forests

MDPI

Article

YOLO-Tea: A Tea Disease Detection Model Improved by YOLOv5

Zhenyang Xue [1], Renjie Xu [2], Di Bai [3,*] and Haifeng Lin [1,*]

[1] College of Information Science and Technology, Nanjing Forestry University, Nanjing 210037, China
[2] College of Information Management, Nanjing Agricultural University, Nanjing 210095, China
[3] Department of Computing and Software, McMaster University, Hamilton, ON L8S 4L8, Canada
* Correspondence: baidi000@njau.edu.cn (D.B.); haifeng.lin@njfu.edu.cn (H.L.); Tel.: +86-25-8542-7827 (H.L.)

Abstract: Diseases and insect pests of tea leaves cause huge economic losses to the tea industry every year, so the accurate identification of them is significant. Convolutional neural networks (CNNs) can automatically extract features from images of tea leaves suffering from insect and disease infestation. However, photographs of tea tree leaves taken in a natural environment have problems such as leaf shading, illumination, and small-sized objects. Affected by these problems, traditional CNNs cannot have a satisfactory recognition performance. To address this challenge, we propose YOLO-Tea, an improved model based on You Only Look Once version 5 (YOLOv5). Firstly, we integrated self-attention and convolution (ACmix), and convolutional block attention module (CBAM) to YOLOv5 to allow our proposed model to better focus on tea tree leaf diseases and insect pests. Secondly, to enhance the feature extraction capability of our model, we replaced the spatial pyramid pooling fast (SPPF) module in the original YOLOv5 with the receptive field block (RFB) module. Finally, we reduced the resource consumption of our model by incorporating a global context network (GCNet). This is essential especially when the model operates on resource-constrained edge devices. When compared to YOLOv5s, our proposed YOLO-Tea improved by 0.3%–15.0% over all test data. YOLO-Tea's $AP_{0.5}$, AP_{TLB}, and AP_{GMB} outperformed Faster R-CNN and SSD by 5.5%, 1.8%, 7.0% and 7.7%, 7.8%, 5.2%. YOLO-Tea has shown its promising potential to be applied in real-world tree disease detection systems.

Keywords: tea leaf diseases; object detection; computer vision; deep learning

Citation: Xue, Z.; Xu, R.; Bai, D.; Lin, H. YOLO-Tea: A Tea Disease Detection Model Improved by YOLOv5. *Forests* **2023**, *14*, 415. https://doi.org/10.3390/f14020415

Academic Editor: Roberto Molowny-Horas

Received: 28 January 2023
Revised: 9 February 2023
Accepted: 15 February 2023
Published: 17 February 2023

1. Introduction

The tea-drinking culture has a long history of development around the world, and the market demand for tea is considerable. Safeguarding the growth safety of tea trees and the quality of tea leaves is vital to promoting the development of the tea industry. Tea leaf pests and diseases can have a significant impact on tea production, leading not only to low tea yields but also to poor tea quality, thereby causing economic damages to tea farmers and tea production manufacturers. Tea production is reduced by about 20% each year due to leaf blight and other diseases that infect the tea leaves [1]. It is important to research how to accurately detect and identify tea diseases and insect pests to reduce tea production losses, improve tea quality, and increase tea farmers' income.

Currently, the identification of tea diseases and insect pests in most tea gardens is based on manual detection [2]. Firstly, inviting experts to tea plantations for field identification is a waste of manpower, money, and time due to the long travel distances [2]. Secondly, most tea farmers do not have a comprehensive knowledge of the diseases and pests of tea. Tea farmers may fail to apply the appropriate pesticide due to false judgements. Thus, it is important to find an effective tea disease and pest identification system to help tea farmers quickly identify diseases and take actions.

With the rise of computer vision technology, researchers are beginning to apply image processing and machine learning techniques to the detection of diseases and pests in agricultural crops. Miranda et al. [3] extended the application of different image processing

techniques for pest detection and extraction by constructing an automated detection and extraction system for the estimation of pest densities in rice fields. Barbedo et al. [4] presented a method for identifying plant diseases based on color transformations, color histograms, and a pair-based classification system. However, the accuracy for identifying multiple plant diseases fluctuated between 40% and 80% when tested. Zhang et al. [5] proposed a leaf-image-based cucumber disease identification using K-means clustering and sparse representation classification. Hossain et al. [6] developed a system for image processing using a support vector machine (SVM) classifier for disease identification. It was able to identify and classify brown spot and algal leaf diseases from healthy leaves. Sun et al. [7] proposed a new method combining simple linear iterative cluster (SLIC) and SVM to achieve accurate extraction of tea tree leaf disease salinity maps in a complex background context. To summarize, the classical machine learning methods (e.g., random forests and support vector machines) for plant disease detection require manual extraction of plant leaf disease features. The manually extracted features may not be the essential characteristics of the crop disease, which will significantly affect the precision of the disease diagnosis.

With the development of deep learning techniques, more and more researchers are investigating using deep learning to detect crop leaf diseases and insect pests. Recent development of image recognition technologies has led to the widespread use of convolutional neural networks (CNN) in deep learning for automatic image classification and recognition of plant diseases [8]. Chen et al. [8] proposed a CNN model called LeafNet to automatically extract features of tea tree diseases from images. Hu et al. [9] proposed a low-shot learning method. They utilized SVM to separate diseased spots in diseased tea photos to remove background interference and a modified C-DCGAN to solve insufficient samples. Hu et al. [1] proposed a tea disease detection method based on the CIFAR10-quick model with the addition of a multiscale feature extraction module and depth-separable convolution. Jiang et al. [10] used CNN to extract rice leaf disease image features before using SVM to classify and predict specific diseases. The CNN-based tea leaf disease identification method outperforms traditional machine learning methods [1,11]. In the above method, the researchers used CNNs to automatically extract crop-disease-specific features instead of manually extracting them. While the above methods have performed well in the treatment of crop diseases, they focus solely on either crop disease image identification or classification.

In recent years, deep-learning-based image detection networks have been divided into two-stage and one-stage detection networks [12]. Faster region-based convolutional neural networks (Faster R-CNN) [13] is one of the more representative two-stage detection networks. Zhou et al. [14] proposed a rice disease detection algorithm based on Faster R-CNN and FCM-KM fusion and achieved relatively good performance. Although the detection accuracy of Faster R-CNN is good, the detection speed is slow and therefore cannot meet the real-time requirements. The one-stage detection networks are more efficient than the two-stage ones, although it is less accurate. You Only Look Once (YOLO) [15], Single Shot MultiBox Detector (SSD) [16], and RetinaNet [17] are representatives of one-stage detection networks. Among them, the YOLO family is widely used in agriculture due to their ability to detect efficiently and accurately. Tian et al. [18] designed a system based on YOLOv3 [19] that can detect apples at three different stages in the orchard in real time. Roy et al. [20] designed a high-performance real-time fine-grained target detection framework that can address obstacles such as dense distribution and irregular morphology, which is based on an improvement of YOLOv4 [21]. Sun et al. [22] proposed a novel concept for the synergistic use of the YOLO-v4 deep learning network for ITC segmentation and a computer graphics algorithm for refinement of the segmentation results involving overlapping tree crowns. Dai et al. [23] developed a crop leaf disease detection method called YOLOv5-CAcT based on YOLOv5. To the best of our knowledge, the YOLO family has already been widely used in the detection of leaf diseases and insect pests in

agricultural crops, which motivates us to consider Yolov5 as a baseline model However, there are scant studies that apply the YOLO family to the detection of tea diseases and pests.

To address this problem, in this paper, we propose YOLO-Tea, a tea disease identification system improved based on YOLOv5. The main contributions of this paper are as follows: (1) We use drones and mobile phones to photograph tea diseases and pests in their natural environment. (2) To address the problem of small targets for tea diseases and insect pests, we integrated self-attention and convolution (ACmix) [24] and convolutional block attention module (CBAM) [25] to the YOLOv5s model to improve the recognizability of the feature images. (3) We replaced the spatial pyramid pooling fast (SPPF) module with the receptive field block (RFB) [26] module to better retain global information on tea disease and pest targets. (4) We incorporated the GCnet [27] structure for YOLOv5 to achieve better tea leaf disease and pest detection while reducing parameter overhead.

We organize the rest of the paper as follows. In Section 2, we detail the dataset used in this paper, the improved modules of YOLO-Tea, and the model performance evaluation criteria. In Section 3, we describe in detail the configuration of the experimental equipment, some of the experimental parameters, the results of the ablation experiments, and the interpretation of the ablation experimental data. In Section 4, experimental results are discussed and the impacts of the CBAM module, ACmix module, SPFP module, and GCnet module on the identification of tea leaf diseases and insect pests are analyzed. Section 5 concludes the overall work.

