on deformable transformers, featuring optimal sparse spatial sampling capabilities for smoke, involving deformable convolutions and transformer-based relationship modeling.

Neural network algorithm models have continuously improved in performance, with network structures moving towards greater depth and width. However, this trend requires more computational power, which leads to substantial computational and memory costs. This has significantly constrained algorithm development. Therefore, the design of lightweight models has become a pressing need. Lightweight models aim to reduce algorithm complexity without sacrificing performance excessively. Representative approaches include parameter pruning and quantization, low-rank decomposition, and knowledge distillation, which have opened avenues for model lightweighting. To meet the real-time requirements of forest fire detection while avoiding high costs, some small-scale lightweight networks have emerged, such as Xception [20], MobileNet [21], and ShuffleNets [22]. These networks have significantly improved the detection speed through sparse convolution operations but tend to exhibit lower accuracy when applied to forest fire detection. In the realm of compressing entire networks, methods, like network pruning and quantization, have matured. Pruning eliminates redundant parameters in trained networks to reduce model parameters and prevent overfitting, whereas quantization compresses network parameters

to approximate the original model with fewer bits, reducing the model size. However, these methods often depend on specific hardware and customized algorithm implementations. In addition to the aforementioned methods, knowledge distillation, which has gained widespread attention in recent years, offers a unique approach. Knowledge distillation is a form of knowledge transfer that involves having a smaller model learn from the outputs of a larger model, simulating knowledge transfer. Knowledge distillation initially emerged from transfer learning. Gou [23] devised a new knowledge distillation framework called multi-target knowledge distillation via student self-reflection or MTKD-SSR, which can not only enhance the teacher's ability in unfolding the knowledge to be distilled, but also improve the student's capacity of digesting the knowledge. Meanwhile, Yuan [24] constructed the conceptual model and theoretical analysis framework of the influence mechanism of the knowledge network arrangement mechanism on knowledge distillation in 2022. Zou [25] proposed a multi-scale feature extraction method using channel-wise split-concatenation to enhance feature mapping's multi-scale representation ability. Li [26] introduced a selective feature fusion module, resulting in a new form of self-distillation called knowledge fusion distillation. Zhao [27] proposed a Relationship-Prototype Network (RPNet) that combines the features of ProtoNet and RelationNet, using prototype distance and non-linear relationship scores for classification.

This paper addresses the limitations of existing forest-fire-detection algorithms, which exhibit poor detection performance in complex backgrounds and insensitivity to small targets. To address these issues, the GS-YOLOv5s algorithm was designed. The original Cross-Stage Local Network (CSL) module had a number of parameters and insufficient feature extraction capability. Therefore, this paper modified the original CSL module, building upon the GS bottleneck and leveraging parallel branches to extract richer gradient information. In addition, to achieve a more lightweight model suitable for deployment on embedded devices, traditional convolutions in the neck network were replaced with GSConv [28]. To enhance the model's performance, the study introduced the Weighted Intersection over Union (WIoU) [29] loss function based on a focusing mechanism. This was performed to address the limitations of the original Generalized Intersection over Union (GIoU) [30] loss function, which failed to accurately reflect the distance and overlap between the prediction box and ground truth box. This improvement helps with overall model optimization, convergence speed, and regression accuracy. Finally, a feature fusion knowledge distillation process was employed, using YOLOv5x as the "teacher network" to distill the enhanced model. This not only achieved model lightweighting, but also improved the detection accuracy.

The remaining sections of this paper are organized as follows. In Section 2, we introduce the dataset used in this study and outline the model evaluation metrics; a description of the proposed GS-YOLOv5s algorithm model is also detailed. In Section 3, a comprehensive description of the experimental setup, including the equipment and experimental parameters is provided, and the effectiveness of the proposed modules is validated. Section 4 provides an explanation and analysis of the overall experiments conducted. We also summarize the work with the most significant quantitative obtained results.

2. Materials and Methods

2.1. Dataset

Datasets play a crucial role in object-detection research, particularly in tasks based on deep learning, where a large forest fire dataset is urgently required to train efficient detectors. In this study, we first obtained fire images from various scenes by developing web scraping scripts. In addition, we captured images of small forest fire targets under natural lighting conditions using UAVs. Subsequently, the dataset was manually annotated and converted into COCO format. In total, we collected and annotated 6200 high-quality forest-fire-detection images from different forest fire scenarios. Out of these, 4960 images were allocated for training, and 1240 images were used for testing. At the same time, a certain proportion of negative samples were set in it, such as vehicle lights being also

included, streetlights, sunlight, and forest firefighter clothing. According to the Microsoft COCO standard, objects smaller than the size of 32 × 32 pixels were defined as small targets. The representative samples is shown in Figure 1.

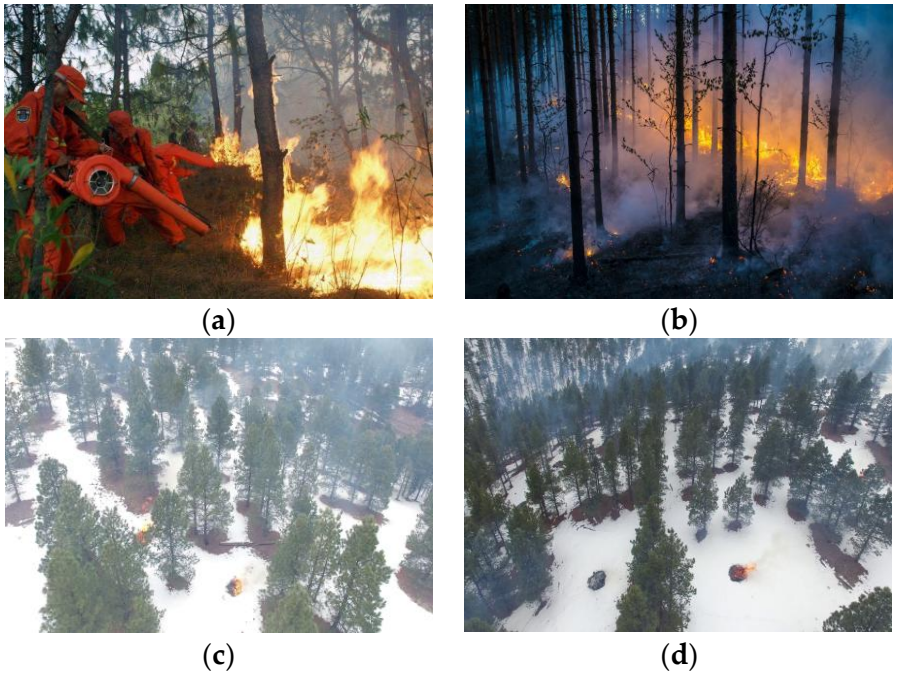


Figure 1. Sample images from the training set: (**a**–**d**) forest fire targets of different backgrounds and sizes.

