agronomy

MDPI

Article

Improved YOLOv8 and SAHI Model for the Collaborative Detection of Small Targets at the Micro Scale: A Case Study of Pest Detection in Tea

Rong Ye [1], Quan Gao [2], Ye Qian [3], Jihong Sun [3] and Tong Li [2,3,*]

[1] College of Food Science and Technology, Yunnan Agricultural University, Kunming 650201, China; 15912913557@163.com
[2] Big Data College, Yunnan Agricultural University, Kunming 650201, China; gaoq@ynau.edu.cn
[3] The Key Laboratory for Crop Production and Smart Agriculture of Yunnan Province, Kunming 650201, China; qy198403@163.com (Y.Q.); sjh918a@163.com (J.S.)
* Correspondence: tli@ynu.edu.cn

Abstract: Pest target identification in agricultural production environments is challenging due to the dense distribution, small size, and high density of pests. Additionally, changeable environmental lighting and complex backgrounds further complicate the detection process. This study focuses on enhancing the recognition performance of tea pests by introducing a lightweight pest image recognition model based on the improved YOLOv8 architecture. First, slicing-aided fine-tuning and slicing-aided hyper inference (SAHI) are proposed to partition input images for enhanced model performance on low-resolution images and small-target detection. Then, based on an ELAN, a generalized efficient layer aggregation network (GELAN) is designed to replace the C2f module in the backbone network, enhance its feature extraction ability, and construct a lightweight model. Additionally, the MS structure is integrated into the neck network of YOLOv8 for feature fusion, enhancing the extraction of fine-grained and coarse-grained semantic information. Furthermore, the BiFormer attention mechanism, based on the Transformer architecture, is introduced to amplify target characteristics of tea pests. Finally, the inner-MPDIoU, based on auxiliary borders, is utilized as a replacement for the original loss function to enhance its learning capacity for complex pest samples. Our experimental results demonstrate that the enhanced YOLOv8 model achieves a precision of 96.32% and a recall of 97.95%, surpassing those of the original YOLOv8 model. Moreover, it attains an mAP@50 score of 98.17%. Compared to Faster R-CNN, SSD, YOLOv5, YOLOv7, and YOLOv8, its average accuracy is 17.04, 11.23, 5.78, 3.75, and 2.71 percentage points higher, respectively. The overall performance of YOLOv8 outperforms that of current mainstream detection models, with a detection speed of 95 FPS. This model effectively balances lightweight design with high accuracy and speed in detecting small targets such as tea pests. It can serve as a valuable reference for the identification and classification of various insect pests in tea gardens within complex production environments, effectively addressing practical application needs and offering guidance for the future monitoring and scientific control of tea insect pests.

Keywords: small object detection; BiFormer; YOLOv8; SAHI; GELAN; tea pest damage

Citation: Ye, R.; Gao, Q.; Qian, Y.; Sun, J.; Li, T. Improved YOLOv8 and SAHI Model for the Collaborative Detection of Small Targets at the Micro Scale: A Case Study of Pest Detection in Tea. *Agronomy* **2024**, *14*, 1034. https://doi.org/10.3390/agronomy14051034

Academic Editor: Gniewko Niedbała

Received: 22 April 2024
Revised: 7 May 2024
Accepted: 8 May 2024
Published: 13 May 2024

1. Introduction

Tea, as a significant economic crop in China, has a rich cultivation history intertwined with cultural significance [1]. However, the expansion of tea planting areas has led to a notable increase in tea insect pests, resulting in detrimental effects on both tea yield and quality, ultimately impacting the profits of tea farmers. Currently, farmers heavily rely on insect taxonomists for pest identification and diagnosis, a manual approach that presents numerous challenges including time-consuming processes, labor-intensive methods, and the potential for misjudgments. These difficulties may result in incorrect prevention and

control measures, ultimately harming tea production [2]. As a solution, the development of an efficient and swift pest detection and identification method is crucial for the tea industry. Such an advancement could enhance tea yield and quality, while simultaneously reducing economic losses [3].

Deep learning, an emerging field within machine learning, utilizes neural networks to mimic the learning and analysis capabilities of the human brain. By harnessing big data and powerful computing resources, deep learning enables computers to learn from data and make decisions through pattern recognition [4]. One of the key advantages of deep learning over traditional machine learning is its ability to automatically extract features, provided there are sufficient training data available [5]. In recent years, deep learning has found widespread applications across various sectors, particularly in agriculture [6–8]. Target detection using deep learning has gained prominence in computer vision research and is extensively employed in crop harvesting [9,10], pest and disease detection [11–13], yield prediction [14–16], unmanned farm monitoring [17,18], and other areas. Through the development of intricate parallel models, deep learning technology has effectively addressed challenges such as limited data resources, information integration difficulties, and the low efficiency of knowledge utilization in agricultural settings.

In recent years, with the rapid development of deep learning and computer vision technology, China's agriculture has ushered in a new era of wisdom. Deep learning methods are increasingly prevalent in crop disease identification research. Currently, two-stage target detection methods like Faster-RCNN [19,20] and one-stage target detection methods like SSD [21] and the YOLO series [22–25] are commonly utilized for crop pests and related issues. Researchers are actively enhancing these algorithm models and striving to implement them in the classification, detection, and identification of crop diseases. Li et al. [26,27] proposed a new Yolov7-TSA lightweight network architecture, which replaced the loss function and integrated the coordinate attention mechanism, achieving good results in the detection and classification of tea diseases. Fuentes et al. [28] designed a detection network for complex plant environments, demonstrating success in tomato pest and disease detection. Dai et al. [29] introduced SWin Transformer and Transformer mechanisms in pest detection, enhancing network robustness and effectiveness. Soeb et al. [30] developed a tea pest and disease dataset, highlighting YOLOv7 as the top performer in target detection and recognition. Deng et al. [31] enhanced YOLOv5 and YOLOv7-tiny for mobile terminal use, enabling the fast and efficient on-site diagnosis of six common pests and diseases.

Current research on small-target object detection primarily focuses on enhancing mainstream target detection network models through methods such as multi-scale feature fusion [31], super resolution [32], context information learning [33], and attention mechanisms [34]. Some studies have introduced new model structures or optimization methods, like the SSAM attention module and MPFPN structure [35], as well as the DW-YOLO model [36], to enhance the accuracy of detecting small target objects. However, existing target detection algorithms still exhibit limitations when it comes to small-target pests in tea. During the training process, deepening network layers can result in the loss of edge information and other features of small detection targets. Moreover, the occlusion caused by vegetation and leaves hinders the visibility of pest targets, impacting the feature extraction capabilities of computer vision models. This occlusion not only complicates insect disease detection but also diminishes model accuracy. Despite significant progress in target detection algorithms, the research on small-target detection faces numerous challenges at its current stage. These challenges mainly include the following:

1. The visual features of small targets may be unclear due to less important feature information and low image resolution;
2. In object detection tasks, extracting effective features is crucial. The quality of feature extraction directly affects the accuracy of detection results. Compared with large-scale targets, the features of small targets are more difficult to extract, which brings certain difficulties to the detection task. In the detection model, after the pooling operation, some important features of small targets may be lost, thus increasing the challenge of detection;

3. In complex environments, small-target detection is hindered by factors like illumination, occlusion, and aggregation, which make it challenging to differentiate the target from the background or similar targets. Consequently, addressing complex background interference is a crucial challenge in small-target detection.

In order to improve the small-target detection performance of a network for tea insect infestations, we introduced an improved YOLOv8 lightweight insect infestation image recognition model that incorporates slice-assisted fine-tuning and reasoning (SAHI) techniques for image slicing. Additionally, a generalized high-efficiency layer aggregation network (GELAN) is designed to replace the C2f module in the backbone network, thereby enhancing feature extraction capabilities and constructing a lightweight model. Furthermore, the neck network of YOLOv8 utilizes an MS structure for feature fusion to improve the extraction of fine-grained and coarse-grained semantic information. The integration of a BiFormer attention mechanism based on the Transformer architecture strengthens target features related to tea insect pests. Finally, an inner-MPDIoU based on the auxiliary frame is employed as a replacement for the original loss function, aiming to enhance its learning ability for complex pest samples and ultimately improve its detection accuracy for small pest targets.

2. Materials and Methods

2.1. Introduction to YOLOv8

YOLOv8 is the current newer YOLO model, which is divided into different versions according to the different depths and widths of the network, including YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8, and YOLO8x. The network structure of these versions is basically the same; the only difference is the depth and width [37]. The YOLOv8 model mainly includes three key network layers: the backbone, neck, and head. The model initially preprocesses the input image via the input layer, followed by feature extraction through the backbone layer. Subsequently, the extracted features are fed into the neck layer, which merges features of varying scales to create a feature pyramid for enhanced information. Ultimately, the prediction results are produced by the head layer. The network architecture of YOLOv8 is illustrated in Figure 1.

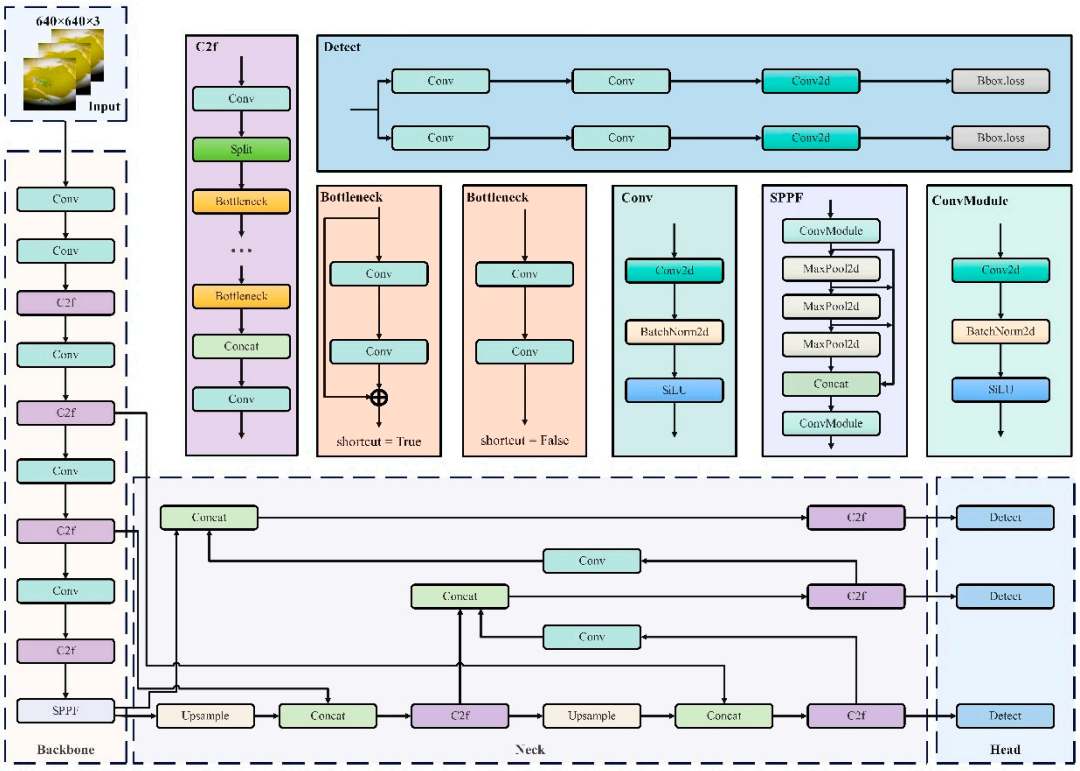

Figure 1. YOLOv8 network architecture.

2.2. Improving YOLOv8 Network Structure

The utilization of deep learning networks has practical significance in detection tasks within various complex practical scenarios [38]. This article presents enhancements based on YOLOv8 to efficiently detect small targets of tea pests in complex environments. The enhanced structure of SAHI-YOLOv8 is illustrated in Figure 2. Its main improvements are as follows.

