agronomy

MDPI

Article

YOLOv8-GDCI: Research on the Phytophthora Blight Detection Method of Different Parts of Chili Based on Improved YOLOv8 Model

Yulong Duan [1], Weiyu Han [1], Peng Guo [1] and Xinhua Wei [1,2,*]

[1] School of Computer Science and Technology, Xinjiang University, Urumqi 830000, China; 107552203990@stu.xju.edu.cn (Y.D.); 107552203999@stu.xju.edu.cn (W.H.); guopeng@stu.xju.edu.cn (P.G.)
[2] College of Agricultural Engineering, Jiangsu University, Zhenjiang 212013, China
* Correspondence: wei_xh@126.com

Abstract: Smart farms are crucial in modern agriculture, but current object detection algorithms cannot detect chili Phytophthora blight accurately. To solve this, we introduced the YOLOv8-GDCI model, which can detect the disease on leaves, fruits, and stem bifurcations. The model uses RepGFPN for feature fusion, Dysample upsampling for accuracy, CA attention for feature capture, and Inner-MPDIoU loss for small object detection. In addition, we also created a dataset of chili Phytophthora blight on leaves, fruits, and stem bifurcations, and conducted comparative experiments. The results manifest that the YOLOv8-GDCI model demonstrates outstanding performance across a gamut of comprehensive indicators. In comparison with the YOLOv8n model, the YOLOv8-GDCI model demonstrates an improvement of 0.9% in precision, an increase of 1.8% in recall, and a remarkable enhancement of 1.7% in average precision. Although the FPS decreases slightly, it still exceeds the industry standard for real-time object detection (FPS > 60), thus meeting the requirements for real-time detection.

Keywords: smart farm; chili phytophthora blight; object detection; YOLOv8n

check for updates

Citation: Duan, Y.; Han, W.; Guo, P.; Wei, X. YOLOv8-GDCI: Research on the Phytophthora Blight Detection Method of Different Parts of Chili Based on Improved YOLOv8 Model. *Agronomy* **2024**, *14*, 2734. https://doi.org/10.3390/agronomy14112734

Academic Editor: Chengming Sun

Received: 12 October 2024
Revised: 6 November 2024
Accepted: 13 November 2024
Published: 20 November 2024

1. Introduction

Chilies, as common seasonings and delicious dishes, are deeply loved by people from all over the world. As technology continues to advance at a rapid pace, China has firmly established itself as the preeminent producer and consumer of chilies globally. The cultivation area of chilies in China reached an impressive 21,474 km^2 in 2019 and continues to exhibit a growth trend [1]. Among the provinces, Xinjiang, as the sixth largest chili-producing region in China, holds advantages over other regions due to its extensive land and relatively sparse population distribution, which offers exceptional conditions for the establishment of smart farms. Amidst the swift evolution of information technology, smart agriculture has emerged as a groundbreaking orientation for modern agricultural development, progressively ushering in profound alterations in agricultural production methodologies. Smart agriculture integrates an array of high-end intelligent equipment such as unmanned tillage machines, unmanned seeders, and agricultural robots by deeply fusing information technology and agricultural technology, achieving intelligent and automated management throughout the entire agricultural production process. Farmers can effortlessly accomplish remote monitoring and precise operation of the farm via mobile terminal devices, significantly enhancing both the efficiency and management level of operations like tillage, seeding, fertilization, irrigation, agricultural condition monitoring, and field inspection [2]. In the actual production of smart farms, crop yields are typically influenced by numerous factors, with the issue of pests and diseases being particularly acute. Chili Phytophthora blight, as one of the major diseases affecting chili yields, cannot be disregarded due to its high incidence rate and potential threat of severe yield reduction, posing significant harm to production [3]. Hence, the timely and accurate identification of chili Phytophthora blight is of great significance to farm yields. Traditional manual identification of chili Phytophthora blight is challenging and consumes a considerable amount of time and human resources. With the improvement of computer vision and image processing technology, object detection technology based on deep learning can achieve more precise real-time detection of chili Phytophthora blight and effectively save time and labor, further elevating the efficiency of smart farms.

When deliberating on the application of deep learning approaches for object detection, the domain is typically categorized into two principal methodologies: the YOLO family [4–10], which embodies an efficient one-stage detection paradigm, and Faster R-CNN [11], renowned for its two-stage detection technique. Generally, two-stage detection models exhibit superior accuracy, yet their significantly higher computational time cost poses a challenge for the real-time monitoring of chili Phytophthora blight. However, through continuous improvements, the detection performance of one-stage detection models has gradually become optimal, and the required time has also been significantly reduced, making them more suitable for real-time monitoring of chili Phytophthora blight.