2. Materials and Methods

2.1. Dataset

2.1.1. Data Acquisition

The photographs of tea diseases and insect pests used in this study were taken outdoors in a natural environment. Photographs of tea diseases and insect pests were taken at the Maoshan tea farm in Jurong, Zhenjiang, Jiangsu Province, China. Tea leaf disease and insect pest images were collected on the afternoon of 1 July 2022. We conducted the image collection in clear, well-lit weather and in a good collection environment. As our images were really taken in a natural environment, we took images of tea with not only dense targets, but also images with sparser targets. In photographs with dense foliage, there is foliage obscuration, foliage overlap, etc. Reflections and shadows on the leaves are also present in the pictures as they were taken in the afternoon when the sun was shining.

We used a drone (DJI Mavic Air 2) and a mobile phone to take images of tea leaf diseases and insect infestations about 50 cm above the tea trees. Our images include images of one disease leaf and one pest leaf, namely, tea leaf blight and green mirid bug. The scientific name of green mirid bug is Apolygus lucorum (Meyer-Dür) [28]. The pest, green mirid bug, is represented by images of tea leaves that have been bitten by the green mirid bug. Some representative samples of our dataset are shown in Figure 1.

Figure 1. Some representative samples of our dataset. (**a**) Tea leaf blight; (**b**) tea leaf blight; (**c**) green mirid bug; (**d**) green mirid bug.

2.1.2. Dataset Annotation and Partitioning

Firstly, the photographs taken using drones and mobile phones were manually screened to obtain a total of 450 images of tea diseases and insect pests, including tea leaf blight and green mirid bug. Each of these 450 images contains multiple tea leaf blight and green mirid bug targets. We counted the total number of tea leaf blight and green mirid bug targets contained in these 450 images through the code. Secondly, we used Labelimg to label the data. The labeling file was saved in txt format. Finally, we randomly divided the labeled images into training set, validation set, and test set in the ratio of 8:1:1. The details of the dataset are shown in Table 1.

Table 1. Target numbers in tea dataset.

Target	Training Set	Validation Set	Test Set
Green mirid bug (D00)	1346	141	177
Tea leaf blight (D10)	1812	192	181

2.2. YOLO-Tea

2.2.1. YOLOv5

YOLOv5 is a member of the YOLO series presented by the Ultralytics LLC team. Depending on the network width and depth, YOLOv5 can be classified as YOLOv5n, YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x. In this paper, our proposed YOLO-Tea tea disease and pest detection model is improved based on YOLOv5s. The image inference speed of the Yolov5s model reaches 455FPS on a subset of researchers' devices, which is widely used by a large number of scholars with this advantage [29].

As shown in Figure 2, YOLOv5s-6.1 is divided into four parts: input, backbone, neck, and head. Cross-stage partial 1 (CSP1) and cross-stage partial 2 (CSP2) in the backbone and neck of YOLOv5 are designed with reference to the cross-stage partial network (CSP-Net) [30] structure. CSP1 is used for feature extraction in the backbone section. CSP2 is used for feature fusion in the neck section. In the backbone, there is a spatial pyramid pooling fast (SPPF) module in addition to the CSP1 module. The function of the SPPF module is to extract the global information of the detection target. In the neck, YOLOv5 uses the path aggregation network (PANet) [31] structure. The PANet structure not only merges the extracted semantic features with the location features, but also the features of the backbone with the head, so that the model obtains more abundant feature information. Finally, the head consists of three branches, with feature maps of different sizes used to detect target objects of different sizes.

2.2.2. Convolutional Block Attention Module

The convolutional block attention module (CBAM) is a simple and effective attention module that can be integrated with any feedforward convolutional neural network [25]. The structure of CBAM is shown in Figure 3. The CBAM module consists of a channel attention module (CAM) and a spatial attention module (SAM). Firstly, the feature map F is input to the channel attention module to obtain the channel attention feature weights $M_c(F)$. Then, F' is obtained by multiplying F with $M_c(F)$. Secondly, F' are passed into the spatial attention module to obtain the spatial attention feature weights $M_s(F')$. Finally, F'' is obtained by multiplying $M_s(F')$ with F'.

2.2.3. ACmix

Convolution and self-attention are two powerful techniques for representation learning [24]. Pan et al. [24] proposed a mixed model that enjoys the benefit of both self-attention and convolution (ACmix). The structure of ACmix is shown in Figure 4.

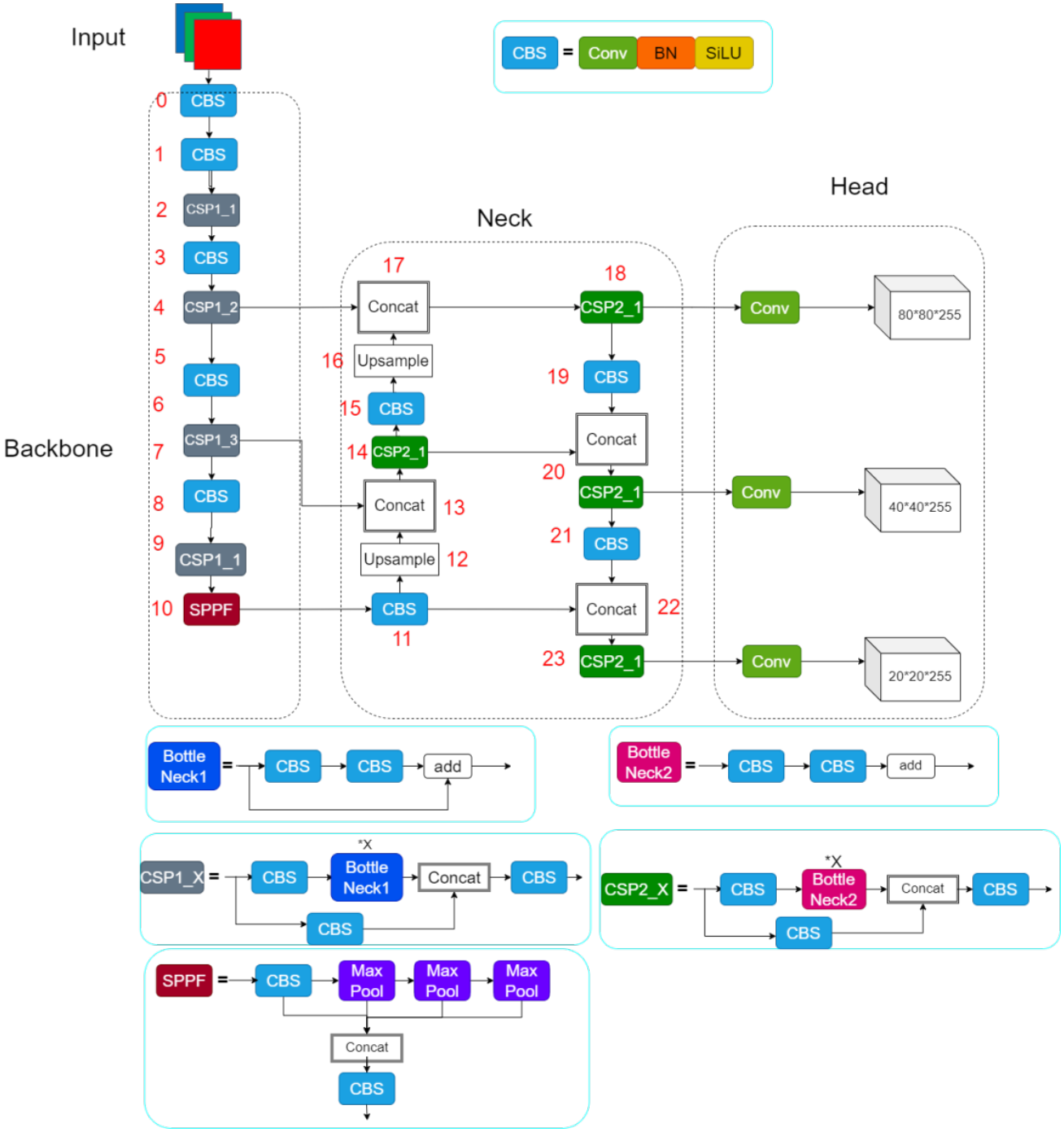

Figure 2. Structure picture of YOLOv5s-6.1.

Firstly, they reshape the input features into N segments using three 1×1 convolutions in the first stage, respectively. Through this method, a feature set containing $3 \times N$ feature maps is obtained. Secondly, in the second stage, they input the feature set obtained in the first stage into two paths: the self-attention path and the convolution path. In the self-attention path, they divide the features obtained in the first stage into N groups. Each group contains three 1×1 convolutional output features from the first stage. The corresponding three feature maps are used as queries, keys, and values, following the traditional multiheaded self-attention module. In the convolution path, since the convolution kernel size is k, the feature map obtained in the first stage is transformed into k^2 feature maps using the light fully connected layer. Subsequently, they generate features by shifting and

aggregation. Finally, the feature maps output by the self-attention path and the convolution path are summed, and the intensity can be controlled by two learnable scalars.