1. In order to solve the problem of detecting small targets in high-resolution images while maintaining a high level of memory utilization, slicing-aided fine-tuning and slicing-aided hyper inference (SAHI) are introduced to slice the input network image, resulting in larger pixel regions for small-target objects, improving the effectiveness of network inference and fine-tuning, and providing more detailed features for subsequent models;
2. A new lightweight network architecture based on gradient path planning, Generalized Efficient Layer Aggregation Network (GELAN), has been designed to replace the C2f module in the backbone network, simplify the backbone network structure, enhance its feature extraction capabilities, and achieve model lightweighting;
3. Secondly, in the neck network of YOLOv8, the MS structure is used for feature fusion. In the first stage of the encoder, the smallest kernel convolution is used, while the largest kernel convolution is used in the final stage, which is consistent with the increment of feature resolution. This improves the extraction of fine-grained and coarse-grained semantic information, enhances the multi-scale feature representation ability of the encoder, and further improves the performance of the model in identifying different scales, complex backgrounds, and small targets;
4. On the basis of these improvements, an inner-MPDIoU is used as the loss function for the model bounding box regression to further enhance its learning ability for small-target samples.

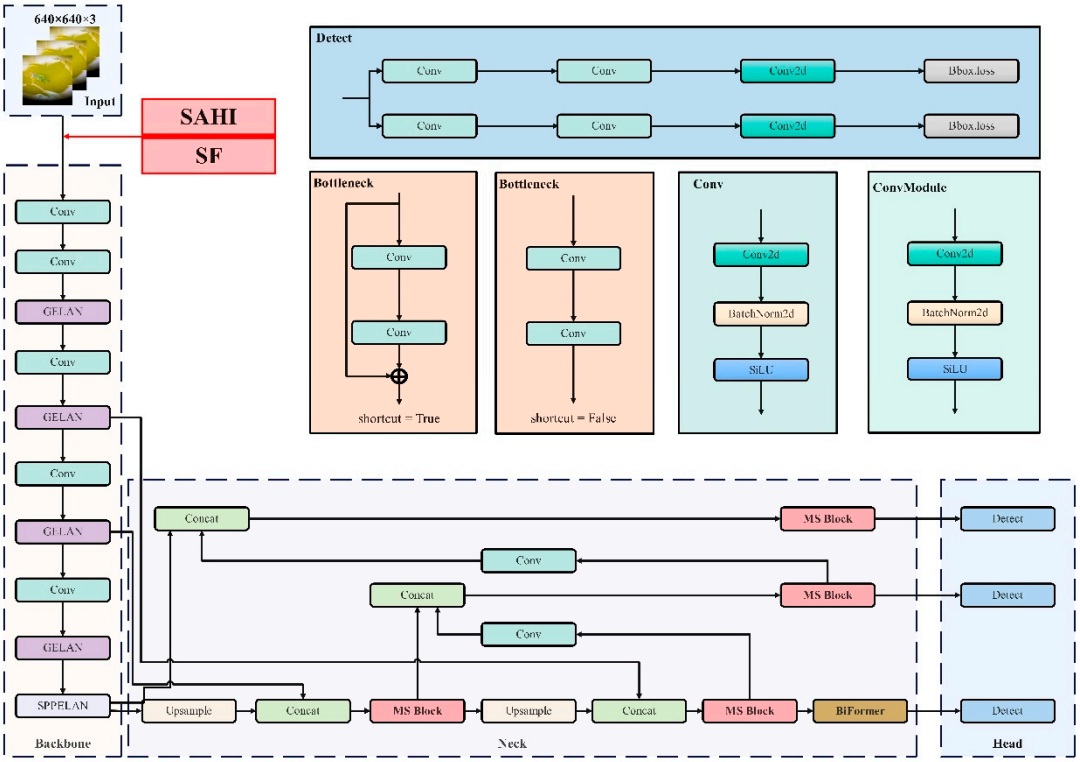

Figure 2. The overall structure of the SAHI-YOLOv8 network.

2.3. Slicing-Aided Module

In order to enhance the detection accuracy of the improved YOLOv8 algorithm for small pest targets, we introduce slicing-aided hyper inference and fine-tuning for a small object detection (SAHI) network. This approach involves two key steps: slicing-aided fine-tuning and slicing-aided hyper inference. These steps create a larger pixel area for small target objects in images, preserving the unique features of small targets and preventing feature loss. Additionally, the method maintains its level of detection accuracy for large targets, ultimately leading to an improved overall detection performance. The innovative SAHI network effectively reduces missed detections and accurately captures the characteristics of small pest targets.

In the field of target detection, one of the most effective ways to improve model detection performance is through data augmentation. Data augmentation not only increases the number of images, but also alters the complexity of labeling and predicting targets. However, when dealing with small-scale targets that have small sizes and low resolutions, there is a risk of blurred edge details, which may not meet the requirements. Additionally, image compression and noise can further complicate small-target detection. To address this issue, this study introduces a slicing strategy that involves using a fixed-size sliding window to segment the original image, conducting target detection on each segmented small picture, and finally overlaying the detection boxes on the original image and using NMS for filtering. This approach proves to be effective in enhancing its detection performance for small targets. The specific structure is illustrated in Figure 3.

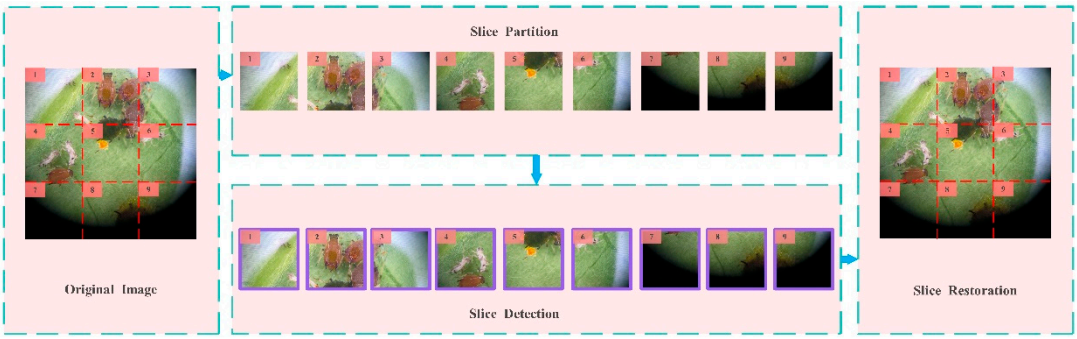

Figure 3. Schematic diagram of slicing-assisted reasoning.

2.3.1. Slicing-Aided Fine-Tuning

Usually, small-scale targets have a small proportion of regions in the image, and due to the low resolution and poor expression ability of edge detail feature information after model input compression, subsequent model training cannot produce good feedback on the prediction results of small targets. Based on the above issues, SA uses slicing-aided fine-tuning to enhance the data of the original dataset, in order to enhance the relative proportion of small-target samples in the dataset. By slicing, the proportion of small targets in the original image is increased, thereby improving the detection model's ability to detect small targets.

Fine-tuning is the process of initializing a network by adjusting the parameters of the first few layers of the output layer using known network structures and training parameters. This method fully utilizes the generalization ability of deep neural networks, while avoiding complex model design and long-term training. Therefore, fine-tuning is currently a relatively suitable choice [39]. The process of slicing-assisted fine-tuning is shown in Figure 4.

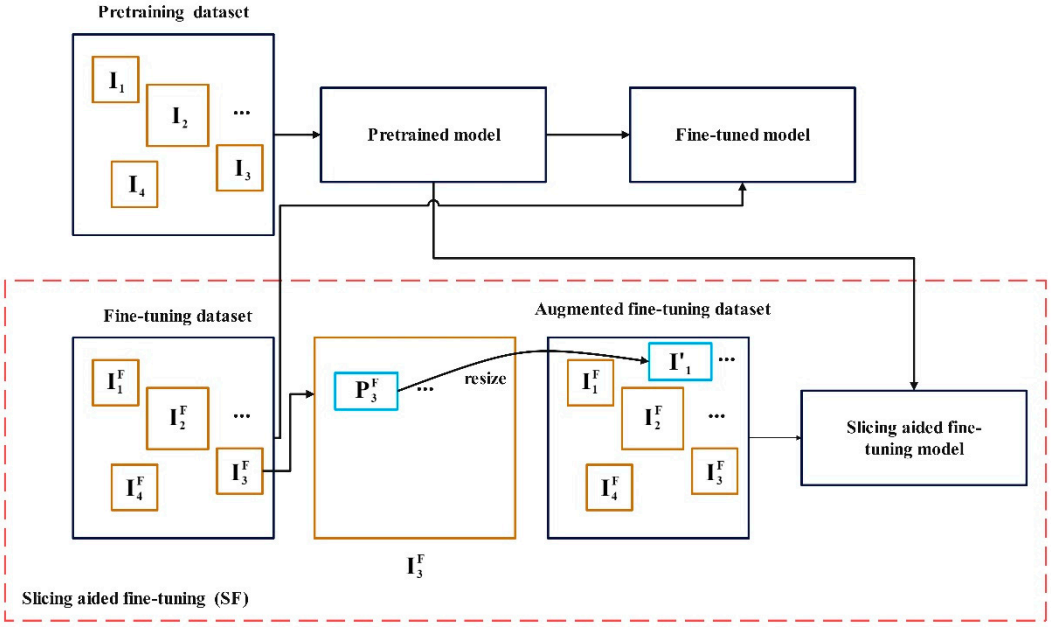

Figure 4. Principle of slicing-aided fine-tuning.

As shown in Figure 4, firstly, the SF process extracts some overlapping patch blocks $P_1^F, P_2^F, \ldots, P_l^F$ from the image $I_1^F, I_2^F, \ldots, I_l^F$ in the dataset using slice boxes and assists in fine-tuning the network initialization parameters by zooming in on small targets. The specific principle is to separate the input image into overlapping patches: (the superscript F indicates fine-tuning). Select M and N as hyperparameters within the pre-defined range of $[M_{min}, M_{max}]$ and $[N_{min}, N_{max}]$. By adjusting the patch size while maintaining the aspect ratio, the image width is maintained at [800, 1333] pixels during the fine-tuning process, resulting in an enhanced image $I_1', I_2', \ldots, I_k'$ that is larger than the sun scale in the original image. By partially scaling up the dataset images, more small target images were added, thus solving the problem of having insufficient edge detail feature information for small targets. This method significantly improves the robustness of the model in small-scale detection.

2.3.2. Slicing-Aided Hyper Inference

Unlike the SF method of focusing on small targets with different relative scales in the original image, SAHI enhances the feature detection ability of small targets in local areas by enabling the detection network to scan image features more finely. SAHI uses image segmentation to segment high-resolution images into resolutions suitable for model detection, starting from the resolution of the image itself. This processing can be seen in Figure 5. SAHI adopts a detection strategy similar to sliding windows, dividing the original input image I into I overlapping slices $P_1^{\mathrm{I}}, P_2^{\mathrm{I}}, \cdots, P_{\mathrm{I}}^{\mathrm{I}}$ of size $p \times q$ (superscript I represents the inference process). Each overlapping slice will separately apply forward-propagation object detection to infer small targets, while utilizing full inference (FI) of the entire image to detect large targets. Finally, SAHI will summarize the results of both full inference and local inference, perform post-processing of detection boxes through NMS to match prediction boxes that meet the IoU set threshold, and remove boxes with low IoU. Through this approach, SAHI effectively solves the problem of small-target detail blur that may occur when processing high-resolution images in object detection models. At the same time, it improves detection performance through auxiliary reasoning and achieves this goal without adding additional parameters.

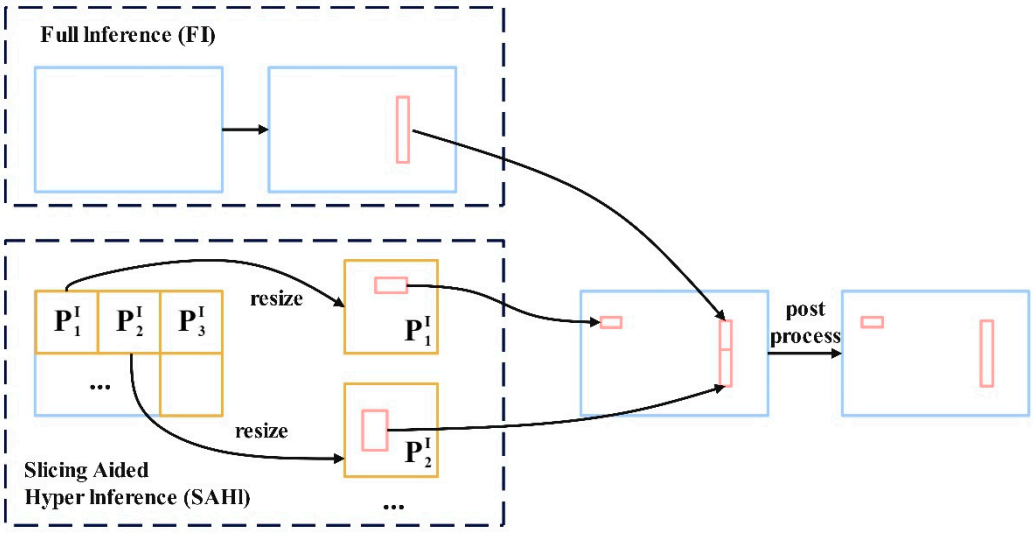

Figure 5. Principle of slice-assisted reasoning.