2.2. YOLOv5

As a representative method in one-stage object detection, the YOLO series of networks is an end-to-end convolutional neural network capable of directly predicting the category and position of objects. YOLO divides the input image into S × S grids, with each grid responsible for detecting objects for which the center falls within it. Each grid predicts two bounding boxes, each represented by a five-dimensional vector: (x, y, w, h, c), where (x, y) denotes the center coordinates of the bounding box, w and h represent the width and the height, and c represents confidence. YOLOv5, compared to YOLOv3 and YOLOv4, is a smaller model that is more suitable for mobile applications. YOLOv5 includes five variants: YOLOv5n, YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x, with increasing model sizes. Taking factors, such as the model size, detection speed, and computational complexity, into account, we chose YOLOv5s as the baseline model for this study. The structure of the YOLOv5s detector is illustrated in Figure 2 and consists of three parts: the backbone, neck, and head. The backbone network extracts features from the input, the neck enhances these features, and the head performs classification and regression based on the extracted features. The local network module was designed based on the CSPNet structure for feature extraction. The Spatial Pyramid Pooling Fusion (SPPF) module is employed to capture the global information of detection targets, concatenating the outputs of three 5 × 5 max-pooling layers before applying a channel-wise split (CBS) operation. The neck network enhances the features extracted by the head network, and the head network's three branches are responsible for detecting large, medium, and small objects.

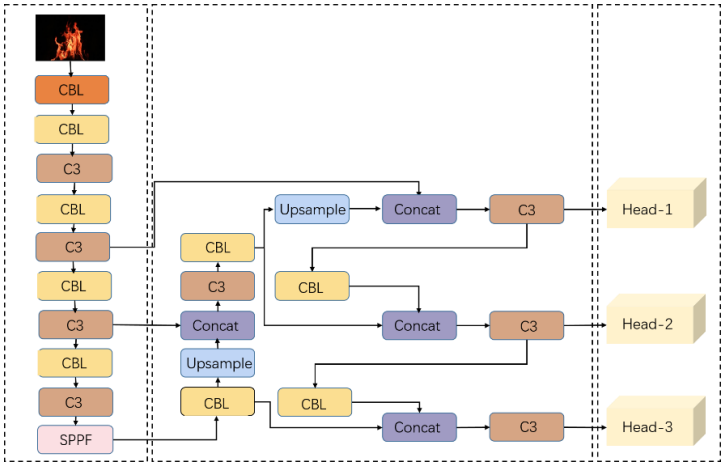
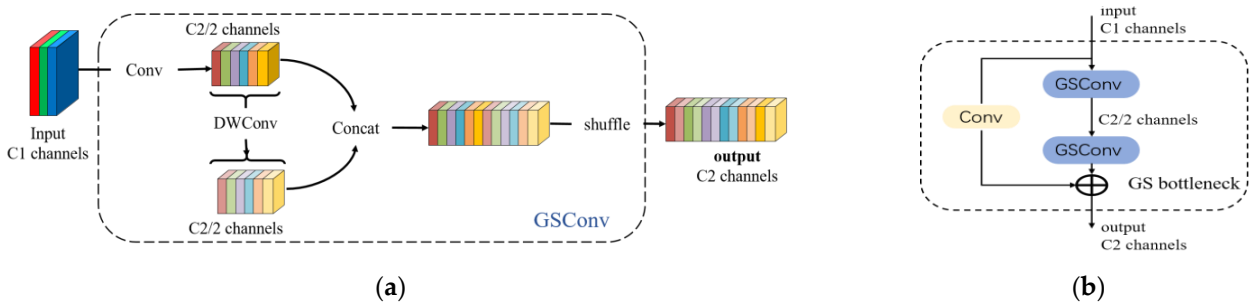


Figure 2. The figure presents a schematic diagram illustrating the structure of the YOLOv5s model. From left to right, the diagram showcases three dashed boxes representing the backbone network, neck network, and head network, respectively. In this diagram, the "CBL" module denotes the combination of convolution, batch normalization, and the Leaky-ReLU activation function. Furthermore, the "C3" module corresponds to a local network consisting of three convolutional structures operating across stages.

2.3. GS-C2

CNNs have demonstrated excellent performance in various computer vision tasks. However, traditional CNNs often require a large number of parameters and computational resources to achieve satisfactory accuracy. Moreover, during the process of feature extraction from images, convolutional neural networks often suffer from the issue of losing semantic information. Dense convolution computations maximize the preservation of hidden connections between each channel. In contrast, existing mainstream lightweight convolutions, such as sparse convolutions, hardly preserve these connections, making it challenging to achieve both model lightweighting and high accuracy. GSConv, with lower time complexity, strives to retain these connections as much as possible and incurs lower computational costs during data reshuffling operations. Data reshuffling is a uniform mixing strategy that allows for information from dense convolutions to be fully integrated into the output of sparse convolutions. It evenly exchanges local feature information across different channels without the need for fancy functionalities. For lightweight detectors, GSConv's advantages become more pronounced. It benefits from the addition of channel-sparse convolution kernels and data reshuffling to enhance nonlinear expression capability. First, a significant number of 1×1 dense convolutions are used to merge independently computed channel information. Second, "channel shuffling" is employed to facilitate channel interaction. Finally, data reshuffling infiltrates the information generated by dense convolution operations into every part of the information generated by sparse convolutions. The GSConv module is depicted in Figure 3a, where "DWConv" represents sparse convolution operations, "shuffle" indicates data reshuffling, C1 represents the number of channels for each convolution kernel, which is also the number of channels in the input feature map, and C2 represents the number of channels in the output feature map. However, if GSConv is applied throughout all stages of the model, the network's depth will increase, leading to higher resistance to data flow and significantly increasing the inference time. Therefore, using GSConv for feature map concatenation in the neck network, where the feature channel dimensions are maximized and the width is minimized, strikes the right balance and reduces redundant information. After concatenating GSConv, it is combined in parallel with standard convolutions. This structure is referred to as GS bottleneck, as shown in Figure 3b.



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

Figure 3. (**a**) GSConv module. (**b**) GS bottleneck module.

It consists of three standard convolution layers and multiple bottleneck modules in YOLOv5s. The C3 module is the primary component for feature learning. By applying the GS bottleneck in a parallel manner to the Cross-Stage Local (CSL) network module, the module structure depicted in Figure 4 is obtained. Compared to the original C3 module in YOLOv5s, which contains three convolutions, incorporating the GS bottleneck reduces one convolution layer. In this study, the module that includes the GS bottleneck with two convolutions is referred to as GS-C2, which not only ensures lightweighting but also provides richer gradient information. GS-C2 aims to improve the detection performance by introducing multiple branches into the network to capture information of different levels and scales simultaneously. This multi-branching design can better adapt to the changes in different scenes and targets, thus improving the accuracy and robustness of target detection.

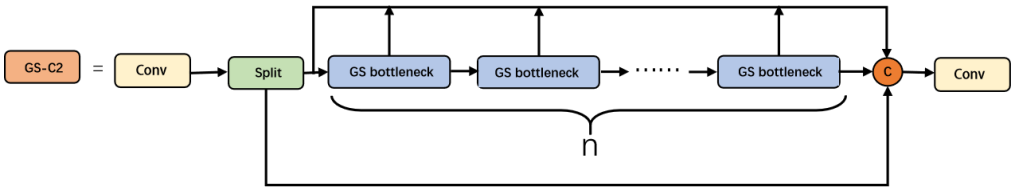


Figure 4. GS-C2 module.