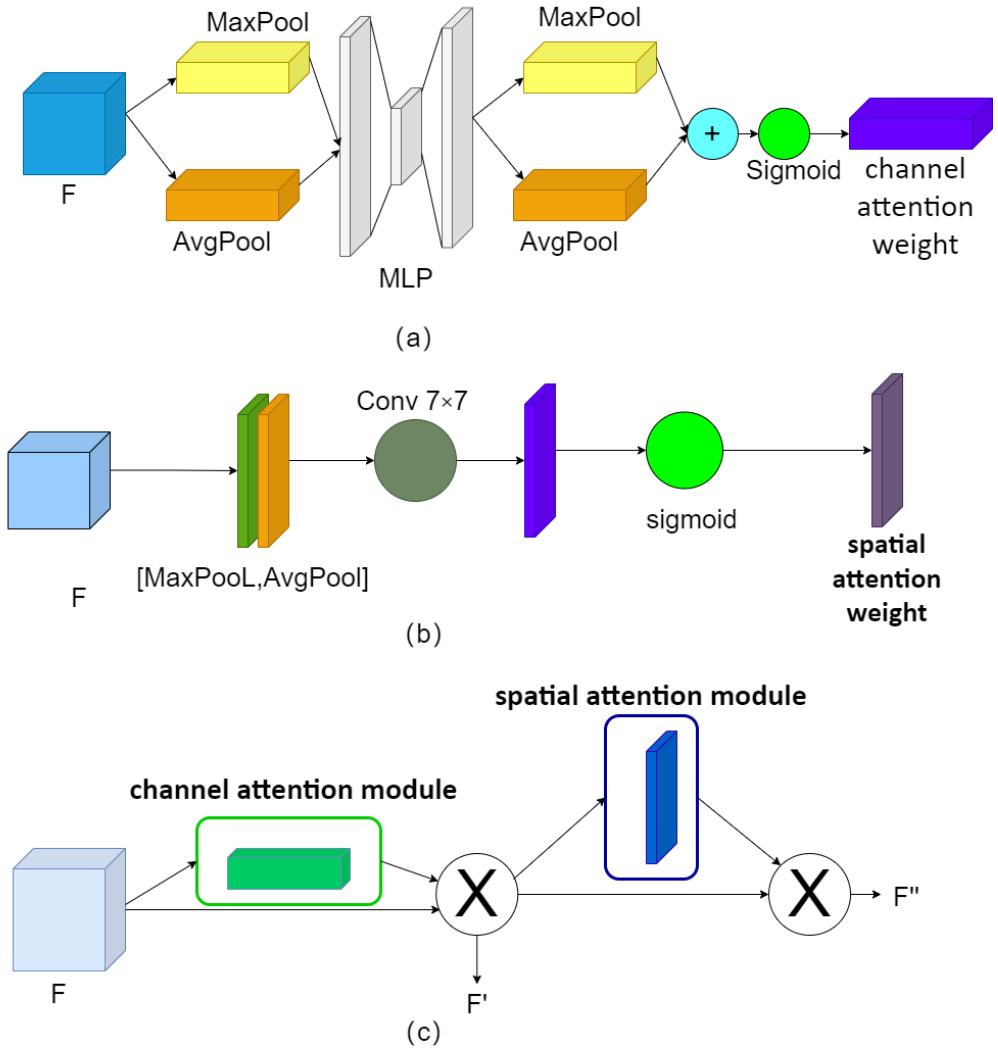

Figure 3. The details of CBAM. (**a**) The structure of CAM. (**b**) The structure of SAM. (**c**) The structure of CBAM.

2.2.4. Receptive Field Block

Liu et al. [26] proposed the receptive field block (RFB), which was inspired by the perceptual fields of human vision, to enhance network extraction capabilities. The RFB network is composed of several convolutional layers with different sizes of convolutional kernels. Each branch uses a combination of convolutional kernels of different scales and cavity convolution with different expansion rates, allowing the perceptual field of each branch to expand to different degrees. The structure of RFB borrows from that of Inception by adding a cavity convolution to Inception, which effectively increases the receptive field.

The structure of RFB is shown in Figure 5. Firstly, the parameters were reduced by 1×1 convolutional dimensionality reduction. Secondly, 1×1, 3×3, and 5×5 convolutions are performed to simulate different scales of perceptual fields. Thirdly, the 1×1, 3×3, and 5×5 convolution kernels are connected with the 3×3 dilated convolutions corresponding to expansion rates of 1, 3, and 5, respectively. Finally, the output of each branch is concatenated to fuse different features and improve the network model's ability to represent different-sized targets. In addition, the RFB module also adopts the shortcut in ResNet, which can effectively mitigate the gradient disappearance and improve the training performance of the network.

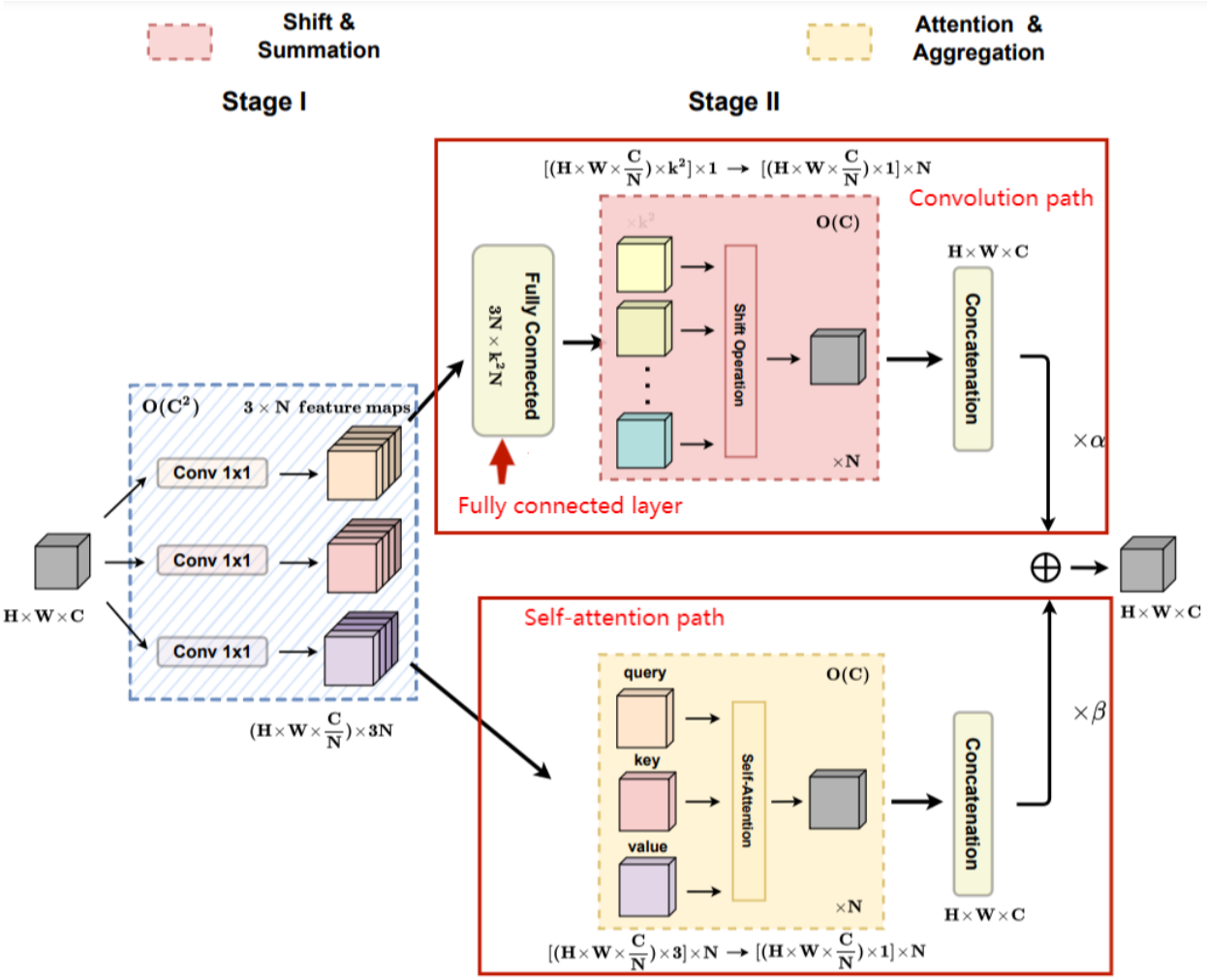

Figure 4. The structure of ACmix.