2.4. Generalized ELAN

In deep neural networks, the issue of information bottleneck [40] is a common challenge in which input data may lose information during the feedforward process. To address this, current methods include using a reversible architecture [41] to repeatedly input data and explicitly maintain input data information, utilizing reconstruction loss to maximize feature extraction through implicit methods and preserve input information, and introducing deep supervision [42] to establish mapping from features to targets using shallow features to ensure important information can be transferred to deeper levels. However, all of these methods have certain limitations during both training and inference stages. For example, reversible architectures often require additional convolutional layers to process input data, leading to increased inference costs. This limitation hinders the effective modeling of high-order semantic information during training. The deep supervision mechanism can result in error accumulation, while shallow supervision may cause information loss, preventing subsequent layers from accessing necessary information. These challenges are particularly pronounced in complex tasks and small-target models. To address these issues, researchers have introduced a new concept called programmable gradient information, which builds upon ELAN [43] to create GELAN. This design considers parameters, computational complexity, accuracy, and inference speed. Following the information bottleneck principle, there may be information loss when converting image X, as indicated in Equation (1):

$$I(X, X) \geq I(X, f_\theta(X)) \geq I\left(X, g_\phi(f_\theta(X))\right) \quad (1)$$

Among these, I represents mutual information, f and g are conversion functions, and θ and ϕ are the parameters of f and g, respectively.

As the number of network layers increases, the likelihood of losing original data also increases, leading to incomplete information, unreliable gradients, and suboptimal convergence during network training. To address these issues, we introduce programmable gradient information (PGI) method, consisting of three components: the main branch, auxiliary reversible branch, and multi-level auxiliary information. As can be seen from Figure 6D, in the inference process of PGI, only the main branch is utilized, eliminating the need for additional inference costs. The auxiliary reversible branch is specifically designed to tackle challenges arising from the increased depth of neural networks, which can create information bottlenecks and hinder the generation of reliable gradients. Additionally, the multi-level auxiliary information component aims to mitigate the problem of error accumulation resulting from deep supervision.

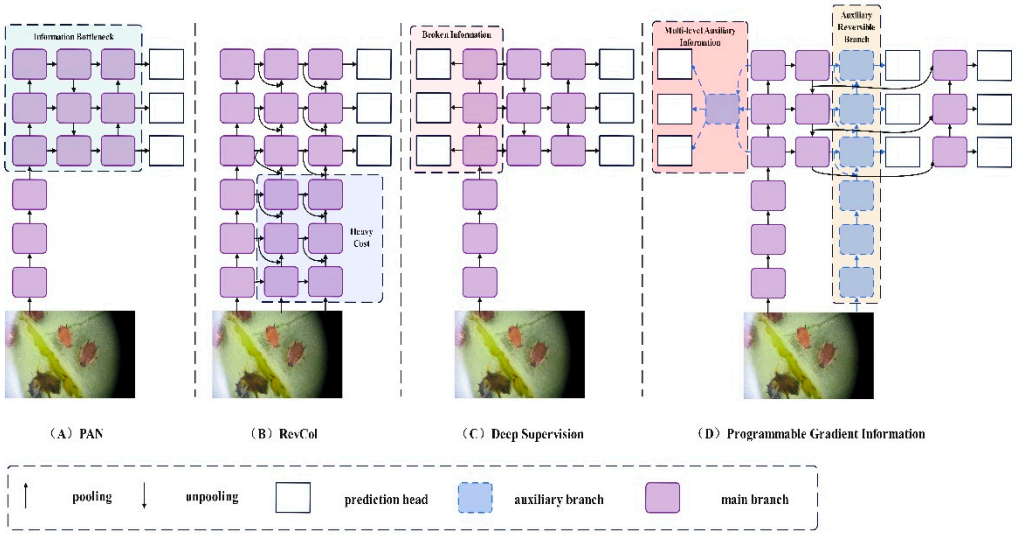

Figure 6. PGI and related network architectures and methods.

Subsequently, in order to achieve object detection, different feature pyramids can be used to perform different tasks. Two neural network architectures for gradient path planning, CSPNet and ELAN, are combined to establish an efficient layer aggregation network (GELAN). GELAN is a balance between being lightweight and having a good inference speed and accuracy. Its overall architecture is shown in Figure 7.

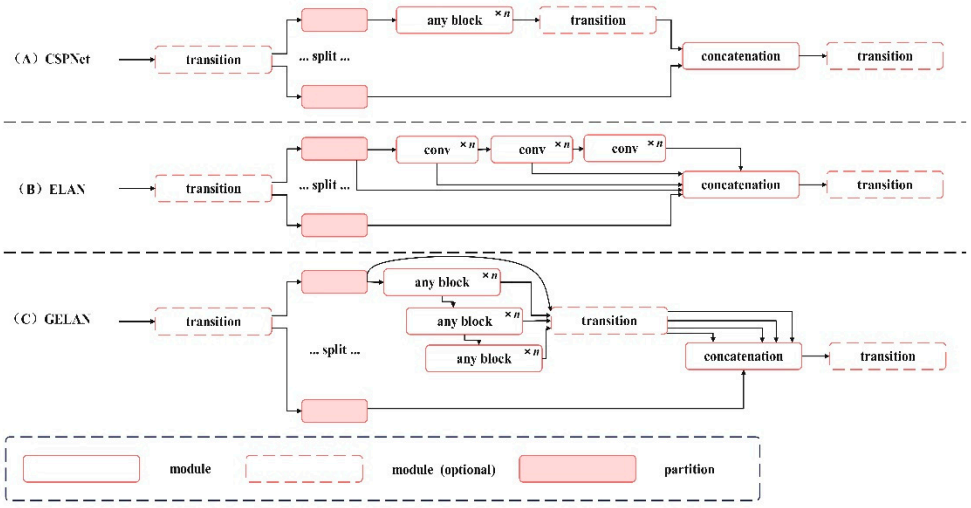

Figure 7. The architecture of GELAN.

2.5. Transformer

In recent years, Transformer [44] has shown significant advancements in both natural language processing and computer vision. In computer vision, models like ViT [45] and DETR [46], based on Transformer, have been highly successful. However, small-target detection tasks present challenges due to limited context information in traditional CNN models when dealing with long-distance small targets. On the other hand, Transformer's self-attention mechanism excels in aggregating global information in small-target detection tasks, capturing long-range dependencies between objects, and effectively modeling the relationship between location information and objects. The Transformer model leverages the self-attention mechanism to process input data with weighted interactions, enabling the capture of long-range dependencies. However, this approach involves complex calculations for each input element and all others, leading to significant computational and memory requirements. To preserve contextual feature information effectively, our research proposes

incorporating long-range dependency modeling to mitigate the impact of noise interference in real-world settings. BiFormer, a variant of the Transformer model, introduces a bi-level routing attention mechanism to allocate computing resources and perceptual features more flexibly. The BRA attention mechanism has shown to be highly effective in detecting small targets due to its dynamic sparse attention mechanism. This mechanism establishes correlation and information interaction between tasks by incorporating two levels of attention mechanisms. Initially, the mechanism filters out irrelevant information at a coarse area level, retaining only a small portion of the routing area. This helps reduce interference from invalid tasks and enhances interaction between relevant information. Subsequently, fine-grained token-to-token attention is applied in these routing areas, allowing for deep interactions between related tasks and obtaining more effective feature information. In comparison to traditional attention mechanisms, BiFormer can adjust the distribution of attention more flexibly and dynamically, adapting to different scales and complexities of input image content. This enables us to capture the characteristics of small targets more accurately.

As shown in Figure 8, the intra task attention is first calculated to weight the features within the task. Assist the model in selecting the most important features in each task. The specific implementation process is as follows:

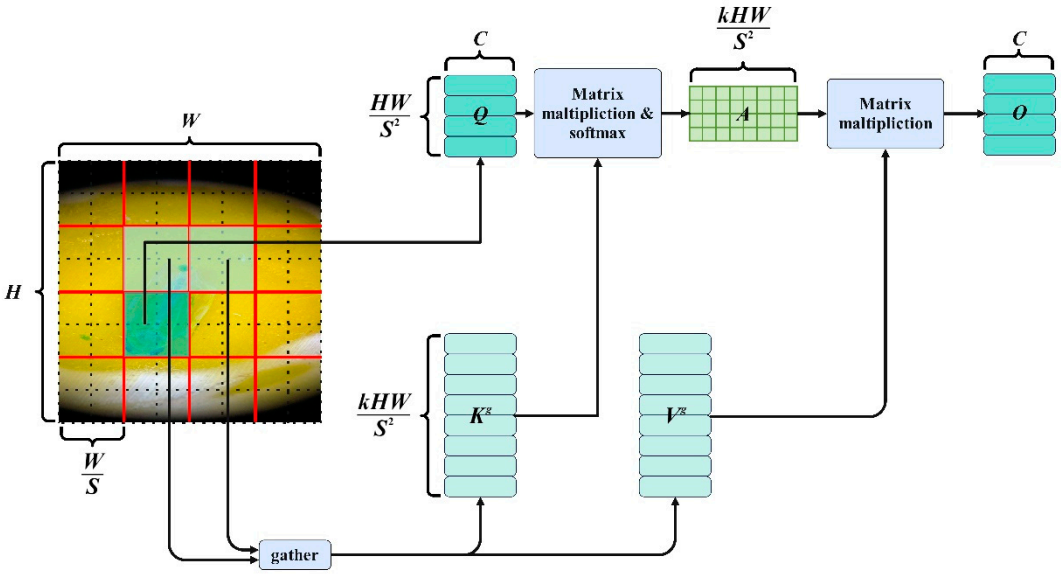

Figure 8. BRA attention mechanism.

For a given 2D feature map $X \in R^{H \times W \times C}$, divide it into $S \times S$ non-overlapping regions, such that each region contains a feature vector $\frac{HW}{S^2}$. Determine $Q, K, V \in R^{S^2 \times \frac{HW}{S^2} \times C}$, and the corresponding linear prediction is as follows:

$$Q = X^r W^q, K = X^r W^k, V = X^r W^v \tag{2}$$

Among these, $W^q, W^k, W^v \in R^{C \times C}$ represents the projection weights of Q, K, V.

Then, by constructing a directed graph, the participation relationship between regions is determined. Firstly, by taking the average of Q and deriving K, $Q^r, K^r \in R^{S^2 \times C}$, the adjacency matrix A^r is obtained using the following formula:

$$A^r = Q^r (K^r)^T \tag{3}$$

Among these, represents the degree of correlation between input information and content areas. Next, the correlation graph is modified by retaining high correlation for each area using the following index matrix:

$$I^r = topkIndex(A^r) \tag{4}$$

Among these, the i-th row of I^r contains the K feature map regions most relevant to the i-th region. This process achieves information exchange between different target tasks, successfully filtering out information that is highly correlated with the detection target and making this information dominant in the model learning process, while other irrelevant information is suppressed, effectively allocating attention.

After obtaining the correlation matrix I^r within the task, attention calculation between tasks can be performed as follows:

$$K^g = gather(K, I^r), V^g = gather(V, I^r) \tag{5}$$

Among these, $K^g, V^g \in R^{S^2 \times \frac{kHW}{S^2} \times C}$, and the collected information is as follows:

$$O = Attention(Q, K^g, V^g) \tag{6}$$

Through the BiFormer routing attention dual-layer routing mechanism, the model can learn the correlation and dependency relationships between different tasks and adaptively allocate attention weights, thereby improving the effectiveness and generalization ability of multi-task learning. It helps the model focus on task-specific important features, promotes information exchange between tasks, reduces overfitting, and improves overall learning performance.