Target detection technology based on the one-stage detection model YOLO has been widely used in crop pest detection. Zhao applied the YOLOv2 model to the task of detecting diseased tomatoes and healthy tomatoes. The highest accuracy of the model could reach up to 0.96, demonstrating that the YOLOv2 model holds great significance in the domain of detecting diseased tomatoes [12]. To tackle the challenges faced in promptly identifying tomato gray leaf spot disease at its onset, Liu created the MobileNetv2-YOLOv3 model. This innovative model replaces the traditional backbone network of YOLOv3 with MobileNetv2 and incorporates the GIoU loss function. Its speed can reach 246 frames per second, with an AP value of 91.32% [13]. To precisely identify and monitor the crucial diseases of rice leaves, such as leaf blight, rice blast, and brown spot, etc., Arun proposed the UAV T-yolo-Rice network model by adding modules like SPP (Spatial Pyramid Pooling), CBAM (Convolutional Block Attention Module), and SCFEM (Specific Context Feature Enhancement Module) on the basis of Tiny YOLOv4. These enhancements improved the model's capability of capturing the lesion characteristics of rice leaves, with an average accuracy of up to 86% [14]. Xie improved on YOLOv5s and proposed the YOLOv5s-BiPCNeXt model. This model utilizes the MobileNeXt backbone to reduce computational complexity and incorporates the ema attention mechanism. It enhances the upsampling algorithm with CARAFE, thereby improving the detection and localization of early small lesions of brown spots and powdery mildew on eggplant leaves [15]. Yue proposed the YOLOv7-GCA model for the detection of four diseases of chili, namely anthracnose, bacterial disease, blossom-end rot, and viral disease, and to cope with the complex field environment. GhostNetv2 was used in place of the original backbone network, while CFnet was employed to take the position of the initial feature fusion module. Eventually, the CBAM attention mechanism was introduced, with the average accuracy being enhanced by 13.4% [16]. To address the challenge of accurately identifying corn leaf spot disease in complex environments, Yang improved upon YOLOv8 by incorporating the Slim-neck module and the GAM attention module. These additions enhanced the model's ability to recognize corn leaf spot disease. The average accuracy rate was improved by 3.56% compared to YOLOv8 [17].

Existing object detection models, such as YOLOv8, despite their achievements in detecting diseases like corn leaf spot and eggplant powdery mildew, encounter difficulties in effectively detecting Phytophthora blight. Phytophthora blight is a fine-grained disease with subtle feature differences. The symptoms of chili Phytophthora blight manifest as water-soaked spots on leaves, irregular patches on fruits, and damp dark green patches on stems. Due to the significant overlap in morphology and color changes of these lesions with the background, and sometimes due to the small size or partial concealment of the lesions, YOLOv8 may miss or misdetect these features. Furthermore, factors such as variations in lighting, overlapping leaves, and the severity of disease progression further interfere with detection, making it even more prone to missed or false detections. Additionally, most studies overlook the effects of pests and diseases on different plant parts, thereby potentially lacking in practical application. In this study, we constructed datasets for various pathogenic sites of chili Phytophthora blight and put forward an enhanced YOLOv8n model named YOLOv8-GDCI based on the pathogenic characteristics of the disease. It inherits the efficiency and accuracy of the YOLO series models and achieves more precise and efficient disease detection by detecting the features of chili Phytophthora blight. The following are the primary advancements that are presented in this paper:

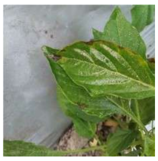
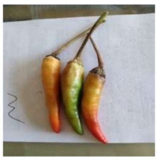
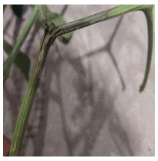
- In this paper, a dataset containing three labels, namely leaf-1, fruit-1, and stem-1, was constructed based on the disease sites, providing data support for the detection of chili Phytophthora blight.
- Using RepGFPN instead of the original Neck network can better integrate the feature maps of different levels without significantly increasing the computational cost, to solve the problem of complex background.
- DySample is used to replace the original upsample algorithm, which can better aggregate context information and introduce less computing overhead.
- To enhance the detection of concealed and compact targets, the CA attention mechanism is integrated into the final stage of the backbone network, which combines both channel and spatial location information.
- Inner-MPDIoU loss replaced CIoU, minimizing corner distances between predicted and true boxes. Furthermore, an auxiliary bounding box has been implemented to enhance the learning capacity for chili disease samples.

2. Materials and Methods

2.1. Data Collection

Chili Phytophthora blight commonly affects leaves, fruits, stem bifurcations, stem bases, and roots. The stem base, being close to the soil surface, has its infected areas easily obscured by coverings such as plastic mulch and soil particles. Additionally, the roots, buried deep in the soil, pose significant challenges for direct observation and detection. In practical scenarios, the stem base and roots may require specific collection efforts by farmers, which does not align with the application scenarios of smart farms. In contrast, other parts of the plant can be collected by agricultural robots or other devices. Therefore, this paper focuses on the visible and easily accessible parts of the chili plant, namely, leaves, fruits, and stem bifurcations. Currently, there are several shortcomings in the datasets available online regarding chili Phytophthora blight: The primary issue lies in the fact that most of the image materials do not align with our specific research needs, as they focus more on the roots and stems of chilies, which are difficult to directly observe in actual production environments. Furthermore, the existing datasets also contain a significant number of images of chili seedlings. The leaves of these seedlings differ greatly in shape, color, and other aspects from the leaves of mature chili plants that we encounter in actual inspections. Additionally, seedlings are often obscured by weeds, soil, or other objects on the ground, which further increases the difficulty and uncertainty of image recognition. Given these circumstances, we have decided to personally embark on constructing a dataset that is more tailored to our research needs. The data were collected in Changji Farm of Xinjiang using vivo X80 (Vivo Company, Dongguan, China) as the acquisition device. Vivo X80 is equipped with a high-resolution camera and advanced optical sensing technology, enabling it to capture detailed close-up images with precision. This makes it highly suitable for capturing the high-quality images required for disease detection in agricultural scenarios. Images of leaves, fruits, and stem forks were captured from various perspectives, including front, side, top-down, and upward views. Additionally, images were taken from multiple distances, including close-up, mid-range, and long-range. Meanwhile, to ensure the image quality, data cleaning was conducted on original images. Using the labelImg 1.8.6 software, the affected leaves, fruits, and stem bifurcations were labeled as “leaf-1”, “fruit-1”, and “stem-1”, respectively, by selecting appropriate bounding boxes to exactly enclose the targets and also annotating any partially obscured targets. A total of 1083 pictures and 4090 labels were obtained, among which the number of leaf labels was the largest (1684), while the number of fruit and stem bifurcation labels was smaller (1275 and 1131, respectively). To present the dataset and the characteristics of chili blight in different parts more intuitively, Table 1 was created.