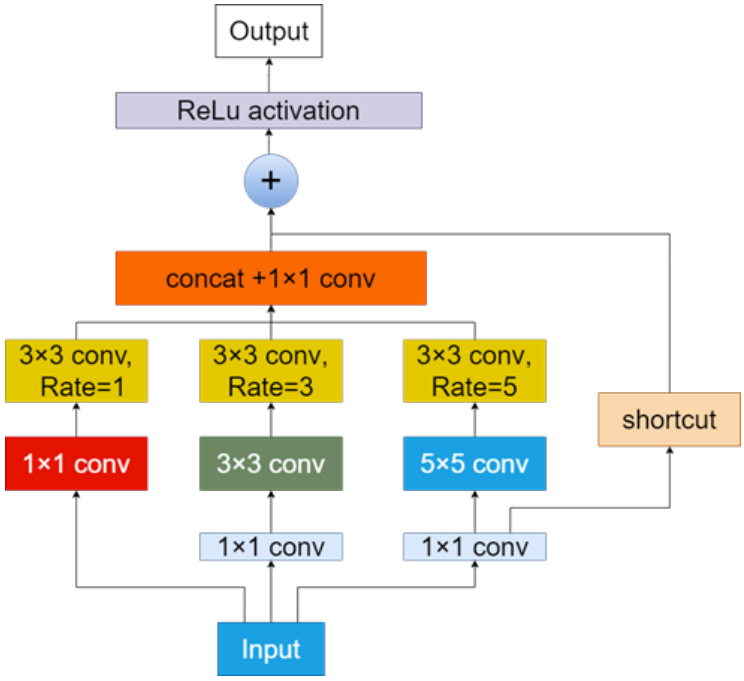

Figure 5. The structure of RFB.

2.2.5. The Methods for Integrating GCNet

Cao et al. [27] proposed a simplified nonlocal (SNL) block based on the nonlocal (NL) [32] block, which maintains the accuracy of the NL block and reduces the computational effort. In addition, they proposed a global context (GC) block based on their proposed SNL block and squeeze-and-excitation (SE) [33] block, which is GCNet. The GC block is lightweight and models the global context efficiently.

The way that we merge GCNet and YOLOv5 is shown in Figure 6. In Figure 6a, we can see that the structure of the GCNet can be summarized in three steps: global context modeling, channel-wise dependency transformation, and feature fusion. The global context modeling of the GCNet uses the further simplified SNL block. The aim of further simplifying the SNL block is to make the GCNet even lighter. In the transform stage, GCNet uses two 1×1 convolutions and applies LayersNorm and ReLU loss functions between these two convolutions. Finally, feature fusion is performed by broadcast element-wise addition. In Figure 6b, we fuse GCnet with the CSP1 structure in the backbone in YOLOv5. The CSP1 module that incorporates the GCNet is named CSP1GC.

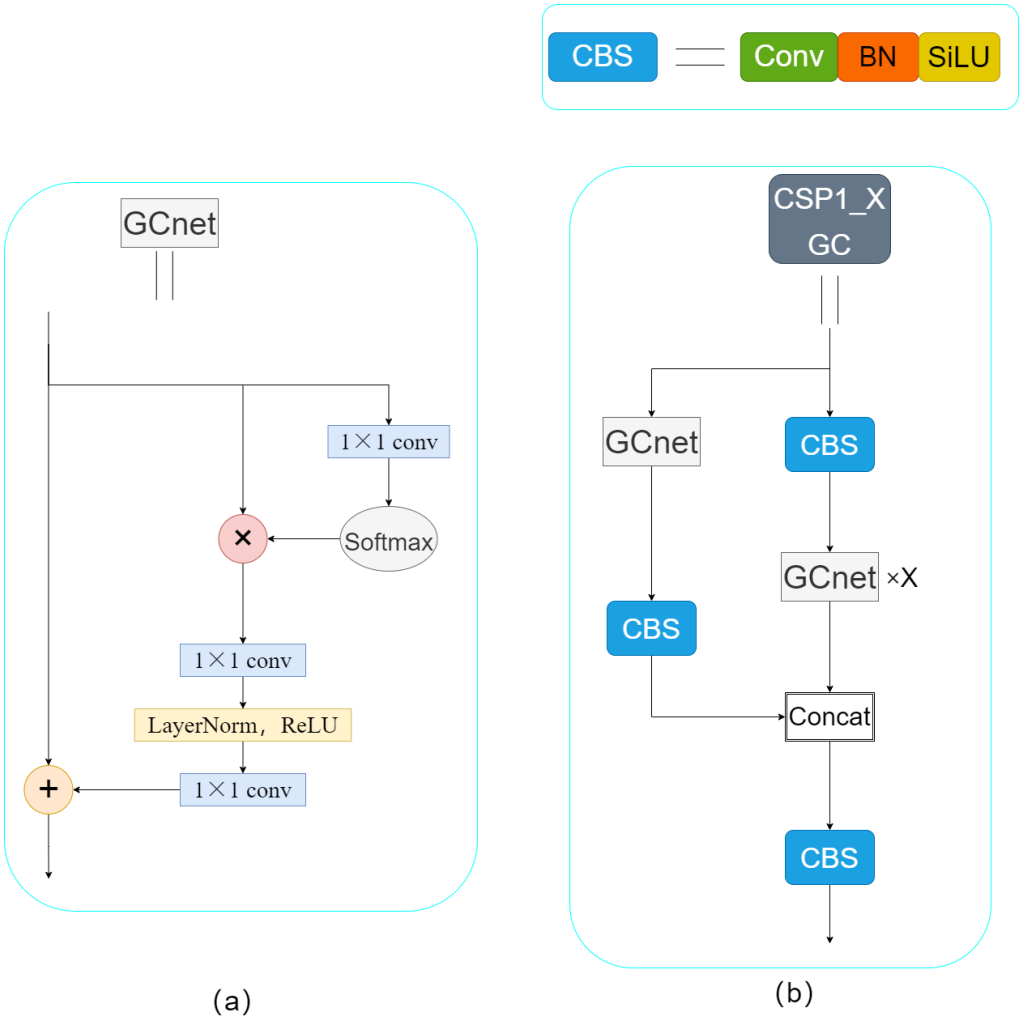

Figure 6. The methods for integrating GCNet. (**a**) The structure of GCNet. (**b**)The CSP1 structure of the fused GCNet.

2.2.6. The Structure of YOLO-Tea

The structure of the YOLO-Tea obtained by improving YOLOv5s is shown in Figure 7. Firstly, we replaced the CPS1 module on layers 2, 4, 6, and 8 of YOLOv5 with the CSP1GC module. The CSP1GC module is not only lighter than the original CSP1 module, but also allows for a better global focus on tea disease and pest targets. Secondly, in order to solve

the problem of missing feature information due to the low pixel count of small targets for tea leaf diseases and pests, we added the ACMix module to the backbone section and the CBAM module to the neck of YOLOv5. The ACmix module was only added at layer 9 in the backbone section due to the high overhead of the ACmix parameter. The CBAM module is lighter so it is added at layers 19, 23, and 27 in the neck section. Thirdly, we replaced the SPPF module in the original YOLOv5 with the RFB module in order to obtain better global information on tea disease and pest targets. In the head, 20×20, 40×40 and 80×80 feature maps are output, which are used to detect large, medium, and small targets, respectively. Each of these three feature maps contains three anchors, so there are a total of nine anchors in YOLO-Tea. This corresponds to three detection heads for large, medium and small targets.