The BiFormer block structure constructed by combining the dual-layer routing attention mechanism is shown in Figure 9. At the beginning, the relative position information of the input is implicitly encoded using depthwise separable convolution (DWConv), and then the dual-layer routing attention mechanism and multi-layer perceptron (MLP) module are sequentially used to model the cross-positional relationship and embed the input information position by position.

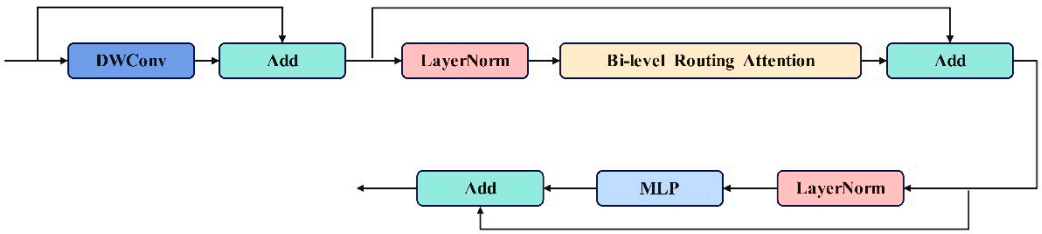

Figure 9. BiFormer block.

2.6. Neck Improvement

Structural design is a key aspect in YOLO model development, significantly impacting model performance. YOLOv4 [47] utilizes cross-stage partial connection (CSPNet) [48] to enhance DarkNet for improved performance. YOLOv6 [49] and PPYOLOE [50] explore reparameterization technology to boost model accuracy without increasing inference costs. YOLOv7 [51] introduces a novel network structure called Extended Efficient Layer Aggregation Network (E-ELAN) to enhance learning efficiency and convergence speed by managing gradient path length. RTMDet [52] incorporates large kernel convolution (5×5) to enhance feature extraction in basic blocks, enabling better context modeling and significantly improving model accuracy.

CSP Block is a network based on stage-level gradient paths that balances gradient combination and computational costs. By dividing channel dimensions to segment gradient flows, gradient flows are propagated through different network paths to enhance gradient

performance. The information flow of gradients is integrated into feature maps from beginning to end, reducing the amount of multiplication and addition operations while ensuring accuracy. Channel rearrangement is aimed at solving the problem of network performance degradation caused by stacking multiple depthwise separable convolutions. In the ELAN block structure, the Concat operation is used to parallelize the results of different CBS convolution treatments, achieving a significant increase in the number of channels, thereby enhancing the expression ability of the image's own features without increasing the information content of each feature. Figure 10 shows the structure of the original CSP block and ELAN block, located in the upper two rows of Figure 10, respectively. Among them, dashed boxes represent deep convolution; N represents the number of convolutional layers; K represents the size of the convolution kernel; and C represents the number of channels.

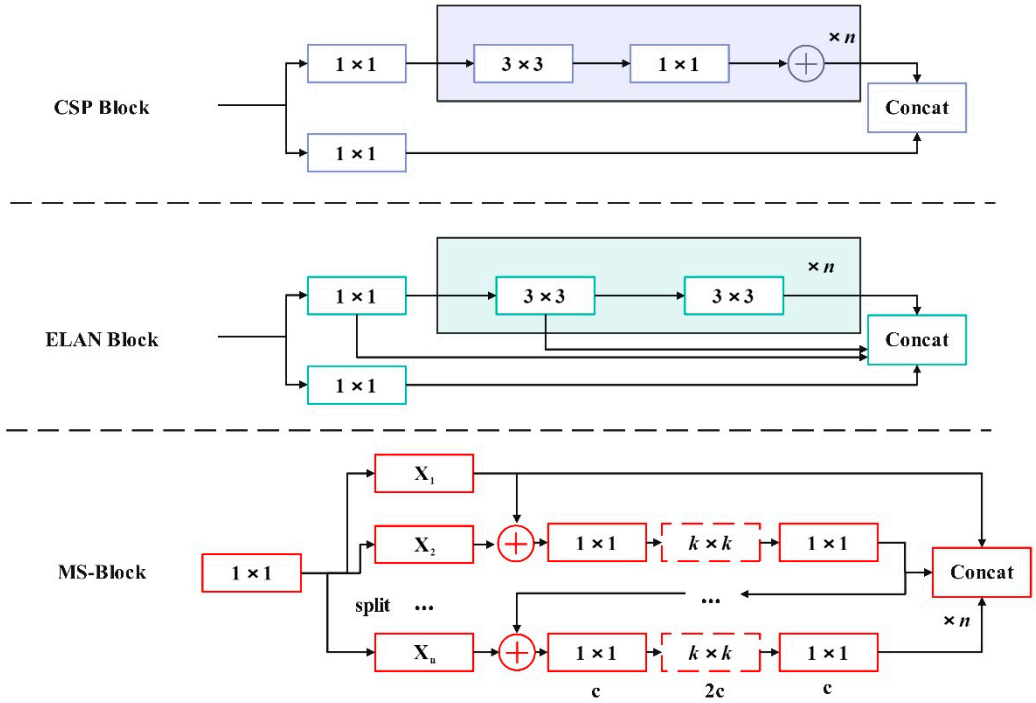

Figure 10. Comparison of three building block models.

The original YOLOv8 model has three detection heads with corresponding neck layer feature map sizes of 80 × 80, 40 × 40, and 20 × 20, each responsible for detecting small-, medium-, and large-scale targets. However, YOLOv8's detection head has limitations in detecting small targets. Adding more feature maps would increase the number of network layers in the neck layer, resulting in more network parameters and calculations. This article focuses on enhancing expressiveness through the use of multi-scale feature representation to improve real-time object detection. The introduction of MS block, depicted in Figure 10, aims to reduce model complexity and the number of calculations while maintaining accuracy.

Based on previous research, we propose a new block with a hierarchical feature fusion strategy [53], called MS block (multi-scale building block), to enhance the ability of real-time object detectors to extract multi-scale features while maintaining fast reasoning speed. Assume $X \in R^{H \times W \times C}$ is the input feature. After undergoing 1×1 convolutional transformation, the channel dimension of X increases to $n \times C$. Then, X is divided into n different channels, represented as $\{X_i\}$, where $i \in 1, 2, 3, \ldots, n$. To reduce computational costs, the number of convolutional layers n is set to 3. Among them, except for X_1, all other channels pass through a reverse bottleneck layer, represented by $IB_{k \times k}(\cdot)$, where

K represents the kernel size. The mathematical representation of Y_i can be described as follows:

$$Y_i = \begin{cases} X_i, & i = 1 \\ IB_{k\times k}(Y_{i-1} + X_i). & i > 1 \end{cases} \tag{7}$$

According to the formula, we do not connect the reverse bottleneck layer to X_1, allowing it to act as a cross-stage connection and retain information from previous layers. Finally, we connect all the segments together and apply 1×1 convolution to enable interactions between them, with each segment encoding features of different scales. When the network deepens, this 1×1 convolution is also used to adjust the number of channels.

2.7. Inner-MPDIOU

Object detection networks typically comprise two main components: bounding box regression and category discrimination. Bounding box regression is responsible for predicting the location and size of the target using a regression network. The accuracy of bounding box regression directly impacts the quality of detection results. However, in single-stage object detection models based on deep learning, existing bounding box regression losses may not adequately capture changes in bounding box positional relationships. Therefore, a carefully crafted loss function is essential for accurately regressing bounding boxes, particularly in the context of single-stage object detection models based on deep learning.

Early target detection primarily relied on L1 loss, specifically the mean absolute error (MAE), for predicting the bounding box coordinates. In recent years, researchers have introduced a range of loss functions based on Intersection over Union (IoU) metrics. These include IoU loss [54], Generalized Intersection over Union (GIoU) loss [38], Distance Intersection over Union (DIoU) loss [39], Complete Intersection over Union (CIoU) loss, and Alpha-Complete Intersection over Union (Alpha-CIoU) loss [55], among others. Aside from Alpha-CIoU loss, other loss functions take into account the three elements of bounding box regression (overlap, center point distance, and aspect ratio) by incorporating corresponding penalty terms. However, various loss functions based on IoU still exhibit certain limitations: GIoU aims to minimize the disjoint area through regression but may revert to IoU loss when two boxes are included; the center point distance penalty in DIoU loss may not effectively address the overlap between bounding boxes; and the aspect ratio penalty in CIoU loss could impact the stability of bounding box regression. IoU (Intersection over Union) remains the predominant standard for evaluating prediction frame loss in the detection field, with its formula depicted in Equation (8):

$$IoU = \frac{\left|B^{pred} \cap B^{gt}\right|}{\left|B^{pred} \cup B^{gt}\right|} \tag{8}$$

In the formula, B and B^{gt} represent the prediction box and GT box, respectively. After defining IoU, their corresponding losses can be defined as follows:

$$L_{IoU} = 1 - IoU \tag{9}$$

The bounding box regression loss function based on IoU continues to iterate and develop, such as GIoU, DIoU, CIoU, EIoU, SIoU, etc. YOLOv8 uses CIoU, and its calculation formula is shown in Equation (2):

$$CIoU = IoU - \left(\frac{\rho^2\left(B^{pred}, B^{gt}\right)}{c^2} + \alpha v\right) \tag{10}$$

In this formula, $v = \frac{4}{\pi}\left(arctan\frac{w^{gt}}{h^{gt}} - arctan\frac{W^{pred}}{h^{pred}}\right)$ is used to measure the aspect ratio, $\alpha = \frac{v}{(1-IoU)+v}$, α represents the balance parameter, and $\rho^2\left(B^{pred}, B^{gt}\right)$ represents the Euclidean distance between the predicted box and the center point of the true box.

This study proposes using the MPDIoU loss function as a more suitable alternative for pest small-target detection, considering limitations of the CIoU loss function. The MPDIoU loss function aims to improve the alignment between predicted and real frames, which is particularly beneficial when center points do not overlap. Moreover, when center points align but length and width values differ, MPDIoU effectively penalizes discrepancies without degradation seen in IoU loss. By utilizing MPDIoU, not only is the calculation process simplified, but model convergence is stabilized and its detection accuracy for small pest targets is enhanced.

These improved loss functions still accelerate convergence by adding new loss terms, without realizing the limitations of IoU itself. Inner IoU [56] uses auxiliary bounding boxes to calculate IoU and improve its generalization ability. The specific calculation process is shown in Equations (11)–(14), in which the scale factor ratio is used to control the size of the auxiliary bounding boxes.

$$b_l = x_c - \frac{w * Ratio}{2} \tag{11}$$

$$b_r = x_c + \frac{w * Ratio}{2} \tag{12}$$

$$b_t = y_c - \frac{h * Ratio}{2} \tag{13}$$

$$b_b = y_c + \frac{h * Ratio}{2} \tag{14}$$

Equations (15)–(17) can perform a transformation on the center point of the detection box to obtain the corner vertices of the auxiliary detection box and perform corresponding transformations on both the predicted box and the real box output by the model. b^{gt} and b^{pred} represent the calculation results of the real box and the predicted box, respectively.

$$inter = \left(min\left(b_r^{gt}, b_r\right) - max\left(b_l^{gt}, b_l\right)\right) * \left(min\left(b_b^{gt}, b_b\right) - max\left(b_t^{gt}, b_t\right)\right) \tag{15}$$

$$union = \left(w^{gt} * h^{gt}\right) * (Ratio)^2 + (w * h) * (Ratio)^2 - inter \tag{16}$$

$$IoU^{inner} = \frac{inter}{union} \tag{17}$$

So, inner IoU actually calculates the IoU between auxiliary borders. When Ratio is [0.5, 1.5] and <1, the auxiliary bounding box is smaller than the actual bounding box, and the effective range of regression is smaller than the IoU loss. However, the absolute value of the gradient is larger than the gradient obtained by the IoU loss, which can accelerate the convergence of high-IoU samples. When Ratio > 1, the auxiliary bounding box is larger than the actual box, expanding the effective range of regression and benefiting low-IoU regression.