Table 1. Characteristics of different pathogenic sites of chili Phytophthora blight.

Disease Site	Label	Label Number	Disease Characteristics	Picture
leaf	leaf-1	1684	Round or irregularly shaped “watered-in” spots appear on the leaf infection spots, the leaf color becomes darker, and then the spots begin to expand and become dry and brown.	
fruit	fruit-1	1275	The disease initiates at either the base or the tip, and spreads with irregular water-soaked patches, causing the fruit to turn brown and the pulp to become soft and rotten.	
stem	stem-1	1131	Initially, dark green, moist, and irregularly shaped patches emerge, which subsequently transform into dark brown to black lesions.	

According to common standards, the dataset was partitioned into training, validation, and testing subsets, with a ratio of 7:1.5:1.5, respectively. To enhance the robustness and generalization ability of the model, we carried out Gaussian noise, brightness change, random clipping and random flip operations on the training set. For Gaussian noise, we added noise with a standard deviation of 25 to simulate the subtle disturbances present in real-world environments. Brightness adjustment was set within a ±20% range to account for varying lighting conditions. The random cropping range was set between 70% and 90% of the original image size, ensuring that the target region remained intact within the image. Random flipping included both horizontal and vertical flips to mimic the target from different viewing angles. These augmentation parameters were designed to introduce moderate variations without compromising data authenticity, thereby enhancing the model’s generalization capability. Eventually, the training set included 3870 images, and the number of verification and test sets was 155 and 154, respectively. The specific effect of data augmentation is depicted in Figure 1.

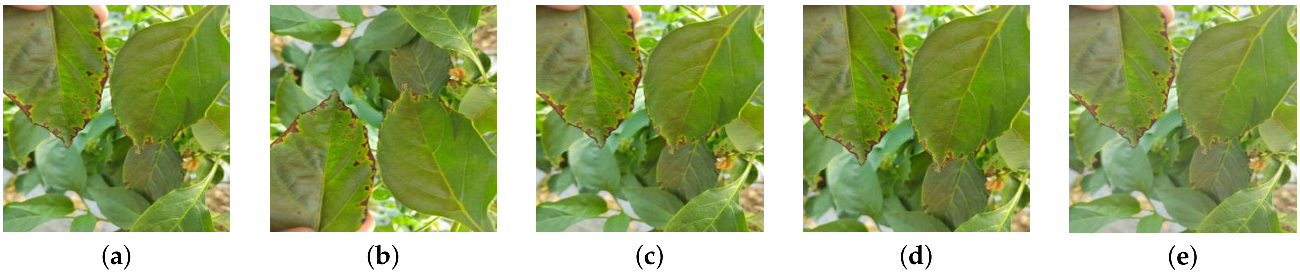
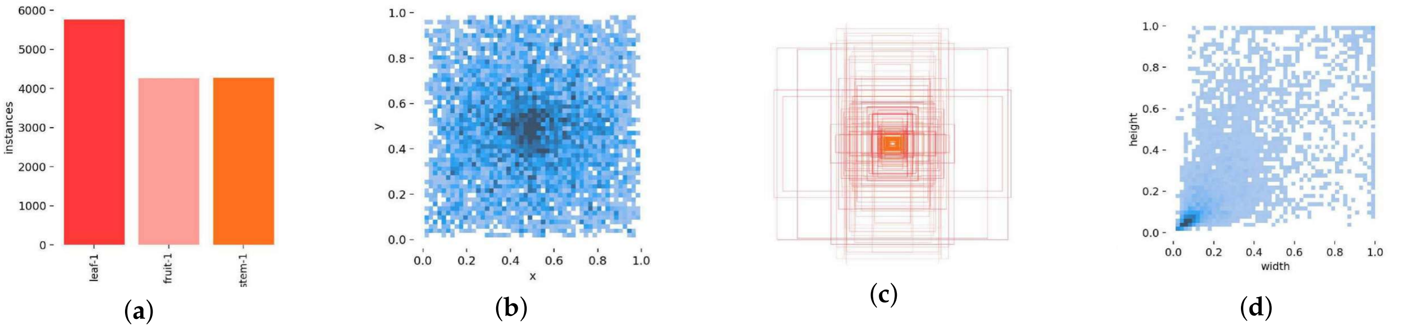


Figure 1. Image data enhancement. (**a**) Original image, (**b**) Random flipping, (**c**) Gaussian noise, (**d**) Random clipping, (**e**) brightness change.