2.4. Boundary Box Regression Loss Based on Focusing Mechanism—WIoU

The role of the loss function in a neural network is of paramount significance, as it serves to quantify the difference between actual and anticipated values at the culmination of each iteration; its overarching purpose lies in shepherding subsequent training phases towards the correct trajectory. The discerning choice of an appropriate loss function expedites the model's convergence during the training regimen. Within the domain of YOLOv5s, the spectrum of losses encompasses classification loss, regression loss, and confidence loss. Particularly noteworthy is the utilization of the GIoU (Generalized Intersection over Union) loss function as a replacement for the original IoU loss function in the realm of regression. Let us denote the predicted box as $\vec{B} = [x, y, w, h]$ and the target box as $\vec{B}_{gt} = [x_{gt}, y_{gt}, w_{gt}, h_{gt}]$. The formulas for IoU and GIoU Loss functions are expressed in Equations (1) and (2), respectively.

$$L_{IoU} = 1 - \mathrm{IoU} = 1 - \frac{W_i H_i}{S_u} \tag{1}$$

$$L_{GIoU} = 1 - \frac{W_i H_i}{S_u} + \frac{W_g H_g - S_u}{W_g H_g} \tag{2}$$

where W_g and H_g represent the width and height of the minimum enclosing box. Meanwhile, the width and height of the predicted box and ground truth box are denoted as w, h and w_{gt}, and h_{gt}. In addition, W_i and H_i represent the width and height of the predicted and ground truth box overlapping each other, respectively, as depicted in Figure 5. Notably,

the predicted and ground truth box are visually represented by the red and green box, with the area of the union being $S_u = wh + w_{gt}h_{gt} - W_iH_i$.

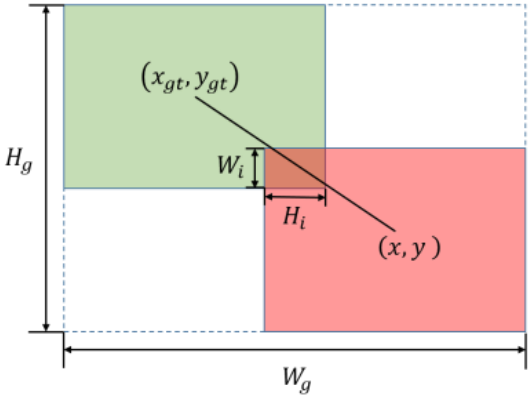


Figure 5. Map of the degree of overlap between the ground truth box (green) and the predicted box (red), where the blue dashed box represents the minimum closed box.

The GIoU loss function has limitations. Notably, when the predicted box is entirely contained within the ground truth box, the GIoU loss function fails to indicate the relative positional relationship between the predicted box and ground truth box, as illustrated in Figure 6. In such cases, the minimum enclosing region corresponds to the ground truth box, as represented by $\frac{W_gH_g - S_u}{W_gH_g} = 0$. The GIoU loss function degenerates into the IoU loss function, which fails to capture the distance and overlap degree between the predicted box and ground truth box. Consequently, this limitation poses challenges for optimizing the overall model, leading to slow convergence rates and inaccurate regression. To address these issues, this paper proposes the use of a Loss function augmented with a focused penalty mechanism R_{WIoU}.

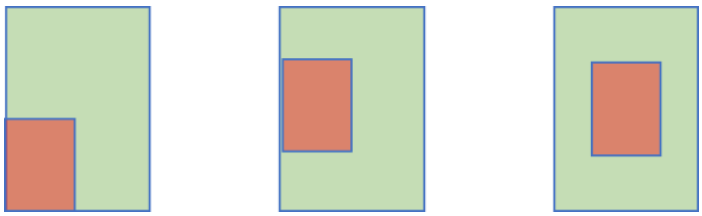


Figure 6. The performance of GIoU when the prediction box is completely contained within the target box; the red box represents the prediction box, and the green box represents the real box.

The WIoU (Weighted Intersection over Union) is an enhanced approach based on the intersection merge ratio. The WIoU loss function places particular emphasis on the significance of image boundaries, thereby effectively mitigating the issue of unclear image boundaries in detection tasks. It computes the WIoU loss function by comparing the predicted results with the real labels, calculating the intersection and union between them, and subsequently dividing the intersection by the union to obtain the IoU value. Distinguishing itself from the conventional IoU loss function, the WIoU loss function assigns weights to the IoU value, prioritizing the target boundary. The penalty term R_{WIoU} and Loss function L_{WIoU} expressions for WIoU are provided below:

$$R_{WIoU} = \exp\left(\frac{(x - x_{gt})^2 + (y - y_{gt})^2}{\left(W_g^2 + H_g^2\right)^*}\right) \tag{3}$$

$$L_{WIoU} = R_{WIoU} \times L_{IoU} \tag{4}$$

To prevent R_{WIoU} from generating gradients that hinder convergence, W_g and H_g are separated from the calculation in dynamic non-monotonic frequency modulation (the superscript $\times$ indicates this operation).

WIoU loss pays more attention to those that are difficult to estimate by introducing an attention mechanism that assigns different weights to samples of different difficulties. This helps improve the model's performance in difficult situations. WIoU is calculated in a way that involves the intersection and union of target and prediction boxes, where the weight of the intersection part is proportional to the size of the intersection area according to the attention mechanism. In this way, more attention is paid to the parts of target boundaries that are difficult to estimate, and the learning effect of the model on these boundaries is improved. In summary, WIoU is a bounding box regression loss that combines cross-entropy loss and attention mechanism to improve the fitting ability of target detection models to difficult-to-estimate samples. Such a design may achieve better performance based on some complex scenarios and difficult samples.

2.5. Multi-Scale Feature Fusion Knowledge Distillation

Distillation, as a specialized knowledge-transfer algorithm, allows for the transfer of knowledge from larger models to smaller ones. Typically, the larger model is referred to as the "teacher", while the smaller one is called the "student", and this network structure is known as the teacher–student network. Feature-based knowledge distillation, proposed by Romero [31] et al., has demonstrated that distillation is an effective method for avoiding redundant parameters in large models and improving the model inference speed.

The feature-based distillation algorithm, in order to ensure the consistency of target features with the features to be trained, creatively introduced adaptation layers to adjust the scale of the original features. The application of multi-scale feature fusion in knowledge distillation can help students model better learning and utilizing information of different scales. Eventually, the distillation loss is computed through a distance measurement between the two. During the distillation process in this study, it was discovered that fusing feature details from different levels significantly enhances model performance, and the efficiency of distillation improves significantly with the incorporation of feature fusion. The fusion of feature information with differing semantic scales aims to address issues related to limited semantic richness and incomplete information representation in the features. In the field of object detection, fusing high-level semantic features with low-level features and making predictions based on larger feature scales effectively alleviates the problem of inadequate representation capacity in low-level features, thus enhancing the network's ability to detect small target objects. The Feature Pyramid Network (FPN) is an early feature fusion network in the field of object detection. It sequentially fuses high-level features through processes, such as sampling, convolution, and addition, and then makes predictions on features of different scales to detect objects of different sizes, ultimately obtaining detection results on different scales. This paper investigates a unique cross-scale feature fusion approach and optimizes traditional feature knowledge distillation by adding the FPN [32] feature fusion structure. FPN usually consists of bottom-up and top-down paths, which can extract rich semantic information based on feature maps with different resolutions. In this way, feature knowledge distillation combined with FPN enables student models to better learn and utilize feature information from different scales, thereby improving the model's performance in tasks, such as target detection, as shown in Figure 7.