MPDIoU [57] is an improved algorithm that directly minimizes the distance between the predicted box and the true box corresponding to the upper-left and lower-right corner points. It can handle both overlapping and non-overlapping bounding boxes well, improving convergence speed, as shown in Figure 11.

$$MPDIoU = IoU - \frac{\rho^2\left(P_1^{pred}, P_1^{gt}\right)}{w^2 + h^2} - \frac{\rho^2\left(P_2^{pred}, P_2^{gt}\right)}{w^2 + h^2} \tag{18}$$

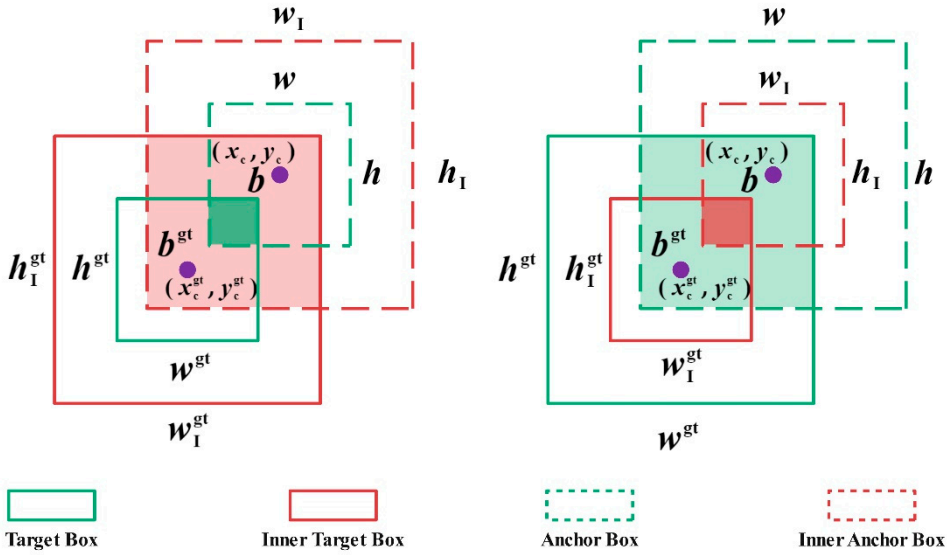

Figure 11. Inner IoU example structure.

In the formula, P_1^{pred}, P_2^{pred}, P_1^{gt}, and P_2^{gt} refer to the points in the upper-left and lower-right corners of the predicted box and the true box, respectively. $\rho^2\left(P_1^{pred}, P_1^{gt}\right)$ is used to calculate the distance between the corresponding points. Using the idea of inner IoU to transform MPDIoU and replacing the IoU calculation part can greatly improve detection performance.

$$MPDIoU^{inner} = IoU^{inner} - \frac{\rho^2\left(P_1^{pred}, P_1^{gt}\right)}{w^2 + h^2} - \frac{p^2\left(P_2^{pred}, P_2^{gt}\right)}{w^2 + h^2} \tag{19}$$

3. Results and Discussion

3.1. Dataset and Experimental Environment

In order to improve the reliability of the model, abundant pest datasets were collected in this study, including those from the Houshan Tea Garden Base of Yunnan Agricultural University, the Hekai Base of Xishuangbanna Prefecture, Yunnan Province, and the ancient tree Tea Base of Lincang City. The LabelImg software was utilized to label a total of 2864 images depicting tea pests under microscopic conditions. During the construction of the dataset, the images were divided into a training set and a test set in an 8:2 ratio to assess the model's generalization capability. Various characteristics of the tea leaves, such as their appearance, texture, color, and potential influencing factors, were observed and recorded. Subsequently, the data were preprocessed, annotated, and compiled into a comprehensive dataset suitable for training and testing purposes. The construction process of this dataset was meticulously documented and is reproducible. The pest images in the dataset encompass a range of scenarios, including (A) single targets, (B) multiple targets in the same category, (C) multiple targets in different classes, and various backgrounds, like (D) an insect plate, (E) a blade plate, and (F) blurred backgrounds, to evaluate the detection performance of the enhanced algorithm across different complexities, scales, and target sizes. The tea insect pest dataset is illustrated in Figure 12.

We cropped the images to contain the smallest rectangle around the disease to simplify the backgrounds. We saved the comments in XML format after editing the images. Figure 13 displays a visual analysis of the tea pest annotation file. It is evident from the figure that the target frame size ratio falls mainly between 0.08 and 0.1. These targets are relatively small compared to the overall image, indicating that this dataset typically contains small-target data. The labels are densely distributed and overlapping, with many small targets and significant scale variations.

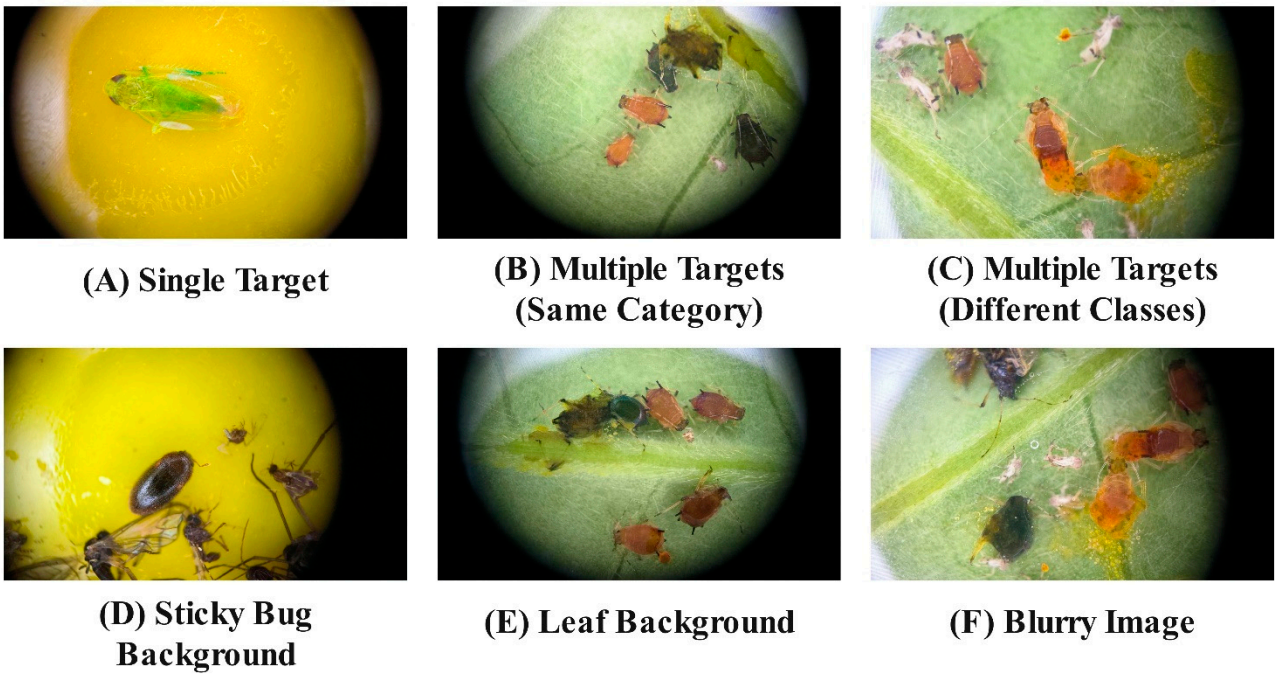

Figure 12. Tea pest image dataset.

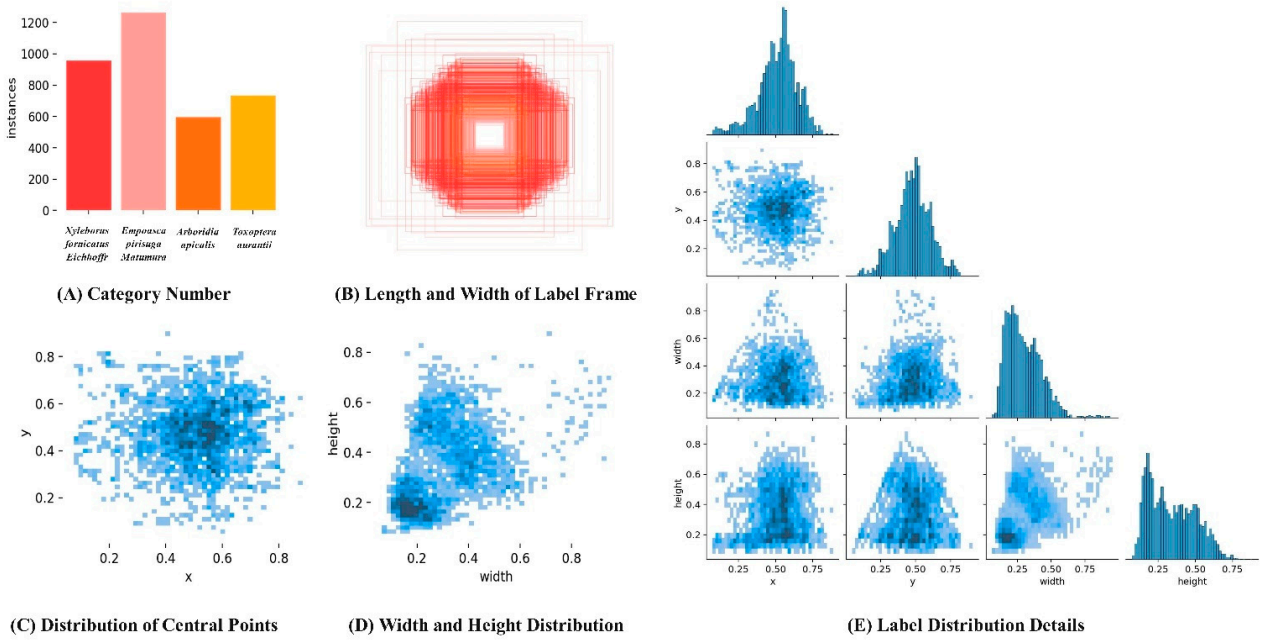

Figure 13. Visualization of training set files.

For model training, this article is based on the PyTorch 1.13.1 deep learning framework and YOLOv8 framework, with the following hardware specifications: NVIDIAGeForce RTX3060 12 GB, Intel (R) CORE (TM) i7-11700, and 32 GB of memory. To ensure fairness in the experiment, no pre-training weights were set. The input image size is 640 × 640, the epoch is 1000, the momentum parameter is set to 0.937, the initial learning rate is 0.01, and the batch size is set to 16.

3.2. Evaluation Indicators

When analyzing the experimental results, this article utilizes precision, recall, F1 score, AP, and mAP as evaluation metrics for model performance. The threshold for the

intersection over union ratio is set at 0.5. Prediction boxes falling below this threshold are considered incorrect predictions, as illustrated in Equations (20)–(24), respectively.

$$Precision = \frac{T_P}{T_P + F_P} \tag{20}$$

$$Recall = \frac{T_P}{T_P + F_N} \tag{21}$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{22}$$

$$AP = \int_0^1 Precision(Recall) dRecall \tag{23}$$

$$mAP = \frac{\sum_{i=1}^{C} AP(i)}{C} \tag{24}$$

In the formula, T_P represents the number of samples correctly identified as belonging to the insect pest image category, F_P represents the number of samples incorrectly identified as belonging to other categories in the insect pest image category, F_N represents the number of samples in the current category of tea insect pest images incorrectly identified by the model as belonging to other categories, and C represents the number of insect pest categories.

As shown in Figure 14, the improved YOLOv8 network has accuracy, recall, and balance scores of 96.32%, 97.95%, and 97.13%, respectively, which are 2.49%, 2.73%, and 2.61% higher than those of the original YOLOv8 network. The results show that the improved YOLOv8 network has made significant improvements in terms of accuracy, recall, and balance score, with the most significant improvement in recall.