The data distribution presented in Figure 2 reveals a clustering tendency towards the central area of the image, with the majority of samples lying within the 0 to 0.2 range. This implies that the subjects under examination mainly belong to the category of small samples. The impact of such distribution characteristics on the detection model for Phytophthora blight is significant: Due to the high proportion of small samples, the detection model needs to possess the ability to identify small targets, which is generally challenging for standard detection methods because the model may struggle to accurately extract lesion features when processing small-sized bounding boxes. Furthermore, the clustering trend of targets centered in the image implies that during the training process, the model is more likely to optimize the lesion features in the central region of the image, potentially compromising the detection accuracy of the peripheral regions. This trend suggests potential improvements through data augmentation techniques, which could involve increasing samples with targets located at the edges of the image to balance the target distribution and enhance the detection of lesions in those areas.



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

Figure 2. The distribution chart of dataset labels. (**a**) The amount of data in the training set, and how many instances there are for each category. (**b**) The size and number of bounding boxes. (**c**) The position of the center point relative to the entire image. (**d**) The aspect ratio of the target in the image compared to the entire image.

2.2. Methods

2.2.1. YOLOv8 Model

The YOLO series is currently the most prevalently utilized real-time object detection algorithm. One of the models, YOLOv8, stands out as the most suitable choice for striking a balance between precision and efficiency, making it particularly valuable for real-time object detection. The YOLOv8 model is mainly partitioned into four parts: Input, Backbone, Neck, and Head [18]. The input side primarily handles the input image and transforms it into a compatible format for the model to process. The backbone network serves as the primary means of extracting features from the input pictures. Compared with the YOLOv5 model, YOLOv8 uses a C2f module instead of a C3 module to obtain more gradient flow information. The neck network aims to enhance and integrate feature maps, thereby enhancing the detection capability for targets of varying sizes. The output receives the feature map and outputs the object detection result. YOLOv8 adopts the decoupage structure at the output end to process the category and position characteristics, respectively, which improves the accuracy and robustness. Furthermore, YOLOv8 utilizes anchor-free detection, leading to a decrease in the necessity for Anchor frames and consequently enhancing the speed of detection. The YOLOv8 network structure is shown in the Figure 3. YOLOv8 models are divided into five types, namely YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l and YOLOv8x. The size of these models increases successively, leading to progressively longer training times and higher detection accuracy. For practical application in agricultural production, the text selected the fastest YOLOv8n for improvement, so as to better realize the real-time detection of chili Phytophthora blight.

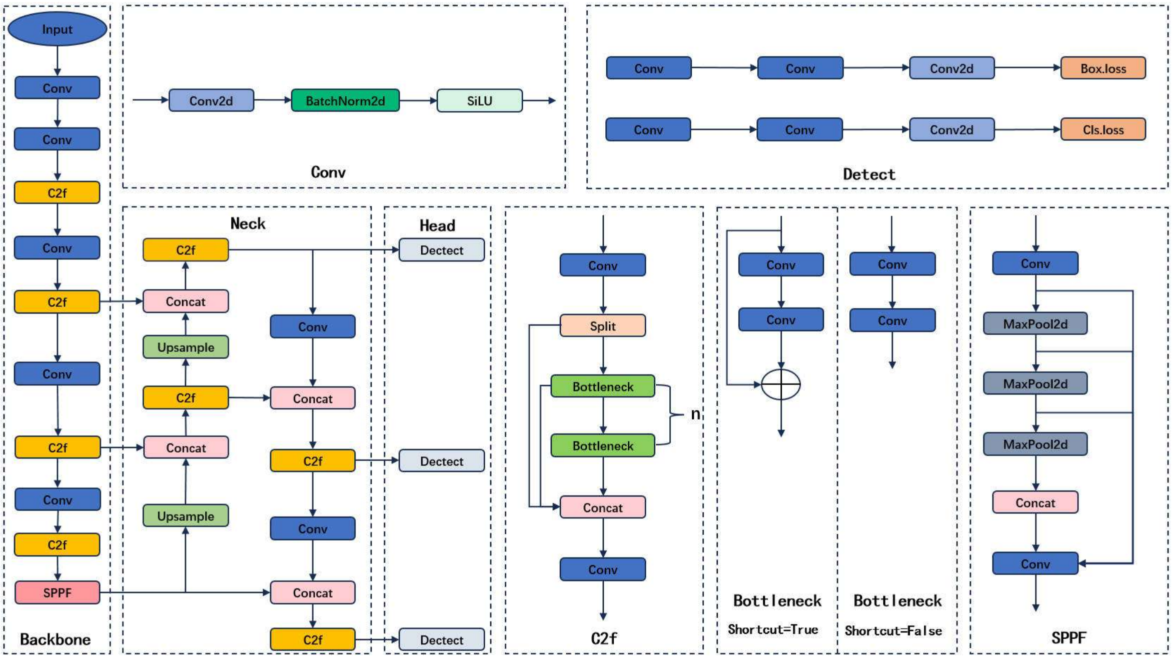


Figure 3. The structure of YOLOv8.