In particular, knowledge distillation primarily leverages the differences in semantic information expression between features on different scales. It utilizes the student network to learn the shallower features from the teacher network. Learning these shallow features at the beginning of training helps the student network better fit the training data. Additionally, due to the deep architecture of the network and to reduce model parameters effectively, a progressive learning strategy is employed. This strategy involves gradually fusing

features from higher layers to lower layers before proceeding with distillation from the teacher network.

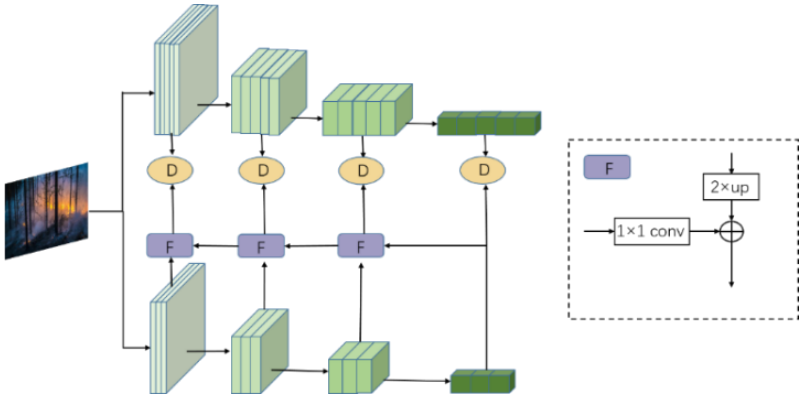


Figure 7. Structure diagram of the improved feature knowledge distillation network, where F is the feature fusion module.

2.6. Overall Architecture of GS-YOLOv5s

Flame detection imposes high requirements for the real-time performance and lightweight nature of detection models. According to the training data of the current mainstream target detection model, this paper takes into consideration the accuracy, efficiency, and scale of the detection model and makes improvements based on the YOLOv5s architecture. The enhanced model also performs well in small object detection. As this paper's primary enhancement is based on GSConv, the model is named GS-YOLOv5s.

The overall architecture of the designed optimization algorithm is illustrated in Figure 8. The backbone network consists of CBL, GS-C2, and the Spatial Pyramid Pooling Module (SPPF). The GS-C2 module incorporates the GS bottleneck into CSPNet, allowing for richer gradient information through additional branch-level cross-layer connections, ultimately leading to improved detection accuracy.

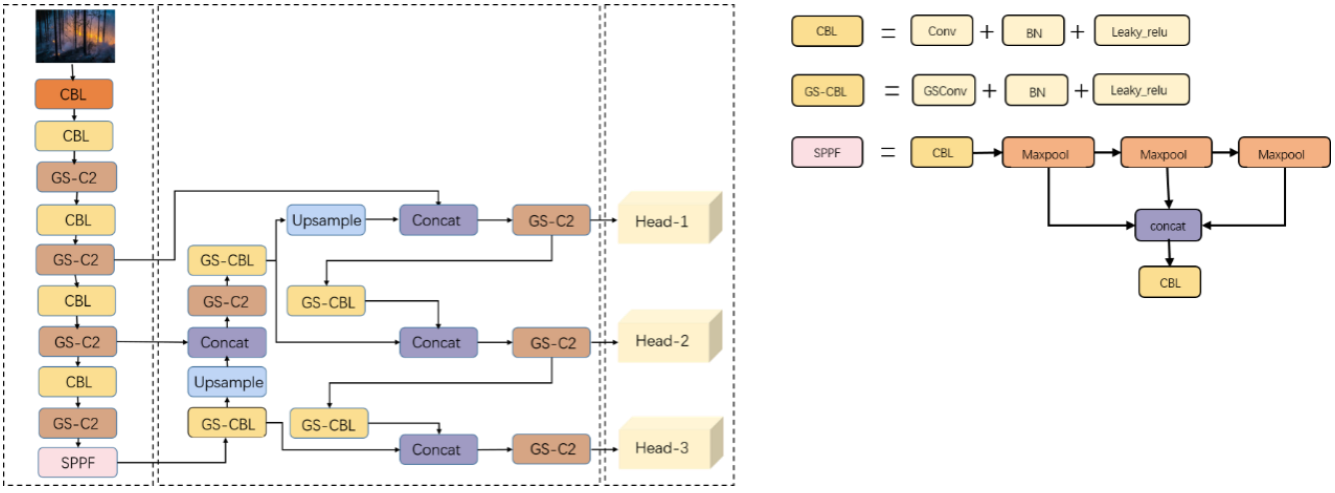


Figure 8. The overall architecture of the GS-YOLOv5 model, as well as the structural diagrams of the "CBL", "GS-CBL", and "SPPF" modules in the model.

In the neck region, information transmission and the fusion of deep and shallow-level feature information are achieved through upsampling, facilitating a top-down information transfer structure. Concatenation operations are performed between deep-level and shallow-level features, enabling the seamless passage of high-resolution features from the deep layers to the shallow layers, thereby implementing the PANet structure. This

effectively leverages the complementary advantages of multi-scale features, enhancing the accuracy of target recognition. Finally, the target is classified and regressed using the head network. Comparing Figure 8 with the original YOLOv5s model, it is evident that Figure 8 replaces standard convolutions and C3 modules with the lightweight GSConv module and GS-C2 module for feature extraction. Additionally, considerations include replacing the original GIoU loss function with the WIoU loss function to measure localization loss and employing YOLOv5x as the "teacher model" for overall knowledge distillation of the improved model's feature fusion.

2.7. Model Evaluation

In this field, common metrics for the accuracy assessment include the precision, recall, and mean average precision (*mAP*), whereas lightweight evaluation metrics encompass parameters and floating-point operations (FLOPs) as indicators of model complexity. The specific descriptions and formulas are provided below. Recall is the ratio of the number of true positive samples correctly detected to the total number of positive samples. Precision represents the ratio of the number of true positive samples correctly detected to the total number of samples detected. The *F*1 score is the harmonic average of precision and recall, which takes into account the precision and recall of the model. *mAP* was utilized for quantitatively evaluating detection accuracy and serves as a critical indicator for assessing the overall model performance. The formulas for precision, recall, *F*1 and *mAP* are as follows:

$$P = \frac{TP}{TP + FP} \tag{5}$$

$$R = \frac{TP}{TP + FN} \tag{6}$$

$$F1 = 2\frac{P * R}{P + R} \tag{7}$$

$$mAP = \frac{1}{n}\sum_{k=1}^{k=n} AP_k \tag{8}$$

In this context, *TP* represents the correct detection of forest fires, *FP* indicates instances where forest fires were not detected when they were present, and *FN* represents cases where the algorithm mistakenly detects forest fires in the absence of an actual forest fire. *P* stands for precision, which calculates the ratio of true positives to all samples predicted as positive, while *R* represents recall, which is the ratio of true positives to all actual positive samples. *mAP* is a fundamental parameter for assessing the accuracy of a network model's training, calculated as the area under the *PR* curve. The number in the lower-right corner of the *mAP* represents the IoU threshold when positive samples are considered detected; for example, $mAP_{0.5}$ indicates detection when the detection probability is greater than 0.5.