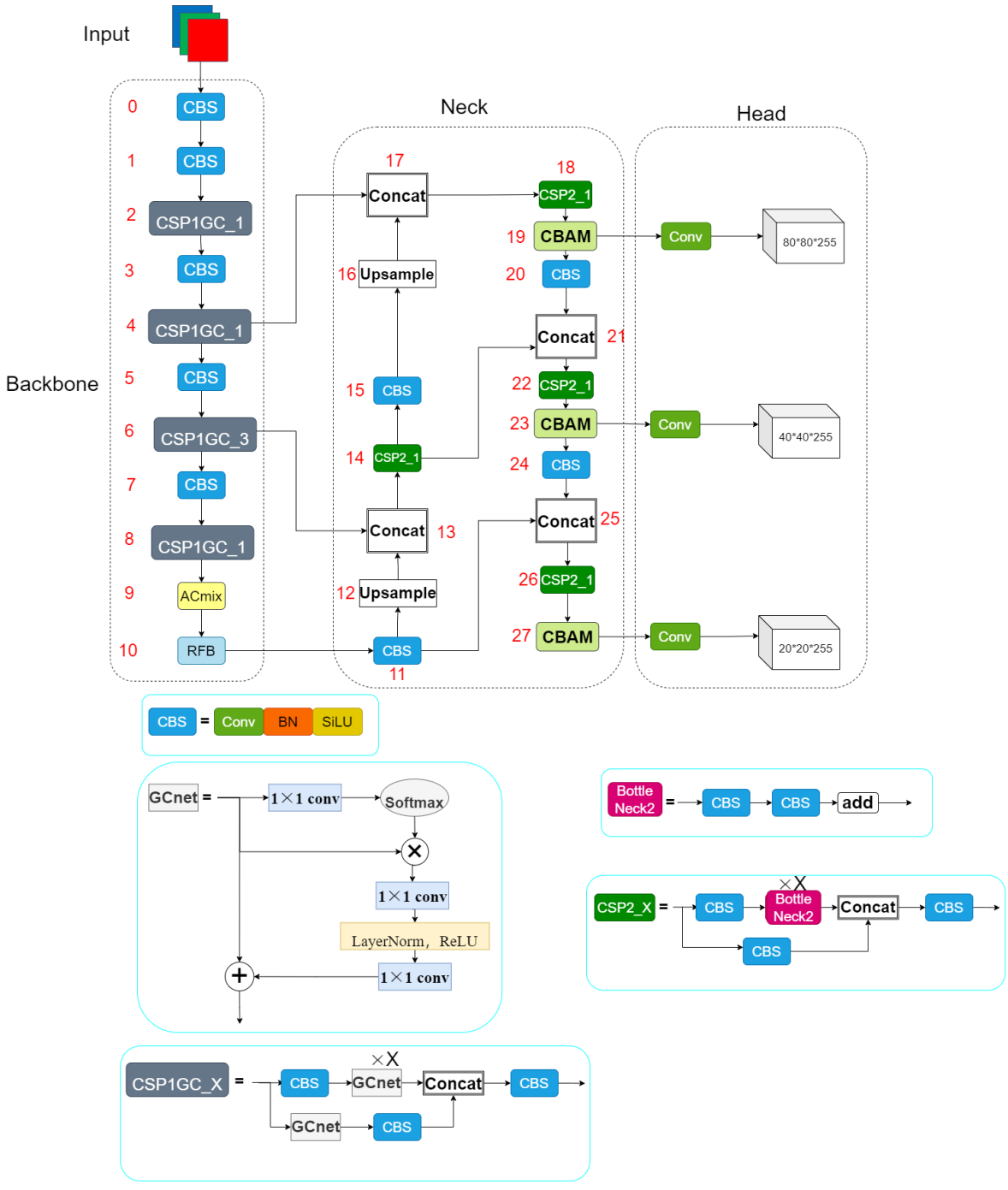

Figure 7. The structure of YOLO-Tea.

2.3. Model Evaluation

After reading the model evaluation criteria in the article of Khasawneh et al. [34], we introduced some additional data to their model evaluation criteria to help us better judge the performance of the model.

We used cross-validation to conduct experiments on the improved model. We used the training and validation sets in Section 2.1.2 for the training of the model. After the training was completed, the model was tested using the test set obtained in Section 2.1.2 to obtain the experimental data.

We chose the Microsoft COCO benchmark, which is the most popular benchmark in the object detection domain to evaluate the performance of the YOLO-Tea model in this paper, furthermore, because the tea leaf disease and insect pest dataset we use includes targets for both tea leaf blight and green mirid bug. The Microsoft COCO evaluation metrics, on the other hand, lacked a separate output for each target category. As a result, two new indicators were added to the Microsoft COCO evaluation metrics: AP_{TLB} and AP_{GMB}. The model evaluation metrics used are shown in Table 2.

Table 2. Microsoft COCO evaluation metrics and the metrics we added.

Microsoft COCO Evaluation Metrics	
Average Precision (AP)	
$AP_{0.5}$	AP at IoU = 0.5
AP Across Scales	AP at IoU = 0.5:0.95
AP_S	AP for small target (Size < 32^2)
AP_M	AP for medium target (32^2< Size < 96^2)
AP_L	AP for large target (Size > 96^2)
Average Recall (AR)	
$AR_{0.5:0.95}$	AR at IoU = 0.5:0.95
AR Across Scales	
AR_S	AR for small target (Size < 32^2)
AR_M	AR for medium target (32^2< Size < 96^2)
AR_L	AR for large target (Size > 96^2)
The metrics we added	
AP_{TLB}	AP for tea leaf blight
AP_{GMB}	AP for green mirid bug

The precision (P) rate represents the ratio of targets detected correctly by the model to all detected targets. The formula for calculating the precision rate is shown in Equation (1). In the formula, TP means that the prediction is tea leaf blight or green mirid bug and the prediction is correct, and FP means that the prediction is tea leaf blight or green mirid bug and the prediction is incorrect.

$$\mathrm{P} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{1}$$

Recall (R) indicates the proportion of targets correctly predicted by the model as a percentage of all targets. The formula for calculating the recall rate is shown in Equation (2). FN indicates that the target is tea leaf blight or green mirid bug target and the model detects it incorrectly.

$$\mathrm{R} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{2}$$

Average precision (AP) is the area enclosed by the axes below the precision–recall curve, which is the curve plotted with precision as the *y*-axis and recall as the *x*-axis. When additional enclosing boxes are accepted, the precision value is shown via a precision–recall curve (i.e., higher recall value due to a low threshold of class probability). A strong model can sustain high precision as recall rises [34]. Typically, the intersection over union (IoU) threshold is set at 0.5. In general, higher AP represents better model performance. Note

that $AP_{0.5}$ in the Microsoft COCO evaluation metrics is equivalent to mAP@0.5. mAP@0.5 is the arithmetic mean of the APs for all target categories. The formulas for calculating AP and mAP are shown in Equations (3) and (4).

$$\mathrm{AP} = \int_0^1 \mathrm{P(r)dr} \tag{3}$$

$$\mathrm{mAP} = \sum_{i=1}^{C} \mathrm{AP_i}/\mathrm{C} \tag{4}$$

Average recall (AR) represents twice the area of the R–IoU curve devolved to the coordinate axis. The value of AR is similar to the value of AP, with higher values representing better model performance. The formula for calculating AR is shown in Equation (5).

$$\mathrm{AR} = 2\int_{0.5}^{0.95} \mathrm{R(IoU)dIoU} \tag{5}$$

3. Results

3.1. Training

The experiment settings in this paper are shown in Table 3. Some of the main training parameters for the tea disease and pest detection model were set as shown in Table 4.

Table 3. The experimental conditions.

Experimental Environment	Details
Programming language	Python 3.9
Operating system	Windows 10
Deep learning framework	Pytorch 1.8.2
GPU	NVIDIA GeForce GTX 1070

Table 4. The main training parameters.

Training Parameters	Details
Epochs	300
img-size(pixel)	640 × 640
Initial learning rate	0.01
Optimization algorithm	SGD

3.2. Ablation and Comparison Experiments

The experimental procedure in this paper is as follows: Firstly, we trained the model using the training and validation sets. Then, we used the test set to test the trained model and obtain the data we needed to evaluate the model. Detailed data from the ablation experiments are shown in Table 5.

In addition to the ablation experiments, we also carried out a number of comparative experiments. We compared our proposed YOLO-Tea model with the Faster RCNN and SSD. In these experiments, our main experimental data of interest are $AP_{0.5}$, AP_{TLB}, and AP_{GMB}. Detailed results from the comparison experiments are shown in Table 6.

The precision–recall curves of the results of the ablation and comparison experiments are shown in Figure 8.

To verify that the CSP1GCNet fused with GCNet reduces the resource consumption of the model, we compared the number of parameters and model size of YOLOv5 fused with GCNet to the original YOLOv5. In addition, we compared the number of parameters and model size for YOLO-Tea and YOLO-Tea without fused GCNet (YOLOv5 + CBAM + ACmix + RFB). Detailed experimental data are shown in Table 7.

Table 5. The data from the ablation experiments.