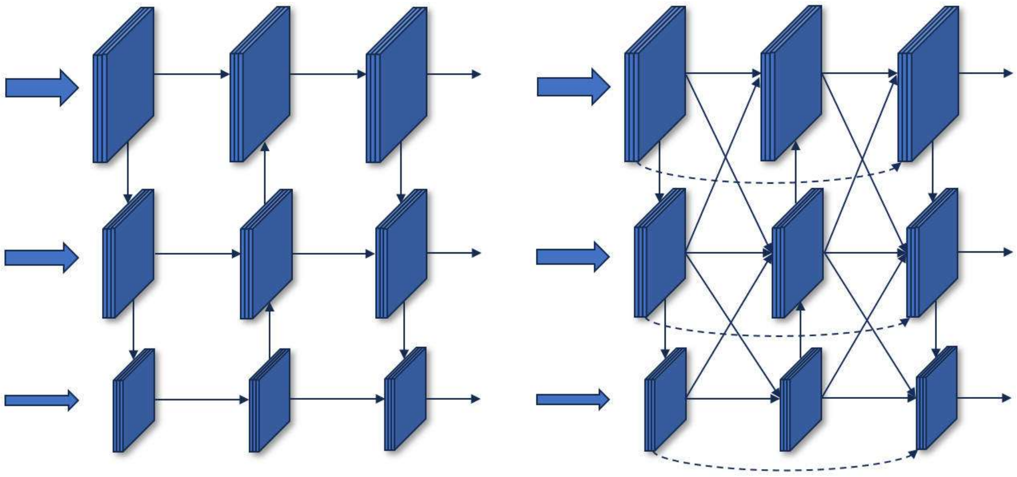
2.2.2. Multi-Scale Feature Fusion Network RepGFPN

The YOLOv8 network continues to adopt the design of YOLOv5 at the Neck end and employs the structure of the feature pyramid PAN [19] with top-down and bottom-up approaches to handle multi-scale features. This network structure only considers two aspects: the same scale at adjacent levels and different scales at the same level. This limits the breadth and depth of information transmission and fusion. The GFPN [20] network architecture overcomes the limitations of PAN by incorporating additional features of varying scales from diagonally above and below, as well as features of similar scales from distinct layers, significantly enhancing the scope and depth of feature fusion. Although GFPN improves the accuracy, it significantly increases the inference time consumption and struggles to comply with the need for instant detection in smart farms for chili Phytophthora blight. The drawback of GFPN lies in that it fails to perform differentiated resource allocation for features of different scales but uniformly adopts the same number of channels for the same scale. This might waste resources on high-resolution feature maps, causing a waste of computing resources. Additionally, GFPN involves a significant amount of upsampling and downsampling operations, leading to an increase in computational complexity. The network structure diagrams of PAN and GFPN are depicted in the Figure 4.

Therefore, this paper adopts the RepGFPN [21] network. RepGfpn basically retains the network structure of GFPN and inherits the $log_2(n) - link$ connections. Specifically, in each level k, the $l^{(th)}$ layer receives the feature maps from at most $log_2 l + 1$ number of preceding layers, and these input layers are exponentially apart from depth i with base 2, as denoted:

$$P_k^l = \text{Conv}\left(\text{Concat}(P_k^0, \ldots, P_k^{l-1})\right) \tag{1}$$