FLOPs, short for floating-point operations, are a measure of the computational workload and can be used to assess the complexity of algorithms or models. Parameter count refers to the total number of parameters during the model training process. Equations (9) and (10) show the formulas for calculating the floating-point operations and parameters.

$$FLOPs = k \cdot (H - s + 1) \cdot (W - s + 1) \cdot c \cdot k^2 \tag{9}$$

$$\text{parameters} = (s \cdot s \cdot n + 1) \cdot \text{c} \tag{10}$$

Among these variables, *H* and *W* denote the height and width of the input feature map, while *k* represents the size of the convolutional kernel. The parameter *c* signifies the number of output channels, *n* indicates the number of input channels, and an additional 1 is employed to represent the convolutional layer's offset.

3. Results

3.1. Training

When the dataset was collected, it was divided into training and testing sets in an 8:2 ratio. The software supplier for PyCharm (2021.2.3) is JetBrains, a company based in the Czech Republic with its headquarters located in Prague. The experimental conditions for training are listed in Table 1.

Table 1. Experimental conditions.

Experimental Environment	Details
Software	Pycharm
Programming Language	Python 3.9
Operating System	Windows 10
Deep Learning Framework	Pytorch 1.8.2
GPU	NVIDIA 3080ti

Hyperparameters can affect the structure and training process of neural networks, thereby affecting the performance of the model. YOLOv5 uses PyTorch as a deep learning framework, so hyperparameter adjustment is usually achieved by modifying the parameters in the training script. The batch-size and learning rate are gradually adjusted through continuous experimentation and observations of model performance. At the same time, when the epochs reach 200, the experimental results tend to converge. Considering the experimental period, 400 is selected as the epochs in this paper. The training parameters for the forest-fire-detection model are specified in Table 2.

Table 2. Training parameter settings.

Training Parameters	Details
Epochs	400
Batch-size	8
Img-size (pixels)	640 × 640
Initial Learning rate	0.01
Optimization algorithm	SGD

3.2. Ablation Experiment

This section validates the effectiveness of GS-YOLOv5s and assesses the impact of each component on the final performance through ablation experiments conducted on the custom dataset, as shown in Table 3. Taking YOLOv5s as the baseline model, the GS-C2 module, GSConv, WIoU loss function, and feature fusion knowledge distillation module were added successively. Initially, the model was trained using the training and testing sets. The results of the ablation experiments are summarized in Table 3.

Table 3. Data of ablation experiments.

MODEL	$mAP_{0.5}$	$mAP_{0.5:0.95}$	Parameters (M)	FLOPs (G)
YOLOv5s + GS-C2	0.846	0.496	5.6	12.1
YOLOv5s + GSConv	0.863	0.513	6.6	15.4
YOLOv5s + WIoU	0.848	0.519	7.0	15.9
YOLOv5s + Feature fusion knowledge distillation	0.857	0.519	7.0	15.9
YOLOv5s + GS-C2 + GSConv	0.859	0.516	5.2	10.2
YOLOv5s + GS-C2 + GSConv + WIoU	0.861	0.515	5.2	10.2
YOLOv5s + GS-C2 + GSConv + WIoU + Feature fusion knowledge distillation	0.872	0.516	5.2	10.2

In addition to the ablation experiments, a series of comparative experiments is also conducted to compare the detection results of GS-YOLOv5s proposed in this paper with popular one-stage and two-stage detection models, such as Faster R-CNN, SSD, and the YOLO series. The primary experimental metrics of interest included $mAP_{0.5}$, $mAP_{0.5:0.95}$, parameters, and the floating-point operations. Detailed experimental results are presented in Table 4 and are also visualized in scatterplot form in Figure 9.

Table 4. Detection results of different methods based on the dataset.

MODEL	P	R	F1	$mAP_{0.5}$	$mAP_{0.5:0.95}$	Parameters (M)	FLOPs(G)
YOLOv5n	0.813	0.712	0.759	0.823	0.497	1.765	4.2
YOLOv5s	0.875	0.762	0.798	0.838	0.516	7.022	15.9
YOLOv5m	0.833	0.803	0.818	0.853	0.53	20.871	48.2
YOLOv5l	0.864	0.788	0.824	0.856	0.537	46.138	108.2
YOLOv5x	0.874	0.782	0.825	0.864	0.549	86.218	204.6
YOLOv3	0.835	0.797	0.815	0.843	0.512	61.524	155.3
YOLOv4	0.767	0.884	0.819	0.781	0.493	64.4	60.363
YOLOv7	0.773	0.876	0.821	0.869	0.513	37.6	106.472
YOLOv8n	0.853	0.759	0.803	0.834	0.515	3.2	8.7
YOLOv8s	0.864	0.793	0.827	0.845	0.52	11.2	28.4
YOLOv8m	0.842	0.804	0.823	0.851	0.523	25.9	78.9
YOLOv8l	0.861	0.795	0.827	0.856	0.543	43.7	165.2
YOLOv8x	0.875	0.825	0.849	0.868	0.549	68.2	257.8
SSD	0.63	0.883	0.735	0.782	0.496	26.3	62.7
Efficientdet	0.837	0.738	0.784	0.798	0.497	3.874	5.2
Faster-R-CNN	0.802	0.613	0.695	0.753	-	137.1	370.2
GS-YOLOv5s	0.867	0.805	0.835	0.872	0.516	5.2	10.2