Model	$AP_{0.5}$	AP_S	AP_M	AP_L	$AR_{0.5:0.95}$	AR_S	AR_M	AR_L	AP_{TLB}	AP_{GMB}
YOLOv5s (baseline)	71.7	31.1	53.6	70.1	50.2	44.6	52.6	72.1	69.3	74.1
YOLOv5s + CBAM	72.4	31.4	54.4	75.3	51.6	42.2	55.4	76.1	73.3	76.3
YOLOv5s + GCnet	73.0	34,2	54.6	72.4	54.0	45.4	57.5	73.3	66	80.1
YOLOv5 + RFB	73.7	33.1	53.9	71.2	52.4	44.9	55.4	67.5	71.2	76.1
YOLOv5 + ACmix	73.7	31.2	53.8	72.1	51.1	41.7	54.9	72.4	70.6	76.8
YOLOv5 + CBAM + ACmix	75.0	33.0	54.2	79.9	52.7	39.8	58.1	80.0	71.2	78.7
YOLOv5 + CBAM + ACmix + RFB	75.8	32.8	56.7	82.5	54.6	44.6	**58.5**	84.5	71.7	79.9
YOLOv5 + CBAM + ACmix + RFB + GCnet (YOLO-Tea)	**79.3 (+7.6)**	**34.2 (+3.1)**	**57.3 (+3.7)**	**85.1 (+15.0)**	**54.7 (+4.5)**	**44.9 (+0.3)**	57.4 (+4.8)	**85.7 (+13.6)**	**73.7 (+4.4)**	**82.6 (+8.5)**

Table 6. The data from the comparative experiments.

Model	$AP_{0.5}$	AP_{TLB}	AP_{GMB}
YOLOv5s	71.7	69.3	74.1
Faster R-CNN	73.8	71.9	75.6
SSD	71.6	65.9	77.4
YOLO-Tea	**79.3**	**73.7**	**82.6**

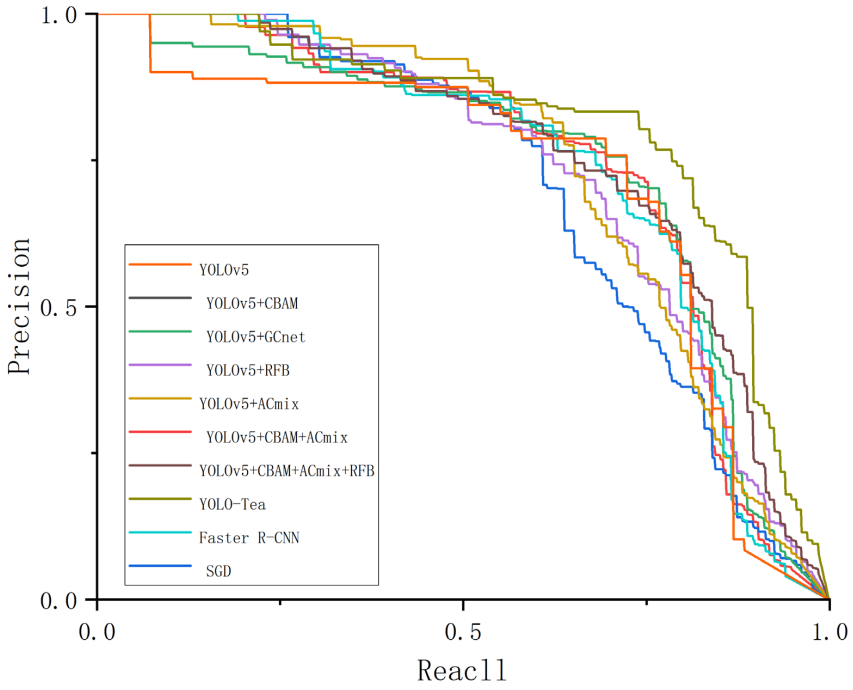

Figure 8. The precision–recall curves of the results.

Table 7. The data of resource consumption experiment.

Model	Parameters	Model Size (MB)
YOLOv5s	7025023	13.7
YOLOv5s + GCnet	6262778	13.0 (−0.7)
YOLOv5 + CBAM + ACmix + RFB	8722131	17.1
YOLO-Tea	7959886	15.6 (−1.5)

3.3. Comparison

The comparison experiments in Section 3.2 show that YOLO-Tea's $AP_{0.5}$, AP_{TLB}, and AP_{GMB} not only improve by 7.6%, 4.4%, and 8.5%, respectively, over the native YOLOv5s, but they also improve over both the Faster R-CNN model and the SSD model. Among them, Faster R-CNN as a two-stage target detection model, $AP_{0.5}$, AP_{TLB}, and AP_{GMB} were 2.1%, 2.6%, and 1.5% higher than YOLOv5s, respectively, but YOLO-Tea's $AP_{0.5}$, AP_{TLB}, and AP_{GMB} were 5.5%, 1.8%, and 7.0% higher than Faster R-CNN. The results from these comparative experiments also demonstrate the design of our proposed YOLO-Tea model.

In Experiment 2 of the ablation experiments in Section 3.2, we added the CBAM module to the YOLOv5s model. Although the AR_S of the model with the CBAM added was reduced by 2.4% compared to YOLOv5s, all other data improved, by 0.3%–5.2%, respectively. This demonstrates the effectiveness of including the CBAM module in YOLOv5s. Similarly, Experiment 3 of the ablation experiment showed that combining YOLOv5s with the GCnet module results in a significant improvement in model performance. Experiment 4 of the ablation experiment showed that replacing the SPPF module in YOLOv5s with the RFB module can also result in effective performance improvements. Experiment 5 of the ablation experiment demonstrated that adding the ACmix module to YOLOv5s improved performance significantly.

In Experiment 6 of the ablation experiment, we not only added CBAM to YOLOv5s, but also fused YOLOv5s with the GCnet module. The test data for the improved module improved over YOLOv5s, except for the ARs which decreased by 4.8% compared to YOLOv5s. This proved that the YOLOv5s model with the addition of the CBAM module was effective when fused with the GCnet module. Similarly, Experiment 7 demonstrated that the addition of the CBAM module to YOLOv5s, the fusion of the GCnet module, and the replacement of the SPPF module with the RFB module can lead to performance improvements.

Finally, in Experiment 8 of the ablation experiment, our proposed YOLO-Tea showed an improvement of 0.3%–15.0% in all test data compared to YOLOv5s. Furthermore, as can be seen from the precision–recall curves shown in Figure 8, YOLO-Tea has an even higher precision as recall rises. Figures 9 and 10 show a comparison between the YOLO-Tea and the YOLOv5s results of detection.

In Figure 9a, YOLOv5s detected 21 targets of tea leaf blight disease. However, these 21 tea leaf blight targets contained one false detection and one duplicate detection. There were also two missed targets for green mirid bug infestation. A total of 20 targets for the tea leaf blight disease and two targets for the green mirid bug pest were correctly identified by YOLO-Tea in Figure 9b. In Figure 9c, YOLOv5s detected seven tea leaf blight disease targets correctly, but missed one green mirid bug target and detected two tea leaf blight disease targets incorrectly. In Figure 9d, YOLO-Tea correctly detected four tea leaf blight disease targets and one green mirid bug pest target, though. However, three tea leaf blight disease targets were missed.

In Figure 10a, while the YOLOv5s correctly detected four green mirid bug infestation targets, it also failed to detected two green mirid bug infestation targets. In Figure 10b, YOLO-Tea correctly detected four targets of green mirid bug infestation. In Figure 10c, while YOLOv5s correctly detected four tea leaf blight targets, it failed to detect two tea leaf blight target and missed one green mirid bug target. In Figure 10d, YOLO-Tea correctly detected six tea leaf blight disease targets and one green blind bug infestation target.