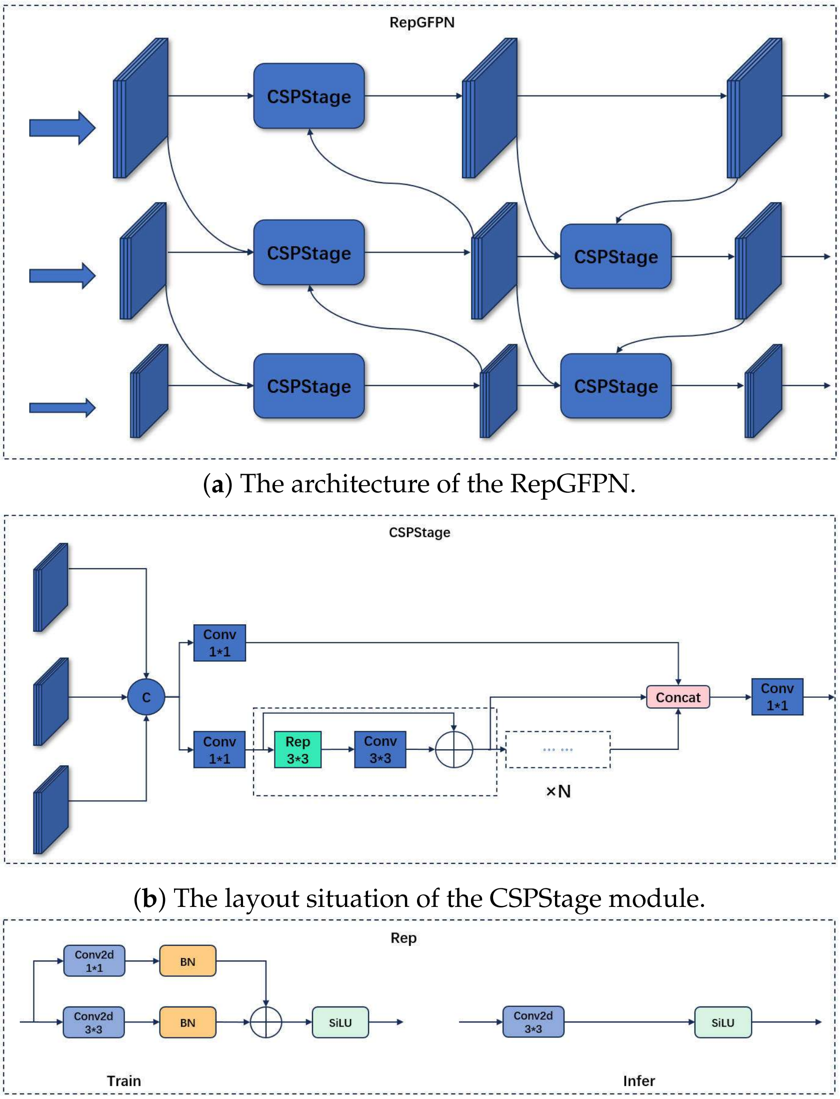
where $l - 2^n \geq 0$, Concat() and Conv() also represent concatenation and 3 × 3 convolution, respectively. The time complexity of the $\log_2 n$-link is only $O(l \cdot \log_2 l)$, rather than $O(l^2)$. This means that the RepGFPN network structure can obtain higher-quality feature maps at a lower cost. This network model uses different numbers of channels for features of different scales, respectively. In practical situations, the lesions caused by chili Phytophthora blight disease vary greatly in size, shape, and distribution. By adjusting the number of channels, RepGFPN can more flexibly extract feature information at different scales, thereby enhancing its ability to recognize lesions of various sizes while ensuring computational efficiency. Furthermore, this network eliminates some of the upsampling connections, avoiding unnecessary computational burdens. For most disease detection scenarios, upsampling operations do not significantly enhance the features of small regions. Reducing upsampling connections allows the model to focus on fine features at the original resolution, improving the detection accuracy of small lesions. In terms of feature fusion, the CSPStage module is tailored for this purpose, and Repconv [22] is employed to substitute the 3 × 3 convolution of GFPN, enhancing the efficacy of feature integration and boosting the model's precision without incurring additional computational burdens. Figure 5 presents the architecture of the RepGFPN network and the layout situation of the CSPStage module.



(**a**) PAN (**b**) GFPN

Figure 4. The structure of PAN (**a**) and GFPN (**b**).

The CSPStage module serves as a crucial component of RepGFPN, primarily accountable for feature fusion. The incoming feature maps have undergone diverse sampling processes to achieve a uniform resolution. Feature maps of a consistent resolution undergo a Contact operation to increase their channel dimensionality. To integrate the information contained within the feature maps, we employ a strategy that integrates multi-branch parallel processing alongside multi-level feature fusion. First, the input feature maps are split into two parallel branches for independent processing. After the first branch undergoes a 1 × 1 convolution for the dimensionality reduction operation, no other operations are performed, and it directly participates in the subsequent concat operation, ensuring the direct retention of the original feature information and providing unmodified original information for the final feature integration. The second branch first applies a Rep3 × 3 convolution layer. During the training process, this layer utilizes the reparameterization technique to obtain diverse feature representations within a multi-branch structure. During the inference stage, it smoothly integrates into an efficient 3 × 3 convolution layer through the application of the same reparameterization technique, guaranteeing model accuracy and boosting inference speed. This multi-branch structure helps the network accelerate the inference process while maintaining high accuracy, enabling it to better distinguish the characteristics of different disease spots, especially in identifying subtle disease spot boundaries and feature differences against complex backgrounds. The specific form of the structure is clearly presented in Figure 5c. Subsequently, a standard 3 × 3 convolution layer is employed to further refine and enhance the feature maps, thereby enhancing their nonlinear expression capacity and local feature capture capability. We perform the addition operation on the feature maps from the second branch, which have been through Rep3 × 3 and 3 × 3 convolution operations, as well as those that have not undergone any convolution operation, to enhance the richness of feature information. This operation is repeated n times to obtain n new feature maps. By utilizing multi-level parallel processing, the unprocessed feature maps from the initial branch undergo a concatenation operation with the n feature maps generated by the subsequent branch. This facilitates the merging of feature maps across different branches and layers, strengthening the model's capacity to gather and understand intricate feature details. This operation enables the comprehensive fusion of cross-level and multi-branch feature maps, making the model more discriminative when identifying complex disease spot edges, colors, and texture features. Finally, a 1 × 1 convolution layer is employed to perform dimensionality reduction and linear transformation on the consolidated feature sets, resulting in the ultimate feature representation for facilitating the network's forward propagation process. This step not only reduces the data dimension but also enhances the discriminability of features through linear transformations, enabling a more precise representation of the characteristics of disease spots. During the forward propagation process, these high-quality features assist the model in discriminating disease spot regions more rapidly and effectively, thereby achieving higher accuracy in practical chili disease detection.



(**a**) The architecture of the RepGFPN.

(**b**) The layout situation of the CSPStage module.

(**c**) The architecture of the Rep module.