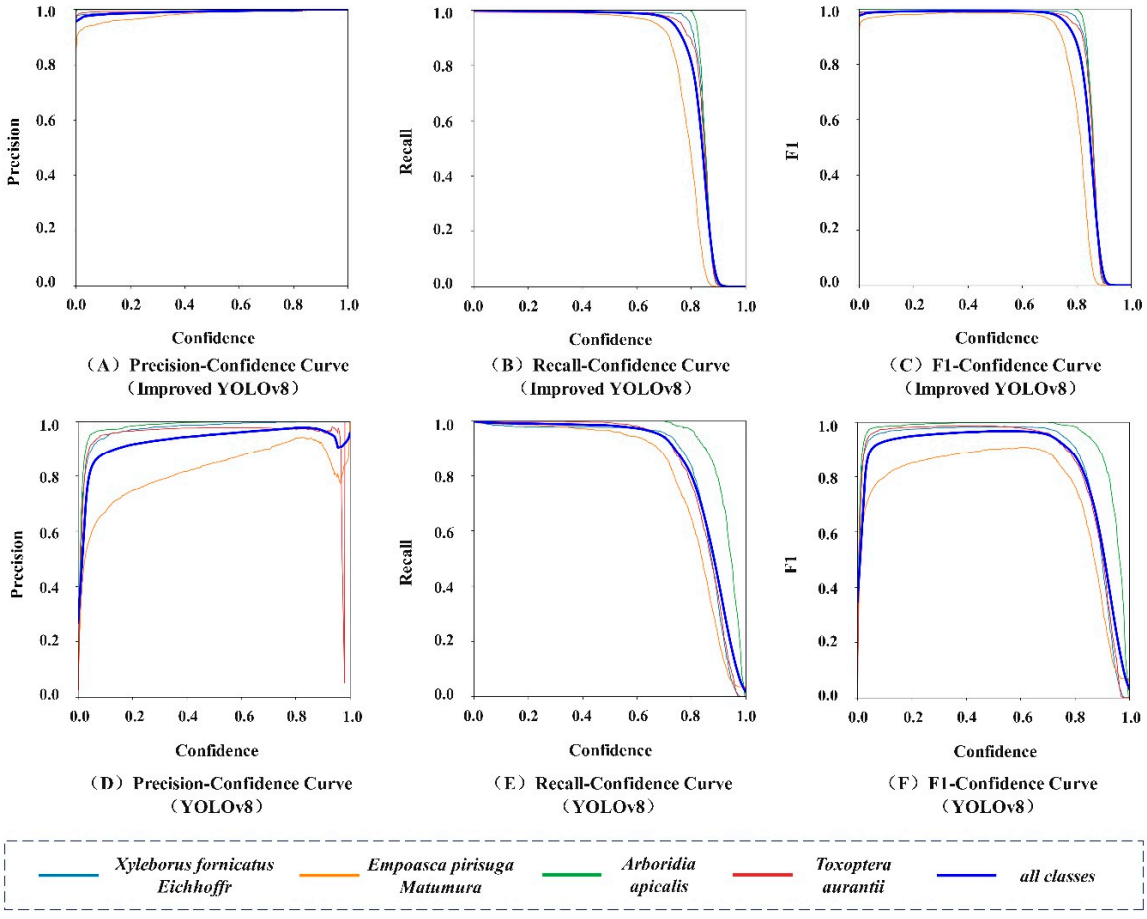

Figure 14. Precision, recall, F1 confidence curve.

3.3. Performance Analysis of Tea Pest Detection Model Based on Improved YOLOv8

To verify the performance improvement of YOLOv8 for the dataset we built, we compared it to the original YOLOv8 model, and the training evaluation index values and loss value change curve shown in Figure 15 were obtained. From the analysis of the changes in loss values, it can be seen that the detection accuracy of each category in the improved YOLOv8 architecture has significantly improved. Overall, the improved YOLOv8 network has a better detection performance compared to that of YOLOv8.

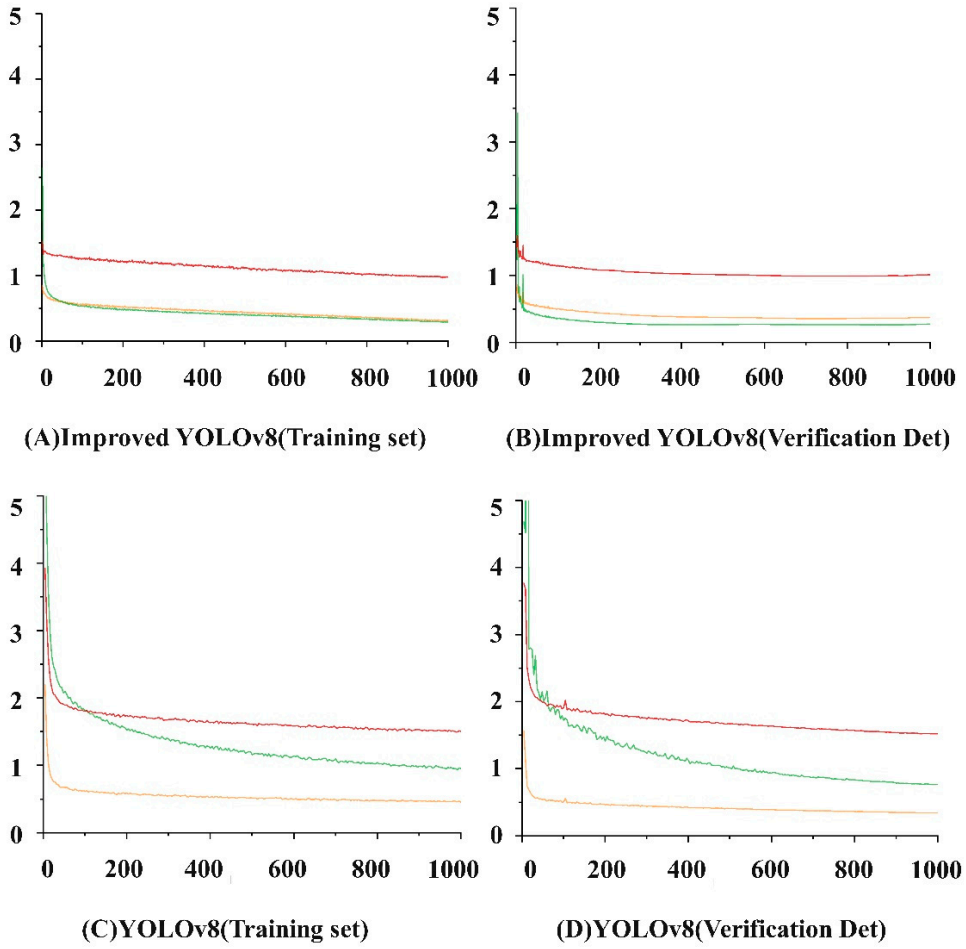

Figure 15. Variation curve of loss function. **Note: Yellow: bounding box loss; Green: classification loss; Red: feature point loss**.

In improving upon YOLOv8, the gradient descent rate of the loss function is very fast in the initial stage of model training. However, when the training reaches the 100th round, the rate of decline in the loss function significantly slows down, and the oscillation of the curve becomes more pronounced, which is 100 rounds earlier than in the original YOLOv8 network. As the training progressed, after 300 rounds, the curve gradually stabilized, 400 rounds earlier than in the original YOLOv8 network. The loss function began to converge, and the bounding box loss, classification loss, and feature point loss stabilized below 2%, 1.5%, and 1%, respectively. By comparing the loss function change curves of the original YOLOv8 and the improved YOLOv8 networks, it can be clearly seen that the improved YOLOv8 network in this study has a significant decrease in bounding box loss, classification loss, and feature point loss, and the feature point loss decreased the most significantly, with a decrease of over 40% for both the training and testing sets.

Figure 16 shows a comparison chart of the tests at different scales, with (A), (B), (C), and (D) showing the original pest map, YOLOv8 heat map, improved YOLOv8 heat map, and actual detection results. The experimental results show that the proposed model has

significant advantages in the detection of small single-pest targets and multiple-pest targets as well as in conditions of low light intensity. The specific model detection results are shown in Table 1.