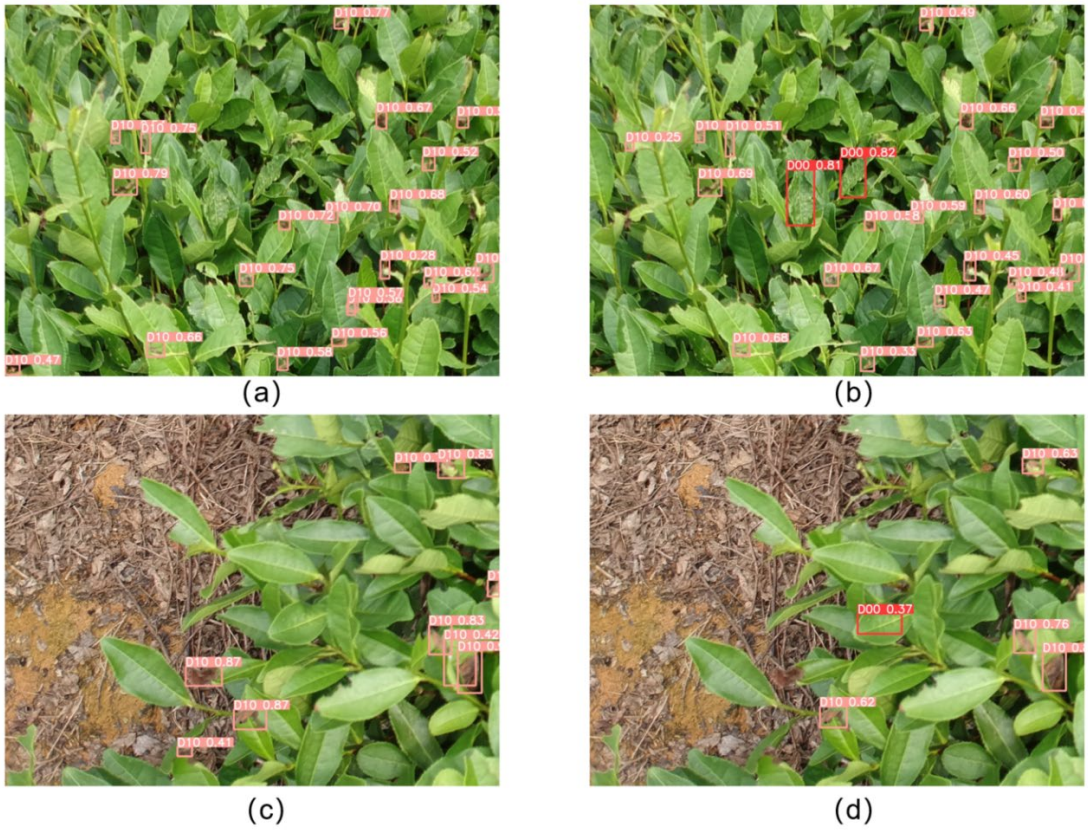

Figure 9. Comparison of model detection results. (**a**) YOLOv5's detection results. (**b**) YOLOv5-Tea's detection results. (**c**) YOLOv5's detection results. (**d**) YOLOv5-Tea's detection results.

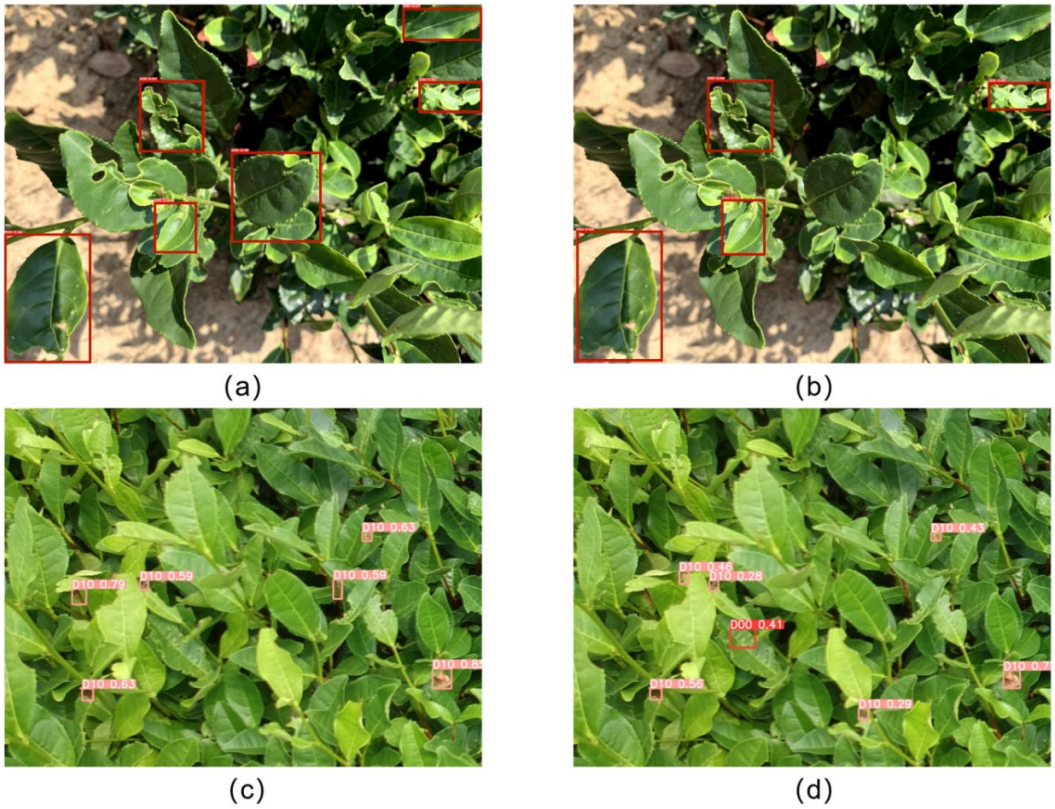

Figure 10. Comparison of model detection results. (**a**) YOLOv5's detection results. (**b**) YOLOv5-Tea's detection results. (**c**) YOLOv5's detection results. (**d**) YOLOv5-Tea's detection results.

4. Discussion

Due to various characteristics such as texture, shape, and color, diseases and insect pests of tea tree leaves are hard to accurately detect. The leaves of tea trees are also much smaller than those of other crops, which makes it difficult to detect disease on such a small target. The YOLOv5s model's performance fell short of what was needed for our subsequent study in the face of these issues. As a result, we enhanced the YOLOv5s model in numerous ways.

First, as YOLOv5 is not able to focus effectively on small targets of tea leaves, we made our model more effective in focusing on tea leaflet targets by adding ACmix and CBAM. ACmix fuses convolution and self-attention, two powerful techniques in computer vision. The detection performance of models with ACmix is improved by 1.3%–2.0%, in terms of $AP_{0.5}$, AP_{TLB}, and AP_{GMB}. However, ACmix has a large number of parameters. ACmix cannot be added to the model in large numbers in order to keep the model real time. CBAM is a lightweight module obtained by concatenating the channel attention module and the spatial attention module. The parameters of CBAM are smaller than those of ACmix, so we added it at three sites in the model (Figure 7). The addition of CBAM improved the performance of the model by 0.7%–4.0% in terms of $AP_{0.5}$, AP_{TLB}, and AP_{GMB}.

Second, RFB is a combination of convolutional kernels of different sizes and dilated convolution, with the creators of RFB believing that larger convolutional kernels have a larger field of perception. We will tentatively refer to the YOLOv5 model with the addition of ACmix and CBAM as Model A. Replacing the SPPF module in Model A with the RFB module improves performance by 0.5%–1.2%.

Thirdly, the addition of ACmix, CBAM, and the replacement of SPPF with RFB to YOLOv5 (7025023) improved the model performance, but the significant increase in the number of parameters in the model (8,722,131) led to a reduction in the real-time performance of the model in detecting tea diseases and insect pests. We chose GCnet to improve the cross-stage partial 1 (CSP1) module in YOLOv5. YOLOv5 with GCnet had a performance increase of 0.8%–1.9%, in terms of $AP_{0.5}$, AP_{TLB}, and AP_{GMB}. The number of parameters for YOLOv5 of the fused GCnet was reduced to 6,262,778. We tentatively refer to YOLOv5 with the addition of ACmix, CBAM, and the replacement of SPPF with RFB as Model B. Model B with GCnet had a performance increase of 2.0%–3.5%, in terms of $AP_{0.5}$, AP_{TLB}, and AP_{GMB}. The number of parameters for Model B of the fused GCnet was reduced to 7,959,886. The model size of Model B after incorporating the GCNet was also reduced from 17.1 MB to 15.6 MB. The experiments show that the improved CSP1 based on GCnet not only reduces the number of parameters to make the model more lightweight, but also improves the performance of the model in detecting tea diseases and insect pests.

We eventually developed the YOLO-Tea tea tree leaf disease and pest detection model through a series of improvements. We will first further enhance YOLO-Tea's performance in subsequent studies. This is due to the fact that we discovered through ablation practice that YOLO-Tea still has shortcomings in the detection of smaller tea tree leaf diseases and insect pests. YOLO-Tea's AP_S and AR_S only improved by 3.1% and 0.3% compared to YOLOv5. These two figures show that there is still room for improvement in the detection of very small targets with YOLO-Tea. Second, based on the quantity and concentration of diseases and pests found, we will also create a system for evaluating the use of pesticides. Tea farmers will have a reference for pesticide dosing thanks to this system for assessing pesticide dose. Third, motivated by Lin's two deep learning bus route planning applications [35,36], we also intend to create a deep learning model for planning individual drones for pesticide spraying on tea plantations in our subsequent research. In addition, the method proposed by Xue et al. [37] allows direct modeling of the detailed distribution of canopy radiation at the plot scale. In our opinion, the method proposed by Xue et al. [37] may be a useful aid to our subsequent continued research on tea diseases and insect pests.

5. Conclusions

The yield and quality of tea leaves are significantly impacted by tea diseases. The precise control of tea diseases is facilitated by high-precision automatic detection and identification. However, because of the illumination, small targets, and occlusions in natural environments, deep learning methods tend to have low detection accuracy. In order to address these issues, we enhanced the YOLOv5s model in this paper.