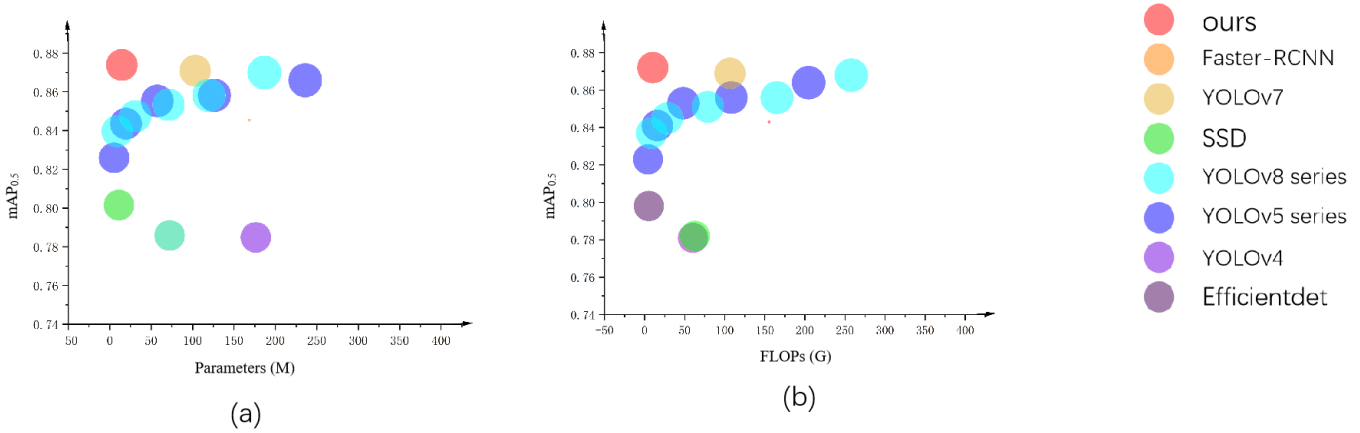


Figure 9. (**a**) Scatter plots of parameter numbers and $mAP_{0.5}$ of the mainstream one-stage and two-stage detectors and GS-YOLOv5s; (**b**) scatter plots of FLOPs and $mAP_{0.5}$ of mainstream one-stage and two-stage detectors and GS-YOLOv5s.

When the training reached 200 epochs, the results stabilized, with subsequent fluctuations remaining within an acceptable range. In other words, the curves show a trend of convergence. The improvements proposed in this study resulted in notable enhancements in the precision, recall, and mean average precision. Finally, in Figure 10, we provide a compare between GS-YOLOv5s and the baseline model YOLOv5s.

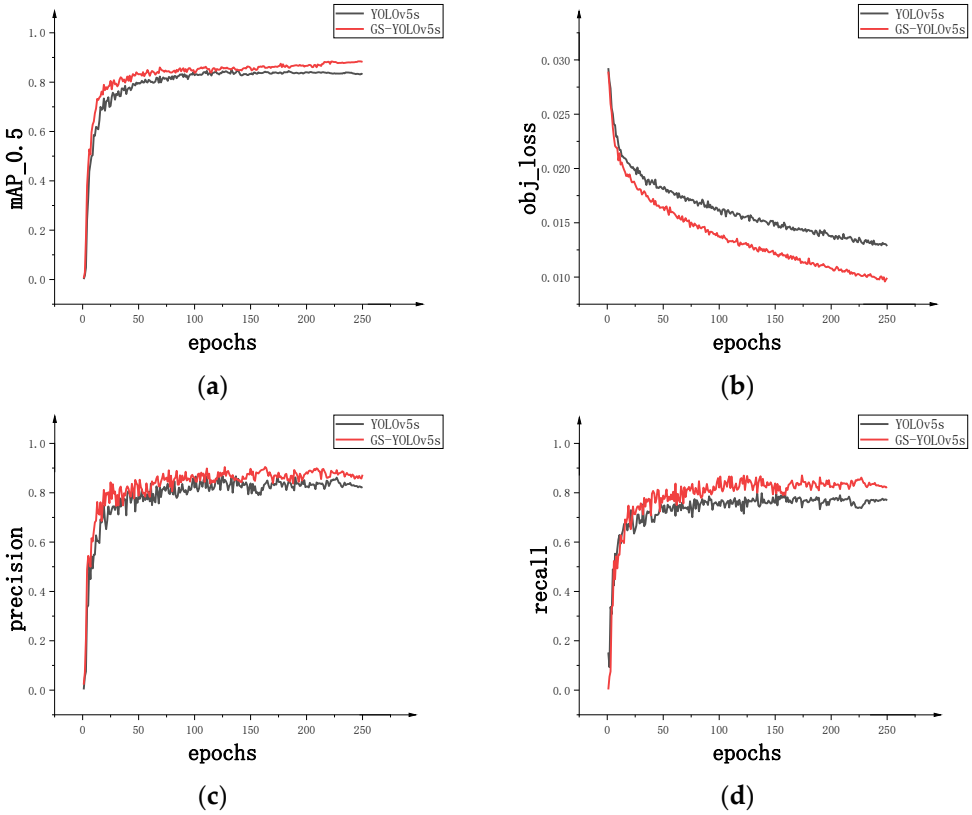


Figure 10. (**a**) Comparison of the mean average precision between GS-YOLOv5s and YOLOv5s. (**b**) Comparison of the regression Loss function between GS-YOLOv5s and YOLOv5s. (**c**) Comparison of the precision between GS-YOLOv5s and YOLOv5s. (**d**) Comparison of the recall between GS-YOLOv5s and YOLOv5s.

3.3. Comparison

According to the ablation experiments, it can be observed that although the YOLOv5s model reduces the number of parameters and improves the accuracy compared to previous models, there is still room for improvement in $mAP_{0.5}$ in forest fire detection. In the ablation experiments, C3 was replaced with GS-C2, WIoU was used instead of the original GIoU, and feature fusion knowledge distillation was applied to the model. After adding the GS-C2 module, the model's parameters and floating-point operations were reduced by 20% and 23.9% respectively, while $mAP_{0.5}$ increased by 0.8%. This indicates that GS-C2 effectively enhances the detection accuracy. Next, by replacing traditional convolution with GSConv in the neck network, parameters and floating-point operations were reduced by 25.7% and 25.1%, respectively, and $mAP_{0.5}$ increased by 2.1%, demonstrating the effectiveness of GSConv in forest fire detection. Furthermore, the addition of a focusing-mechanism-based WIoU loss function increases the overall $mAP_{0.5}$ by 2%, demonstrating WIoU's effectiveness in mitigating the challenges of the original GIoU loss function, such as slow convergence and inaccurate regression. Finally, knowledge distillation based on feature fusion using YOLOv5x as the "teacher model" improves $mAP_{0.5}$ by 1.9%, which indicates that feature fusion-based knowledge distillation effectively enhances detection accuracy without increasing the number of parameters. Comparing experimental data, we can conclude that these four improvements all contribute to improving the accuracy of forest fire detection to different degrees and further achieving lightweighting.

Next, the four improvements were fused in sequence in ablation experiments while adding GS-C2 and GSConv. The model's $mAP_{0.5}$ significantly surpassed that of only adding GS-C2 or GSConv, with further reductions in parameters and floating-point operations. These results indicate that GS-C2 and the WIoU loss function together can effectively enhance the model's forest fire detection performance. Replacing the original GIoU loss

function with the WIoU loss function on this basis elevated $mAP_{0.5}$ to a higher level. Compared to just adding one or two of the improvements, simultaneously incorporating GS-C2, GSConv, and WIoU significantly improves the forest-fire-detection accuracy while achieving lightweighting. Ultimately, through feature fusion-based knowledge distillation, the $mAP_{0.5}$ was raised to 87.2%, a 3.4% improvement compared to the baseline YOLOv5s model. This demonstrates that feature fusion-based knowledge distillation not only compensates for valuable information but also reduces the computational complexity. In conclusion, the GS-YOLOv5s structure proposed in this paper outperforms YOLOv5s, with a 3.4% increase in the $mAP_{0.5}$, while reducing overall parameters and floating-point operations by approximately 26% and 36%, respectively.