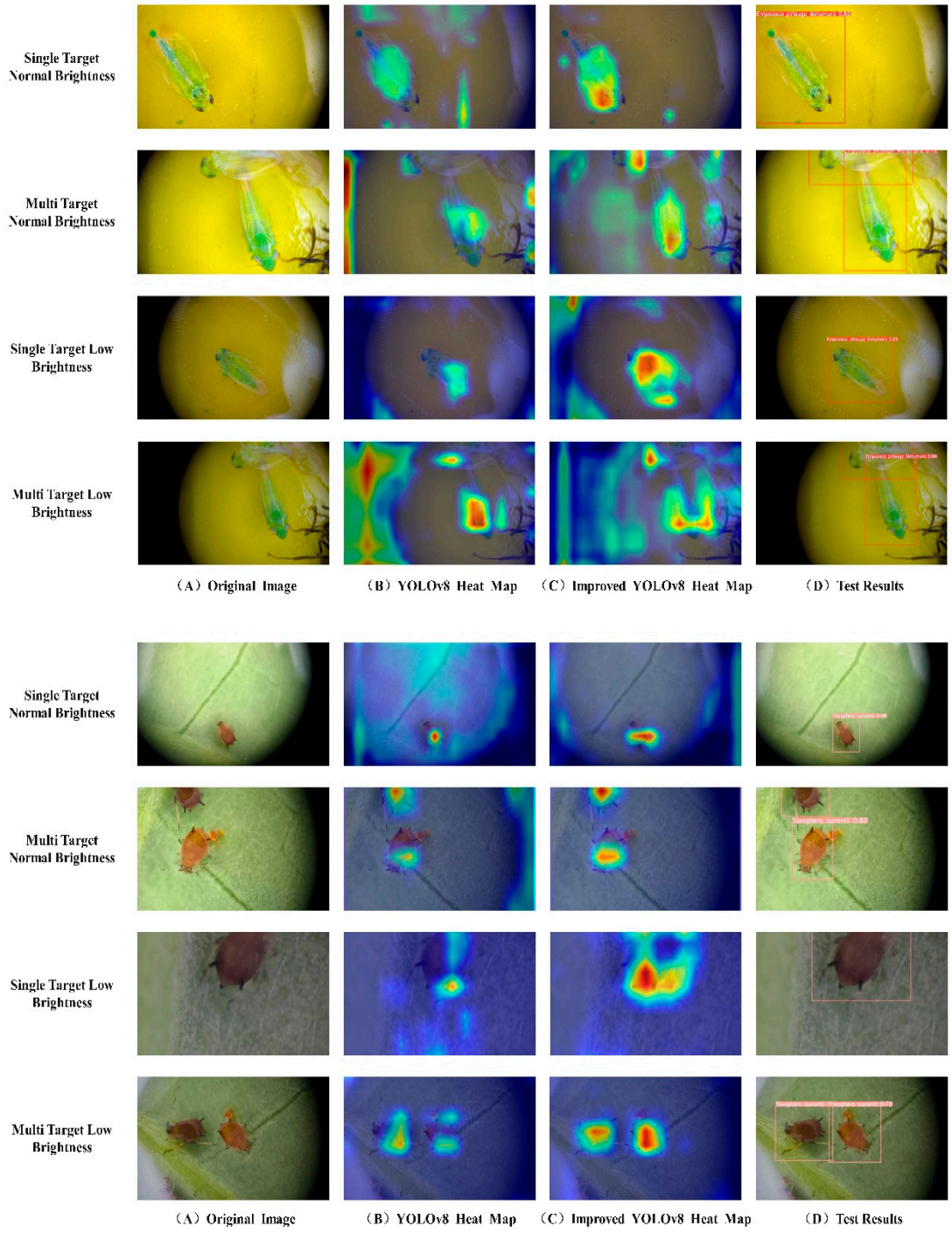

Figure 16. *Cont.*

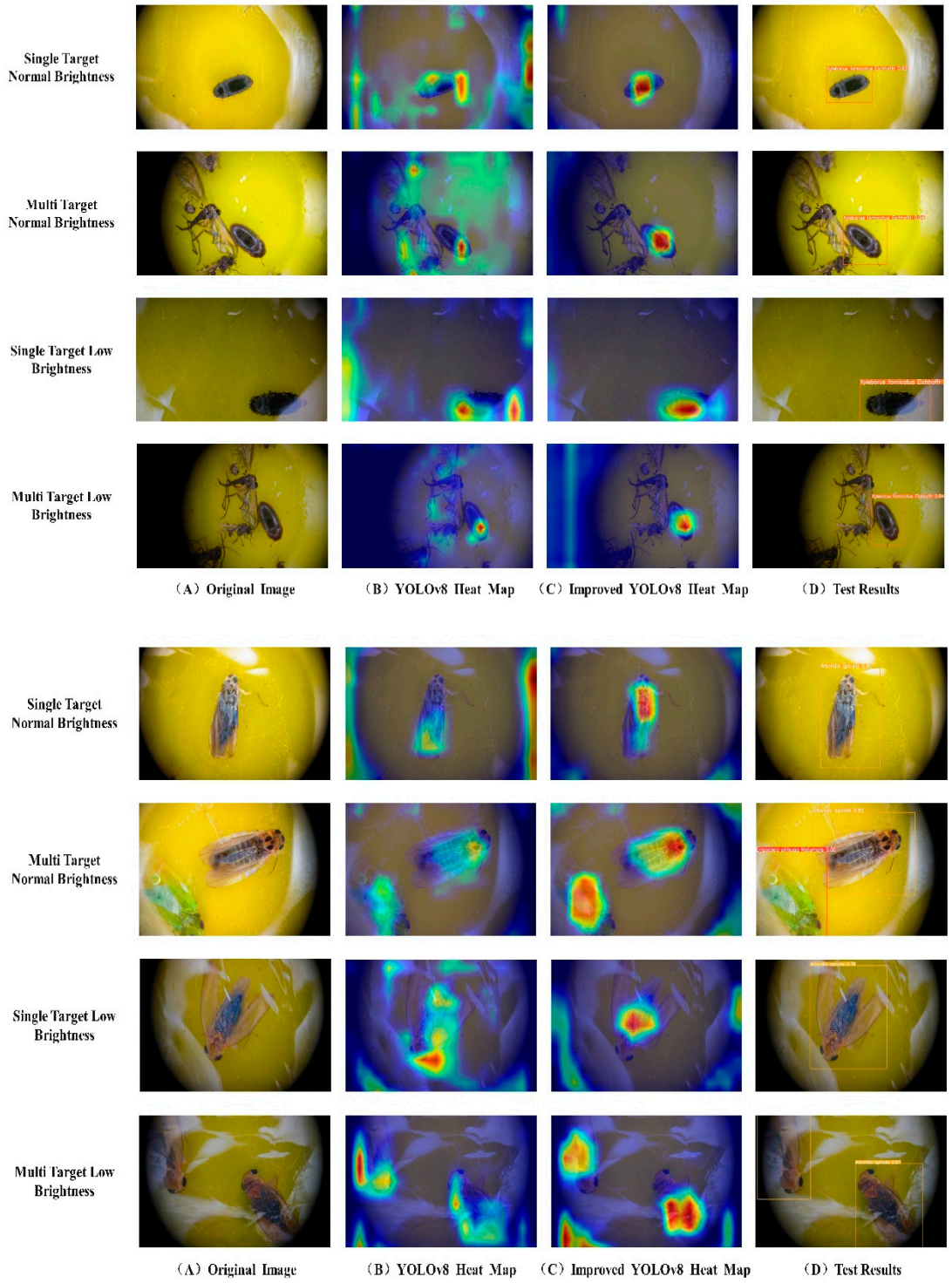

Figure 16. Comparison of the detection effects of four different tea pest datasets.

Table 1. Comparison of model effect before and after improving the dataset.

Model	P/%	R/%	F1/%	AP1	AP2	AP3	AP4	mAP@0.5/%
YOLOv8	93.83	95.22	94.52	95.73	94.66	96.35	95.14	95.46
Improved YOLOv8	96.32	97.95	97.13	98.54	97.18	98.71	98.25	98.17

Note: AP1: AP (*Xyleborus fornicatus Eichhoffr*); AP2: AP (*Empoasca pirisuga Matumura*); AP3: AP (*Arboridia apicalis*); AP4: AP (*Toxoptera awranrii*).

3.4. Comparative Experiments on Detection of Different Models

To verify the superiority of the algorithm in small-object detection, the same dataset and experimental conditions were used to test and compare the improved YOLOv8 network with current mainstream object detection algorithms, including Faster R-CNN [58], SSD [59], and YOLO series models, such as YOLOv5 [60], YOLOv7 [61], and YOLOv8 [62,63].

From the results in Table 2, it can be seen that for the scenario of the small-target detection of tea pests, Faster R-CNN and SSD have poor detection performances, barely reaching about 80%. In terms of indicators, the improved YOLOv8 network has an mAP@50 score of 98.17%, with an average accuracy that is 17.04, 11.23, 5.78, 3.75, and 2.71 percentage points higher than those of Faster R-CNN, SSD, YOLOv5, YOLOv7, and YOLOv8, respectively. Its performance is better than the other object detection models. All of the data in the table are the average values. In terms of accuracy and recall, the improved YOLOv8 model can achieve scores of 96.32% and 97.95%, which are higher than those of the original YOLOv8 model. Additionally, the FPS of the improved algorithm in this paper is 95. Taking into account various indicators, the improved model in this paper is more suitable for small-object detection.

Table 2. Comparison of detection performance of different models.

Model	P/%	R/%	F1/%	mAP@0.5/%	FPS/s
Faster R-CNN	79.11	76.14	77.60	81.13	39
SSD	81.08	84.64	82.82	86.94	58
YOLOv5	86.32	86.99	86.65	92.39	67
YOLOv7	87.91	87.60	87.75	94.42	74
YOLOv8	93.83	95.22	94.52	95.46	76
Improved YOLOv8	96.32	97.95	97.13	98.17	95

3.5. Ablation Experiment

To verify the impact of the improved method on the YOLOv8 model, ablation experiments were conducted on the pest dataset using the aforementioned improved modules of GELAN, MS block, BRA, and inner IOU to demonstrate the effectiveness and necessity of using the improved method. "$\surd$" indicates the module was added to the model, and "-" indicates it was not added.

When each module was applied to the YOLOv8 model separately, ablation experiments were conducted to determine its detection accuracy and detection speed, as shown in Table 3. The results showed that each improvement improved its detection performance to a certain extent. When using the YOLOv8 model, its mAP@0.5 score is 95.46 and FPS scores is 76. After adding the GELAN design, the detection accuracy is not significantly affected, but the detection frame rate per second is improved. Combining the GELAN design with MS block can still retain its advantage of accelerating the detection speed. After adding the bi-level routing attention mechanism, more flexible computation allocation and feature perception were achieved, resulting in a slight improvement in its detection accuracy by 1.12 percentage points. After adding the MS block, BRA, and inner IOU loss functions, compared to that of the original YOLOv8 model, the accuracy increased by 1.96 percentage points, the recall increased by 1.46 percentage points, and the average accuracy increased by 2.47 percentage points. When adding the four modules simultaneously, its mAP@0.5 score is 98.17%, FPS score is 95, and detection speed is relatively improved by 25%, which is more in line with the hardware requirements for detection and leads to a better real-time performance. Ultimately, its mAP@0.5 score improved by 2.71%, and its overall detection indicators were effectively improved. Overall, these four improvements are useful in balancing the detection speed and accuracy, meeting the requirements of being lightweight and having a high level of precision.

Table 3. Experimental results of different improvement methods.

Model	GELAN	MS	BRA	Inner IOU	P/%	R/%	mAP@0.5/%	FPS/s
YOLOv8	-	-	-	-	93.83	95.22	95.46	76
+G	√	-	-	-	91.51	91.47	95.97	85
+M	-	√	-	-	93.81	90.61	96.25	72
+B	-	-	√	-	92.26	93.08	96.58	95
+G +M	√	√	-	-	93.70	96.52	97.75	99
+G +B	√	-	√	-	95.12	94.61	97.24	82
+M +B	-	√	√	-	95.69	95.92	97.78	74
+M +B +I	-	√	√	√	95.79	96.68	97.93	97
+G +M +B +I	√	√	√	√	96.32	97.95	98.17	95

Note: √: uses the algorithm; -: does not use the algorithm.

4. Conclusions

This paper presents an enhanced small-target detection algorithm for YOLOv8, focusing on detecting small tea pest targets at a micro level. The algorithm aims to address issues such as a high density of small targets, significant positioning errors, false detections, and missed detections. The proposed algorithm shows promising advancements in the field of target detection.

1. In response to the problem of a large proportion of background information in images, the SAHI-assisted inference algorithm is applied to the detection network, which increases the detection effect of small targets in local areas through slicing. This provides a novel method for small-target image analysis and meets the demand for high-resolution images under normal shooting.
2. The methods for improving the model's small-object detection capability include designing GELAN and MS as well as introducing a bi-level routing attention mechanism and loss function. GELAN is designed as the backbone network, utilizing PGI to solve information bottleneck problems, ensuring that the feedforward level of the main branch preserves important features while keeping the model lightweight. The neck layer introduces a multi-scale building block to enhance the real-time object detector's ability to extract multi-scale features and improve the inference speed. The BiFormer dual-layer routing attention mechanism and C2f module can guide the network to focus on receptive field information at different scales and for different key pest characteristics. At the same time, adopting inner-MPDIoU instead of the CIOU calculation method accelerates the boundary box regression process and promotes an improvement of the model's generalization ability.
3. We constructed a dedicated tea pest dataset and conducted practical tests on our model and other mainstream models for four scenarios: a single target under normal lighting, multiple targets under normal lighting, a single target under low lighting, and multiple targets under low lighting. The experimental results show that the detection performance of our model is good in these four scenarios, with an mAP@0.5 score that reaches 98.17%, which is 17.04%, 11.23%, 5.78%, 3.75%, and 2.71% higher than those of Faster R-CNN, SSD, YOLOv5, YOLOv7, and YOLOv8, respectively.

The improved version based on the YOLOv8 model performs well in handling pest detection tasks for densely distributed natural scenes with a complex scale, accurately and efficiently extracting and applying pest image features. Therefore, in the field of pest detection, the YOLOv8 model based on Transformer has enormous research potential and important significance.

The primary focus of our future work is to implement the enhanced YOLOv8 algorithm on embedded devices. This will optimize the application of object detection algorithms in pest detection projects within smart agriculture, enhancing accuracy and efficiency in more intricate scenarios, and thereby providing robust support for the sustainable development of the agricultural industry.

Author Contributions: Conceptualization, visualization, writing—original draft preparation, R.Y.; methodology, R.Y. and Y.Q.; software, R.Y. and J.S.; formal analysis, R.Y.; investigation, R.Y., Q.G. and T.L.; conceptualization, writing—review and editing, funding acquisition, T.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Development and Demonstration Center of Yunnan Provincial Major Science and Technology Special Program—Integrated Research on Key Technologies of Smart Agriculture (202302AE090020); the Major Science and Technology Special Program of Yunnan Province—Construction of Crop Growth Model and Intelligent Control of Field (202202AE09002103); the Yunnan Provincial Basic Research Program—Research on Information Security Risk Analysis and Processing Methods for Smart Agriculture (202201AT070981); and the Yunnan Provincial Science and Technology Talent and Platform Program—Yunnan Provincial Key Laboratory of Crop Production and Smart Agriculture (202105AG070007).

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding authors.

Acknowledgments: We would like to thank the editors and anonymous reviewers for their valuable comments and suggestions.