First, we added the ACmix and CBAM modules to address the issue of false detection caused by small targets. We then enhanced retaining the global information of small tea disease and insect pest targets by swapping the SPPF module out for the RFB module. In order to reduce the number of parameters and boost performance, GCnet and YOLOv5s were finally combined.

To prove that our proposed YOLO-Tea model performs better than YOLOv5s, Faster R-CNN, and SSD, ablative experiments and comparison experiments were carried out. The experiment results show that our model has great potential to be used in real-world tea diseases monitoring applications.

The dataset used in this paper was taken in good afternoon light and does not take into account early morning and poor night light conditions for the time being. Further research will be conducted in the future to address the early morning and late night conditions. In addition, the ccd and other equipment may be affected by the high ambient temperature of the working environment in the afternoon. In future research, we will address these issues to further improve the performance of our model.

Author Contributions: Z.X. devised the programs and drafted the initial manuscript and contributed to writing embellishments. R.X. helped with data collection, data analysis, and revised the manuscript. H.L. and D.B. designed the project and revised the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by The Jiangsu Modern Agricultural Machinery Equipment and Technology Demonstration and Promotion Project (NJ2021-19) and The Nanjing Modern Agricultural Machinery Equipment and Technological Innovation Demonstration Projects (NJ [2022]09).

Data Availability Statement: Not applicable.

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

Abbreviations

CNNs	Convolutional neural networks
YOLOv5	You Only Look Once version 5
CBAM	Convolutional block attention module
SPPF	Spatial pyramid pooling fast
GCNet	Global context network
SLIC	Simple linear iterative cluster
SVM	Support vector machine
Faster R-CNN	Faster region-based convolutional neural networks
SSD	Single Shot Multibox Detector
ACmix	Self-attention and convolution
CSP1	Cross-stage partial 1
CSP2	Cross-stage partial 2
CSPNet	Cross-stage partial networks
CAM	Channel attention module
SAM	Spatial attention module
RFB	Receptive field block
SE	Squeeze-and-excitation
SNL	Simplified Nonlocal
NL	Nonlocal
P	Precision
R	Recall
AP	Average precision

IoU	Intersection over union
AR	Average recall

References

1. Hu, G.; Yang, X.; Zhang, Y.; Wan, M. Identification of tea leaf diseases by using an improved deep convolutional neural network. *Sustain. Comput. Inform. Syst.* **2019**, *24*, 100353. [CrossRef]
2. Bao, W.; Fan, T.; Hu, G.; Liang, D.; Li, H. Detection and identification of tea leaf diseases based on AX-RetinaNet. *Sci. Rep.* **2022**, *12*, 2183. [CrossRef] [PubMed]
3. Miranda, J.L.; Gerardo, B.D.; Tanguilig, B.T., III. Pest detection and extraction using image processing techniques. *Int. J. Comput. Commun. Eng.* **2014**, *3*, 189. [CrossRef]
4. Barbedo, J.G.A.; Koenigkan, L.V.; Santos, T.T. Identifying multiple plant diseases using digital image processing. *Biosyst. Eng.* **2016**, *147*, 104–116. [CrossRef]
5. Zhang, S.; Wu, X.; You, Z.; Zhang, L. Leaf image-based cucumber disease recognition using sparse representation classification. *Comput. Electron. Agric.* **2017**, *134*, 135–141. [CrossRef]
6. Hossain, S.; Mou, R.M.; Hasan, M.M.; Chakraborty, S.; Razzak, M.A. Recognition and detection of tea leaf's diseases using support vector machine. In Proceedings of the 2018 IEEE 14th International Colloquium on Signal Processing & Its Applications (CSPA), Penang, Malaysia, 9–10 March 2018.
7. Sun, Y.; Jiang, Z.; Zhang, L.; Dong, W.; Rao, Y. SLIC_SVM based leaf diseases saliency map extraction of tea plant. *Comput. Electron. Agric.* **2019**, *157*, 102–109. [CrossRef]
8. Chen, J.; Liu, Q.; Gao, L. Visual tea leaf disease recognition using a convolutional neural network model. *Symmetry* **2019**, *11*, 343. [CrossRef]
9. Hu, G.; Wu, H.; Zhang, Y.; Wan, M. A low shot learning method for tea leaf's disease identification. *Comput. Electron. Agric.* **2019**, *163*, 104852. [CrossRef]
10. Jiang, F.; Lu, Y.; Chen, Y.; Cai, D.; Li, G. Image recognition of four rice leaf diseases based on deep learning and support vector machine. *Comput. Electron. Agric.* **2020**, *179*, 105824. [CrossRef]
11. Sun, X.; Mu, S.; Xu, Y.; Cao, Z.; Su, T. Image recognition of tea leaf diseases based on convolutional neural network. In Proceedings of the 2018 International Conference on Security, Pattern Analysis, and Cybernetics (SPAC), Jinan, China, 14–17 December 2018.
12. Jiao, L.; Zhang, F.; Liu, F.; Yang, S.; Li, L.; Feng, Z.; Qu, R. A survey of deep learning-based object detection. *IEEE Access* **2019**, *7*, 128837–128868. [CrossRef]
13. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*. [CrossRef]
14. Zhou, G.; Zhang, W.; Chen, A.; He, M.; Ma, X. Rapid detection of rice disease based on FCM-KM and faster R-CNN fusion. *IEEE Access* **2019**, *7*, 143190–143206. [CrossRef]
15. 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.
16. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016.
17. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017.
18. Tian, Y.; Yang, G.; Wang, Z.; Wang, H.; Li, E.; Liang, Z. Apple detection during different growth stages in orchards using the improved YOLO-V3 model. *Comput. Electron. Agric.* **2019**, *157*, 417–426. [CrossRef]
19. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
20. Roy, A.M.; Bose, R.; Bhaduri, J. A fast accurate fine-grain object detection model based on YOLOv4 deep neural network. *Neural Comput. Appl.* **2022**, *34*, 3895–3921. [CrossRef]
21. Bochkovskiy, A.; Wang, C.; Liao, H.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
22. Sun, C.; Huang, C.; Zhang, H.; Chen, B.; An, F.; Wang, L.; Yun, T. Individual tree crown segmentation and crown width extraction from a heightmap derived from aerial laser scanning data using a deep learning framework. *Front. Plant Sci.* **2022**, *13*, 914974. [CrossRef]
23. Dai, G.; Fan, J. An industrial-grade solution for crop disease image detection tasks. *Front. Plant Sci.* **2022**, *13*, 921057. [CrossRef]
24. Pan, X.; Ge, C.; Lu, R.; Song, S.; Chen, G.; Huang, Z.; Huang, G. On the integration of self-attention and convolution. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022.
25. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018.
26. Liu, S.; Huang, D. Receptive field block net for accurate and fast object detection. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018.
27. Cao, Y.; Xu, J.; Lin, S.; Wei, F.; Hu, H. Gcnet: Non-local networks meet squeeze-excitation networks and beyond. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, Seoul, Republic of Korea, 27–28 October 2019.
28. Lu, Y.H.; Qiu, F.; Feng, H.Q.; Li, H.B.; Yang, Z.C.; Wyckhuys KA, G.; Wu, K.M. Species composition and seasonal abundance of pestiferous plant bugs (Hemiptera: Miridae) on Bt cotton in China. *Crop Prot.* **2008**, *27*, 465–472. [CrossRef]

29. Qian, J.; Lin, H. A Forest Fire Identification System Based on Weighted Fusion Algorithm. *Forests* **2022**, *13*, 1301. [CrossRef]
30. Wang, C.Y.; Liao HY, M.; Wu, Y.H.; Chen, P.Y.; Hsieh, J.W.; Yeh, I.H. CSPNet: A new backbone that can enhance learning capability of CNN. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020.
31. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path aggregation network for instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.
32. Wang, X.; Girshick, R.; Gupta, A.; He, K. Non-local neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.
33. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.
34. Khasawneh, N.; Fraiwan, M.; Fraiwan, L. Detection of K-complexes in EEG signals using deep transfer learning and YOLOv3. *Clust. Comput.* **2022**, 1–11. [CrossRef]
35. Lin, H.; Tang, C. Intelligent Bus Operation Optimization by Integrating Cases and Data Driven Based on Business Chain and Enhanced Quantum Genetic Algorithm. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 9869–9882. [CrossRef]
36. Lin, H.; Tang, C. Analysis and optimization of urban public transport lines based on multiobjective adaptive particle swarm optimization. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 16786–16798. [CrossRef]
37. Xue, X.; Jin, S.; An, F.; Zhang, H.; Fan, J.; Eichhorn, M.P.; Jin, C.; Chen, B.; Jiang, L.; Yun, T. Shortwave radiation calculation for forest plots using airborne LiDAR data and computer graphics. *Plant Phenom.* **2022**, *2022*, 9856739. [CrossRef]