With these comprehensive improvements, the network can accurately detect forest fires in complex backgrounds and small targets, making it more suitable for deployment on embedded devices, allowing for quicker and more efficient forest fire detection. Visual comparisons of detection results between the proposed model and other widely used detection methods in the field of computer vision, including Faster R-CNN, SSD, and the YOLO series, validate the accuracy of GS-YOLOv5s. In summary, the distilled GS-YOLOv5s demonstrates superior performance compared to other existing detectors based on the dataset.

To illustrate detection performance more intuitively, the detection results for YOLOv5s, Faster R-CNN, and GS-YOLOv5s in complex backgrounds and small target scenarios and visualizations of the results are displayed in Figures 11 and 12, respectively.

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

(**d**) (**e**) (**f**)

Figure 11. GS-YOLOv5's performance in detecting forest fire targets in complex backgrounds. (**a**) YOLOv5s fail to detect some flames in the detection results. (**b**) Faster R-CNN has a relatively accurate detection result and a low probability of obtaining the detection result. (**c**) The detection results of GS-YOLOv5s are the most accurate. (**d**) YOLOv5s have a false detection in the upper left corner of the detection result. (**e**) Faster R-CNN mistakenly detected the forest firefighter's helmet as a flame based on the detection results. (**f**) The detection results of GS-YOLOv5s are the most accurate.

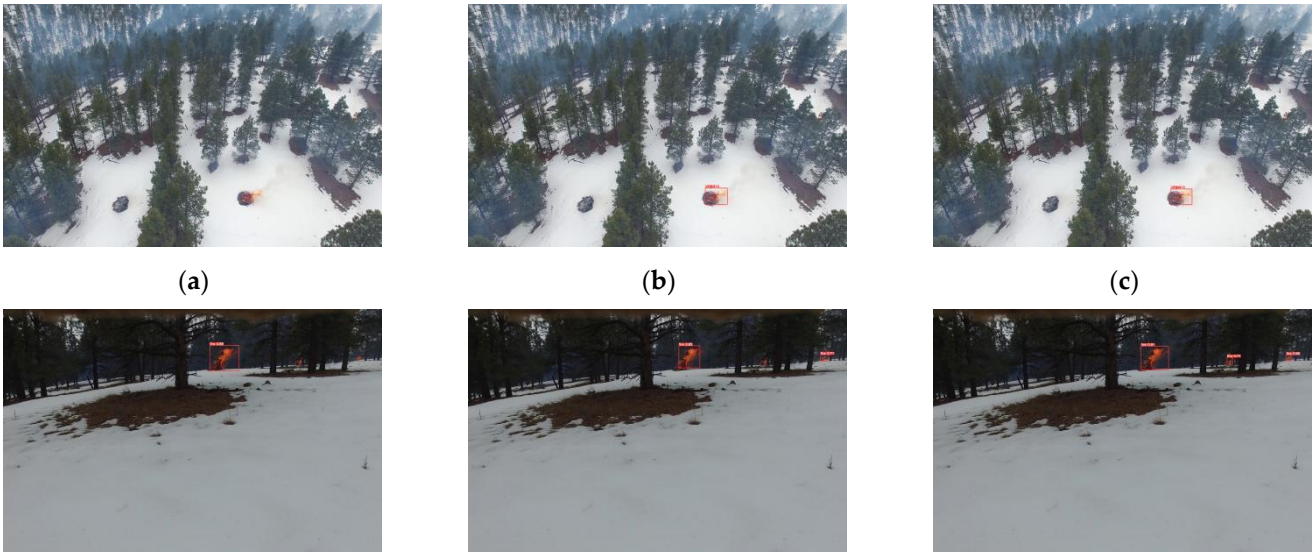


Figure 12. GS-YOLOv5's performance in detecting small target forest fires. (**a**) YOLOv5s fail to detect the flame target in the detection result. (**b**) The Faster R-CNN results are relatively accurate for the detection. (**c**) The detection results of GS-YOLOv5s are the most accurate. (**d**) YOLOv5s missed the flame detection in the upper right corner of the detection result. (**e**) Faster R-CNN missed the flame detection in the upper right corner of the detection result. (**f**) The detection results of GS-YOLOv5s show that both small target flames can be detected.

4. Discussion and Conclusions

In the task of object detection, a forest fire is difficult to detect as an object without a fixed shape. Forest fires, especially in complex environments, can be easily missed or falsely identified. Many large-scale forest fires often result from a lack of timely detection and intervention in their early stages, leading to the spread of the forest fire and causing significant loss of life and property. Therefore, improving the performance of detectors is of great importance for identifying small forest fire targets and interference caused by complex backgrounds.

Through experiments, it was found that the baseline model has limited capabilities in detecting small targets or targets with complex backgrounds, leading to missed and false detections. Additionally, the algorithm's computational load hinders its deployment on mobile devices. Therefore, in this paper, GSConv is added to reduce the number of model parameters, the GS bottleneck is integrated into the interstage local network module through feature branching, and WIoU based on the focusing mechanism is used to replace the original GIoU. Finally, the model is distilled through knowledge distillation based on feature fusion to achieve both improved detection accuracy and model lightweighting. The accuracy of the model was verified using 360 complex-background or small-target forest fire images. The detection accuracy is significantly improved when using GS-YOLOv5s. $mAP_{0.5}$, in detecting forest fires in our test set, increased by 3.1%, reaching 87.2%, while testing based on the same forest fire dataset, the $mAP_{0.5}$ of YOLOv5 was only 83.8%.

However, the model proposed in this paper still has its shortcomings, and further optimization of the forest-fire-detection network is necessary. Firstly, we will continue to explore more robust data-annotation methods as high-quality datasets can significantly enhance the model's detection capabilities. Furthermore, the number of parameters of current high-precision models are still large, so we will research ways to strike a balance between precision and lightweighting to enable the deployment of high-precision forest fire detectors on mobile devices, facilitating timely flame detection. While the model proposed in this paper demonstrated real-time performance when deployed on drones for data capture, it is still susceptible to false positives due to factors, like lighting and obstructions

during the capture process. In future work, we aim to enhance the stability of the detection model and reduce false positives.

In the ongoing research, we plan to equip drones with different types of cameras, including panoramic and stabilizing high-definition cameras, to capture clearer images. Additionally, we draw inspiration from Dong [33] who proposed a High-Speed Railway Signaling (HSRIS) object detection method based on adaptive target orientation features in convolutional neural networks (CNNs). This work provides insights for our future endeavors, where we will design regression methods using adaptive approaches. Equally important, inspired by the work of Wang [34] and colleagues, our future research will continue to refine the feature fusion module to enable the more precise extraction of object boundary information, especially for small objects.

Author Contributions: Conceptualization, T.W.; methodology, Y.C.; validation, Y.C. and T.W.; Writing—original draft, Y.C.; writing—review & editing, T.W.; supervision, T.W.; project administration, H.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Key Research and Development plan of Jiangsu Province (Grant No: BE2021716), Jiangsu Graduate Research and Practice Innovation Program (SJCX21_0338).