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

References

1. Fu, C.; Yan, W.; Yuan, Y.; Yang, J.; Liu, S.; Wan, S.; Dong, Y. The current situation, problems and countermeasures of the cultivation of tea geographical indication products cultivation in Yunnan Province. *Qual. Saf. Agro-Prod.* **2023**, *3*, 89–93. [CrossRef]
2. Drew, L. The growth of tea. *Nature* **2019**, *566*, S2–S4. [CrossRef]
3. Singh, V.; Misra, A. Detection of plant leaf diseases using image segmentation and soft computing techniques. *Inf. Process. Agric.* **2017**, *4*, 41–49. [CrossRef]
4. Hinton, G.E.; Salakhutdinov, R.R. Reducing the dimensionality of data with neural networks. *Science* **2006**, *313*, 504–507. [CrossRef]
5. Bengio, Y.; Lecun, Y.; Hinton, G. Deep Learning for AI. *Commun. ACM* **2021**, *64*, 58–65. [CrossRef]
6. Dargan, S.; Kumar, M.; Ayyagari, M.R.; Kumar, G. A survey of deep learning and its applications: A new paradigm to machine learning. *Arch. Comput. Methods Eng.* **2020**, *27*, 1071–1092. [CrossRef]
7. Too, E.C.; Li, Y.; Njuki, S.; Liu, Y. A comparative study of fine-tuning deep learning models for plant disease identification. *Comput. Electron. Agric.* **2019**, *161*, 272–279. [CrossRef]
8. Shaopeng, J.; Hongju, G.; Xiao, H. Research progress on image recognition technology of crop pests and diseases on deep learning. *Trans. Chin. Soc. Agric. Mach.* **2019**, *50*, 313–317.
9. Wang, C.; Han, Q.; Li, C.; Li, J.; Kong, D.; Wang, F.; Zou, X. Assisting the Planning of Harvesting Plans for Large Strawberry Fields through Image-Processing Method Based on Deep Learning. *Agriculture* **2024**, *14*, 560. [CrossRef]
10. Zhang, G.; Tian, Y.; Yin, W.; Zheng, C. An Apple Detection and Localization Method for Automated Harvesting under Adverse Light Conditions. *Agriculture* **2024**, *14*, 485. [CrossRef]
11. Feng, L.; Wu, B.; Zhu, S.; Wang, J.; Su, Z.; Liu, F.; He, Y.; Zhang, C. Investigation on data fusion of multisource spectral data for rice leaf diseases identification using machine learning methods. *Front. Plant Sci.* **2020**, *11*, 577063. [CrossRef]
12. Conrad, A.O.; Li, W.; Lee, D.-Y.; Wang, G.-L.; Rodriguez-Saona, L.; Bonello, P. Machine learning-based presymptomatic detection of rice sheath blight using spectral profiles. *Plant Phenomics* **2020**, *2020*, 1–10. [CrossRef]
13. Ganatra, N.; Patel, A. A multiclass plant leaf disease detection using image processing and machine learning techniques. *Int. J. Emerg. Technol.* **2020**, *11*, 1082–1086.
14. Ibañez, S.C.; Monterola, C.P. A Global Forecasting Approach to Large-Scale Crop Production Prediction with Time Series Transformers. *Agriculture* **2023**, *13*, 1855. [CrossRef]
15. Jing, J.; Zhai, M.; Dou, S.; Wang, L.; Lou, B.; Yan, J.; Yuan, S. Optimizing the YOLOv7-Tiny Model with Multiple Strategies for Citrus Fruit Yield Estimation in Complex Scenarios. *Agriculture* **2024**, *14*, 303. [CrossRef]
16. Kumar, C.; Mubvumba, P.; Huang, Y.; Dhillon, J.; Reddy, K. Multi-stage corn yield prediction using high-resolution UAV multispectral data and machine learning models. *Agronomy* **2023**, *13*, 1277. [CrossRef]
17. Chen, J.; Hu, X.; Lu, J.; Chen, Y.; Huang, X. Efficient and Lightweight Automatic Wheat Counting Method with Observation-Centric SORT for Real-Time Unmanned Aerial Vehicle Surveillance. *Agriculture* **2023**, *13*, 2110. [CrossRef]
18. Shah, S.A.; Lakho, G.M.; Keerio, H.A.; Sattar, M.N.; Hussain, G.; Mehdi, M.; Vistro, R.B.; Mahmoud, E.A.; Elansary, H.O. Application of drone surveillance for advance agriculture monitoring by android application using convolution neural network. *Agronomy* **2023**, *13*, 1764. [CrossRef]
19. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: To-wards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef]

20. Kundur, N.C.; Mallikarjuna, P.B. Insect Pest Image Detection and Classification using Deep Learning. *Int. J. Adv. Comput. Sci. Appl. IJACSA* **2022**, *13*, 411–421. [CrossRef]
21. Deng, Z.; Wang, P.; Song, X.; Wang, C.; Chen, J.; Wu, L. Research on Granary Pest Detection Based on SSD. *J. Comput. Eng. Appl.* **2020**, *56*, 214–218.
22. Faisal, M.S.A.B. A pest monitoring system for agriculture using deep learning. *Res. Prog. Mech. Manuf. Eng.* **2021**, *2*, 1023–1034.
23. Liu, C.; Chen, H.; Zeng, X.; Xiang, T.; Kou, X. Farmland Pest Detection Based on YOLO-V5l and ResNet50. *Artif. Intell. Robot. Res.* **2022**, *11*, 236.
24. Hong, J.; Xin, H.; Lu, H.; Liu, R.; Chen, L.; Chen, Z. Tobacco insect recognition in cigarette factory using YOLOV3 model. *Tob. Sci. Technol.* **2020**, *53*, 77.
25. Liu, J.; Wang, X. Tomato Diseases and Pests Detection Based on Improved Yolo V3 Convolutional Neural Network. *Front. Plant Sci.* **2020**, *11*, 521544. [CrossRef]
26. Lin, W.; Zhang, J.; He, N. Real-time detection method of dendrolimus superans-infested larix gmelinii trees based on improved YOLO v4. *Trans. Chin. Soc. Agric. Mach.* **2023**, *54*, 304–312+393.
27. Li, W.; Zhang, W.; Zhou, W.; Han, T.; Wang, P.; Liu, H.; Xiong, M.; Sun, Y. Research and Application of Lightweight Yolov7-TSA Network in Tea Disease Detection and Identification. *J. Henan Agric. Sci.* **2023**, *52*, 162–169.
28. Fuentes, A.; Yoon, S.; Kim, S.C.; Park, D.S. A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. *Sensors* **2017**, *17*, 2022. [CrossRef]
29. Dai, M.; Dorjoy, M.H.; Miao, H.; Zhang, S. A new pest detection method based on improved YOLOv5m. *Insects* **2023**, *14*, 54. [CrossRef]
30. Alam Soeb, J.; Jubayer, F.; Tarin, T.A.; Al Mamun, M.R.; Ruhad, F.M.; Parven, A.; Mubarak, N.M.; Karri, S.L.; Meftaul, I.M. Tea leaf disease detection and identification based on YOLOv7 (YOLO-T). *Sci. Rep.* **2023**, *13*, 6078. [CrossRef]
31. Deng, J.; Yang, C.; Huang, K.; Lei, L.; Ye, J.; Zeng, W.; Zhang, J.; Lan, Y.; Zhang, Y. Deep-Learning-Based Rice Disease and Insect Pest Detection on a Mobile Phone. *Agronomy* **2023**, *13*, 2139. [CrossRef]
32. Guo, C.; Fan, B.; Zhang, Q.; Xiang, S.; Pan, C. Augfpn: Improving multi-scale feature learning for object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 12595–12604.
33. Bai, Y.; Zhang, Y.; Ding, M.; Ghanem, B. Sod-mtgan: Small object detection via multi-task generative adversarial network. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 206–221.
34. Qiao, S.; Chen, L.C.; Yuille, A. Detectors: Detecting objects with recursive feature pyramid and switchable atrous convolution. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 10213–10224.
35. Lim, J.S.; Astrid, M.; Yoon, H.J.; Lee, S.I. Small object detection using context and attention. In Proceedings of the 2021 international Conference on Artificial Intelligence in Information and Communication (ICAIIC), Jeju Island, Republic of Korea, 13–16 April 2021; pp. 181–186.
36. Liu, Y.; Yang, F.; Hu, P. Small-object detection in UAV-captured images via multi-branch parallel feature pyramid networks. *IEEE Access* **2020**, *8*, 145740–145750. [CrossRef]
37. Chen, Y.; Zheng, W.; Zhao, Y.; Song, T.H.; Shin, H. DW-yolo: An efficient object detector for drones and self-driving vehicles. *Arab. J. Sci. Eng.* **2023**, *48*, 1427–1436. [CrossRef]
38. Yu, J.; Jiang, Y.; Wang, Z.; Cao, Z.; Huang, T. UnitBox: An advanced object detection network. In Proceedings of the 24th ACM International Conference on Multimedia, Amsterdam, The Netherlands, 15–19 October 2016; pp. 516–520.
39. Rezatofighi, H.; Tsoi, N.; Gwak, J.; Sadeghian, A.; Reid, I.; Savarese, S. Generalized intersection over union: A metric and a loss for bounding box regression. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 658–666.
40. Sarasaen, C.; Chatterjee, S.; Breitkopf, M.; Rose, G.; Nürnberger, A.; Speck, O. Finetuning deep learning model parameters for improved superresolution of dynamic MRI with prior-knowledge. *Artif. Intell. Med.* **2021**, *121*, 102196. [CrossRef]
41. Tishby, N.; Zaslavsky, N. Deep learning and the information bottleneck principle. In Proceedings of the IEEE Information Theory Workshop (ITW), Jerusalem, Israel, 26 April–1 May 2015; pp. 1–5.
42. Cai, Y.; Zhou, Y.; Han, Q.; Sun, J.; Kong, X.; Li, J.; Zhang, X. Reversible column networks. In Proceedings of the International Conference on Learning Representations (ICLR), Kigali, Rwanda, 1–5 May 2023.
43. Wang, L.; Lee, C.-Y.; Tu, Z.; Lazebnik, S. Training deeper convolutional networks with deep supervision. *arXiv* **2015**, arXiv:1505.02496.
44. Wang, C.-Y.; Liao, H.-Y.M.; Yeh, I.-H. Designing network design strategies through gradient path analysis. *J. Inf. Sci. Eng. JISE* **2023**, *39*, 975–995.
45. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Curran Associates Inc.: Long Beach, CA, USA, 2017; pp. 6000–6010.
46. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16×16 words: Transformers for image recognition at scale. In Proceedings of the 9th International Conference on Learning Representations ICLR, Addis Ababa, Ethiopia, 30 April 2020.

47. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with Transformers. In Proceedings of the 16th European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; Springer: Glasgow, UK, 2020; pp. 213–229.
48. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
49. Wang, C.-Y.; Liao, H.-Y.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; pp. 390–391.
50. Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W.; et al. Yolov6: A single-stage object detection framework for industrial applications. *arXiv* **2022**, arXiv:2209.02976.
51. Xu, S.; Wang, X.; Lv, W.; Chang, Q.; Cui, C.; Deng, K.; Wang, G.; Dang, Q.; Wei, S.; Du, Y.; et al. Pp-yoloe: An evolved version of yolo. *arXiv* **2022**, arXiv:2203.16250.
52. Wang, C.-Y.; Bochkovskiy, A.; Liao, H.-Y.M. Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023.
53. Lyu, C.; Zhang, W.; Huang, H.; Zhou, Y.; Wang, Y.; Liu, Y.; Zhang, S.; Chen, K. Rtmdet: An empirical study of designing real-time object detectors. *arXiv* **2022**, arXiv:2212.07784.
54. Gao, S.-H.; Cheng, M.-M.; Zhao, K.; Zhang, X.-Y.; Yang, M.-H.; Torr, P. Res2net: A new multi-scale backbone architecture. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 652–662. [CrossRef]
55. Zheng, Z.; Wang, P.; Liu, W.; Li, J.; Ye, R.; Ren, D. Distance-IoU loss: Faster and better learning for bounding box regression. In Proceedings of the 34th AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; AAAI: New York, NY, USA, 2020; p. 12993.
56. He, J.; Erfani, S.; Ma, X.; Bailey, J.; Chi, Y.; Hua, X.S. Alpha-IoU: A family of power intersection over union losses for bounding box regression. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 20230–20242.
57. Zhang, H.; Xu, C.; Zhang, S. Inner-IoU: More Effective Intersection over Union Loss with Auxiliary Bounding Box. *arXiv* **2023**, arXiv:2311.02877.
58. Siliang, M.; Yong, X. MPDIoU: A Loss for Efficient and Accurate Bounding Box Regression. *arXiv* **2023**, arXiv:2307.07662.
59. Bari, B.S.; Islam, N.; Rashid, M.; Hasan, J.; Razman, M.A.M.; Musa, R.M.; Ab Nasir, A.F.; Majeed, A.P.A. A real-time approach of diagnosing rice leaf disease using deep learning-based faster R-CNN framework. *PeerJ Comput. Sci.* **2021**, *7*, e432. [CrossRef]
60. Lyu, Z.; Jin, H.; Zhen, T.; Sun, F.; Xu, H. Small Object Recognition Algorithm of Grain Pests Based on SSD Feature Fusion. *IEEE Access* **2021**, *9*, 43202–43213. [CrossRef]
61. Önler, E. Real time pest detection using YOLOv5. *Int. J. Agric. Nat. Sci.* **2021**, *14*, 232–246.
62. Jia, L.; Wang, T.; Chen, Y.; Zang, Y.; Li, X.; Shi, H.; Gao, L. MobileNet-CA-YOLO: An improved YOLOv7 based on the MobileNetV3 and attention mechanism for Rice pests and diseases detection. *Agriculture* **2023**, *13*, 1285. [CrossRef]
63. Zhang, L.; Ding, G.; Li, C.; Li, D. DCF-Yolov8: An Improved Algorithm for Aggregating Low-Level Features to Detect Agricultural Pests and Diseases. *Agronomy* **2023**, *13*, 2012. [CrossRef]