Data Availability Statement: The data presented in this study is available on request from the corresponding authors, and the dataset was jointly completed by the team, so the data is not publicly available.

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

References

1. Reddy, P.; Kalyanasundaram, P. Novel detection of forest ffre using temperature and carbon dioxide sensors with improved accuracy in comparison between two different zones. In Proceedings of the 2022 3rd International Conference on Intelligent Engineering and Management (ICIEM), London, UK, 27–29 April 2022; pp. 524–527.
2. Peruzzi, G.; Pozzebon, A.; Van Der Meer, M. Fight Fire with Fire: Detecting Forest Fires with Embedded Machine Learning Models Dealing with Audio and Images on Low Power IoT Devices. *Sensors* **2023**, *23*, 783. [CrossRef] [PubMed]
3. Kadir, E.; Rahim, S.; Rosa, S. Multi-sensor system for land and forest fire detection application in Peatland Area. *Indones. J. Electr. Eng. Inform. (IJEEI)* **2019**, *7*, 789–799.
4. Benzekri, W.; 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**, *12*, 5. [CrossRef]
5. Girshick, R.; Donahue, J.; Darrell, T. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
6. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards Real-time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef] [PubMed]
7. Cai, Z.; Nuno, V. Cascade R-CNN: Delving into High Quality Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 6154–6162.
8. Tan, M.; Le, Q.V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In Proceedings of the Machine Learning Research, Long Beach, CA, USA, 10–15 June 2019; arXiv:1905.11946.
9. Tan, M.; Pang, R.; Le, Q.V. EfficientDet: Scalable and Efficient Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; arXiv:1911.09070.
10. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S. SSD: Single Shot Multibox Detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; pp. 21–37.
11. 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.
12. Ding, X.; Gao, J.D. A new intelligent fire color space approach for forest fire detection. *J. Intell. Fuzzy Syst.* **2022**, *42*, 5265–5281. [CrossRef]
13. Mondal, M.S.; Prasad, V.; Kumar, R. A Multi-Layered Filtering Approach to Enhanced Fire Safety and Rapid Response. *Autom. Fire Detect. Suppr. Comput. Vis.* **2023**, *59*, 1555–1583.
14. Huang, J.; He, Z.; Guan, Y.; Zhang, H. Real-Time Forest Fire Detection by Ensemble Lightweight YOLOX-L and Defogging Method. *Sensors* **2023**, *23*, 1894. [CrossRef] [PubMed]
15. Hu, J.; Shen, L.; Albanie, S. Squeeze-and-excitation Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.

16. Sun, B.; Wang, Y.; Wu, S. An Efficient Lightweight CNN Model for Real-time Fire Smoke Detection. *J. Real-Time Image Process.* **2023**, *20*, 74. [CrossRef]
17. Zhou, M.; Liu, S.; Li, J. Multi-scale Forest Flame Detection Based on Improved and Optimized YOLOv5. *Fire Technol.* **2023**, *59*, 3689–3708. [CrossRef]
18. Lu, K.; Huang, J.; Li, J.; Zhou, J.; Chen, X.; Liu, Y. MTL-FFDET: A Multi-Task Learning-Based Model for Forest Fire Detection. *Forests* **2022**, *13*, 1448. [CrossRef]
19. Huang, J.; Zhou, J.; Yang, H.; Liu, Y.; Liu, H. A Small-Target Forest Fire Smoke Detection Model Based on Deformable Transformer for End-to-End Object Detection. *Forests* **2023**, *14*, 162. [CrossRef]
20. Chollet, F. Xception: Deep Learning with Depthwise Separable Convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 June 2017; IEEE: Piscataway, NJ, USA. [CrossRef]
21. Howard, A.G.; Zhu, M.; Chen, B. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv* **2017**, arXiv:1704.04861.
22. Zhang, X.; Zhou, X.; Lin, M. Shufflenet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6848–6856.
23. Gou, J.P.; Xiong, X.S.; Yu, B.S.; Du, L.; Zhan, Y.B.; Tao, D.C. Multi-target Knowledge Distillation via Student Self-reflection. *Int. J. Comput. Vis.* **2023**, *131*, 1857–1874. [CrossRef]
24. Yuan, J.L.; Jiang, Q.L.; Pan, Y. The Influence Mechanism of Knowledge Network Allocation Mechanism on Knowledge Distillation of High-Tech Enterprises. *Comput. Intell. Neurosci.* **2022**, *2022*, 8246234. [CrossRef] [PubMed]
25. Zou, P.; Teng, Y.; Niu, T. Multi-scale Feature Extraction and Fusion for Online Knowledge Distillation. In Proceedings of the Computer Vision and Pattern Recognition, New Orleans, LA, USA, 16 June 2022; arXiv:2206.08224.
26. Li, L.; Su, W.; Liu, F.; He, M.; Liang, X. Knowledge Fusion Distillation: Improving Distillation with Multi-scale Attention Mechanisms. *Neural Process Lett.* **2023**, *55*, 6165–6180. [CrossRef] [PubMed]
27. Zhao, J.; Qian, X.; Zhang, Y.; Shan, D.; Liu, X.; Coleman, S.; Kerr, D. A Knowledge Distillation-based Multi-scale Relation-prototypical Network for Cross-domain Few-shot Defect Classification. *J. Intell. Manuf.* **2023**, 1–17. [CrossRef]
28. Li, H.; Li, J.; Wei, H. Slim-neck by GSConv: A Better Design Paradigm of Detector Architectures for Autonomous Vehicles. *arXiv* **2022**, arXiv:2206.02424.
29. Tong, Z.; Chen, Y.; Xu, Z. Wise-IoU: Bounding Box Regression Loss with Dynamic Focusing Mechanism. *arXiv* **2023**, arXiv:2301.10051.
30. Rezatofighi, H.; Tsoi, N.; Gwak, J.Y.; Sadeghian, A.; Reid, I.; Savarese, S. Generalized Intersection over Union: A Metric and A Loss for Bounding Box Regression. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 658–666.
31. Romero, A.; Ballas, N.; Kahou, E. FitNets: Hints for Thin Deep Nets. In Proceedings of the 3rd International Conference on Learning Representations (ICLR), San Diego, CA, USA, 7–9 May 2015. [CrossRef]
32. Lin, T.; 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 (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
33. Dong, Z.; Wang, M.; Wang, Y.; Liu, Y.; Feng, Y.; Xu, W. Multi-Oriented Object Detection in High-Resolution Remote Sensing Imagery Based on Convolutional Neural Networks with Adaptive Object Orientation Features. *Remote Sens.* **2022**, *14*, 950. [CrossRef]
34. Wang, M.; Cui, X.; Wang, T.; Jiang, T.; Gao, F.; Cao, J. Eye Blink Artifact Detection Based on Multi-dimensional EEG Feature Fusion and Optimization. *Biomed. Signal Process. Control* **2023**, *83*, 104657. [CrossRef]