Figure 5. Multi-scale feature fusion network structure and module design. (**a**) RepGFPN removes the up-sample connection and uses the CSPStage module for feature fusion. (**b**) The CSPStage module performs feature fusion operations. ×N means that there are N structures, which are the same as those in the dashed box. (**c**) The technology of 3 × 3 Rep represents a method for reparameterizing models, leading to decreased computational requirements and enhanced model efficiency.

2.2.3. Lightweight DySample Upsampling

The upsampling operator of YOLOv8 employs the nearest neighbor [23] interpolation algorithm for upsampling. It merely utilizes pixels within a relatively narrow range for prediction and fails to preserve the details of the feature map effectively. In practical situations, there may be instances where small objects are difficult to detect against complex background settings, resulting in a reduction in image details. To tackle this problem, kernel-based dynamic upsampling algorithms (such as CARAFE [24], FADE [25], and SAPA [26]) have been proposed, which have a larger receptive field and can achieve high-quality upsampling. Nevertheless, the integration of the CARAFE algorithm's dynamic convolution with the supplementary self-network for dynamic kernel generation leads to a notable escalation in computational demands and temporal expenditure, posing obstacles for real-time detection. Therefore, we adopt the DySample [27] algorithm. Dysample uses a point-based sampling method, balancing the relationship between performance improvement and computational cost, and improves the detection accuracy at a small cost. Figure 6 distinctly presents the elaborate network framework of Dysample.

Firstly, a feature vector X, characterized by dimensions $C \times H_1 \times W_1$, is introduced as the input. Here, C represents the channel count, whereas H_1 and W_1 individually signify the vertical and horizontal dimensions, respectively, of the feature map. The data from all channels of this feature map collectively contribute to forming a preliminary feature map, which supplies sufficient information for subsequent sampling. This feature vector captures all possible spatial information and subtle features that are associated with disease characteristics.

Subsequently, the sampling point generator will establish a sampling set S in accordance with specific rules and algorithms. The dimension of this sampling set S is $2g \times H_2 \times W_2$. The role of the sampling point generator is to dynamically adjust the positions of sampling points based on specific feature regions in the feature map, in order to capture finer image details. This method of dynamically generating sampling points can automatically identify important areas, enabling the model to focus on small yet crucial disease features, thereby improving detection accuracy. Subsequently, the grid sample function is employed to construct a new feature vector X′ by bilinear interpolation. This operation effectively smooths the features and accomplishes more detailed upsampling with less computational cost, enhancing the detection effect for small targets (such as minute diseases on leaves). The process can be defined as follows:

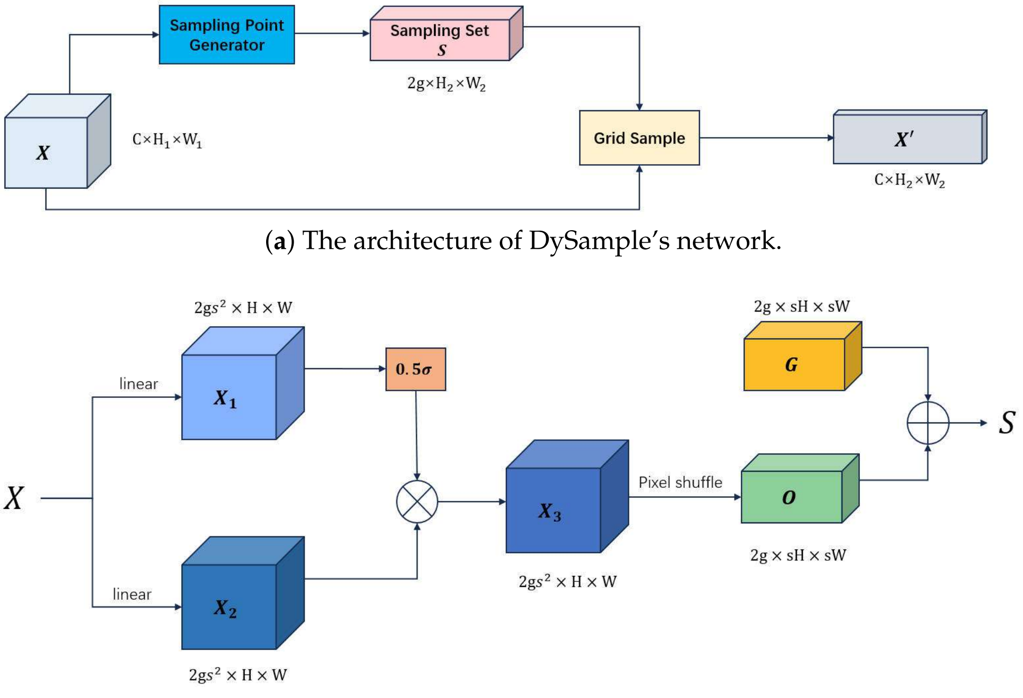
$$X' = Gridsample(X, S) \tag{2}$$

Illustrated in Figure 6b is the process by which the sampling point generator generates the sampling set S. In practical scenarios, the field environment is complex and variable. The lesions of Phytophthora blight on chili leaves typically manifest as subtle changes in color and texture. To accurately detect these small and easily overlooked lesions, DySample's offset matrix generation and dynamic adjustment process adaptively adjusts the sampling positions based on input features, particularly demonstrating good flexibility for lesions of different sizes and shapes. The sampling point generator accepts a feature matrix x as input and uses two linear layers to generate two offset matrices X_1 and X_2, respectively, and the sizes of these two matrices are $2gs^2 \times H \times W$. To effectively prevent the overlap of the sampling positions, a dynamic adjustment ability is introduced. First, the generator activates X_1 through the sigmoid function and then multiplies it by a static factor of 0.5, achieving a soft offset adjustment. The benefit of doing so is that it becomes more sensitive to areas containing fine lesions without losing the overall structure. The adjusted X_1 is then element-wise multiplied with the second offset X_2 to generate a new offset matrix X_3. X_3 is then reshaped into an offset O with a size of $2g \times sH \times sW$ through the pixel shuffle operation. This operation not only reshapes the dimension of the offset but also achieves efficient upsampling of the feature map by re-arranging the elements and preserving the crucial spatial information. Ultimately, the sampling set S is derived through the combination of the offset O and the initial grid G, which is expressed in the subsequent formula.

$$O = 0.5 \cdot \sigma(\mathrm{linear}_1(X)) \cdot \mathrm{linear}_2(X) \quad (3)$$

$$S = G + O \quad (4)$$

In summary, DySample leverages the generation and dynamic adjustment of offset matrices to enable the model to perform a more precise sampling of the delicate and complex lesion areas on chili plants. By adaptively adjusting the positions of sampling points during the upsampling process, it effectively retains the spatial information and crucial details of the feature map, without wasting computational resources on redundant information. This allows the model to more sensitively capture the characteristics of diseases, improving the detection accuracy of small areas even in complex field backgrounds. Furthermore, DySample's point sampling method achieves efficient feature preservation at a low computational cost, making the model both accurate and suitable for real-time detection tasks.



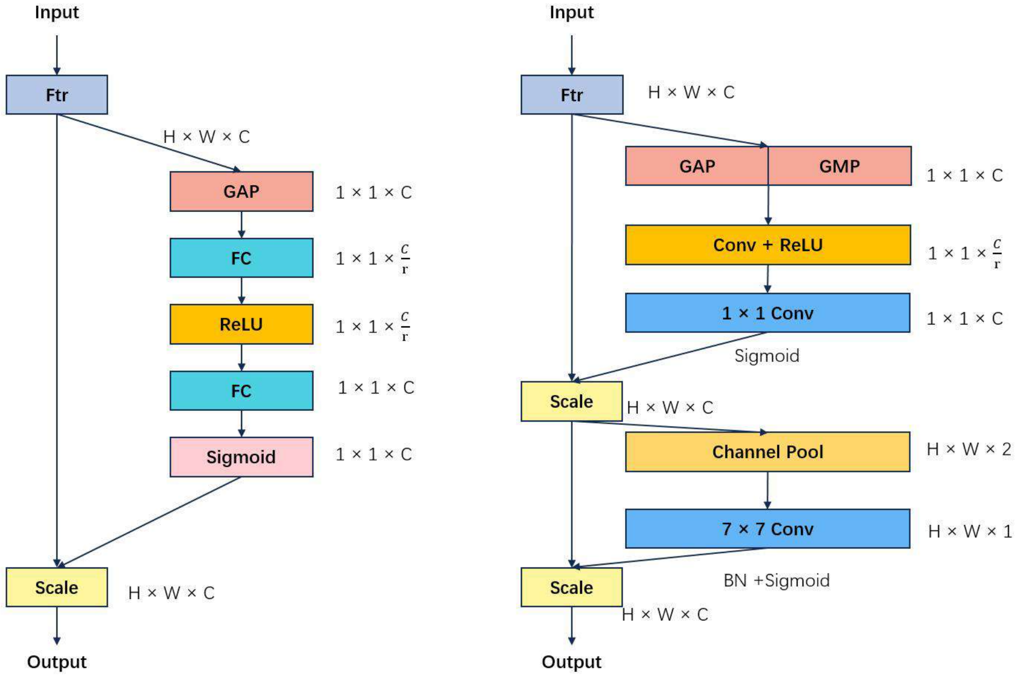
(**a**) The architecture of DySample's network.

(**b**) The configuration of the Generator module's structure.

Figure 6. The design of DySample's modules and its network architecture. (**a**) X represents the input feature, X' represents the upsampled feature, and S represents the sampling set. The sampling point generator produces a sampling set, which is then utilized to resample the input feature through the grid sampling function. (**b**) X_1,X_2,X_3 represent offsets with a size of $2gs^2 \times H \times W$. O represents the generated offset. G represents the original grid and σ denotes the sigmoid function.

2.2.4. CoordAtt Module

In the detection of chili Phytophthora blight, situations such as overlapping detection targets and complex background environments may be confronted. We can choose to introduce the attention mechanism to effectively solve these problems. The attention mechanism enables the network to focus its attention on pertinent information, while simultaneously filtering out less crucial details, thereby minimizing distraction from unnecessary data. This not only enhances detection accuracy but also greatly improves the efficiency of computing resource utilization [28]. Specifically, the SE channel attention mechanism [29] dynamically adjusts the significance of each channel, effectively boosting the network's capacity to prioritize and amplify essential information. However, its limitation lies in only focusing on the differences between channels and ignoring the position information in the image space, which to a certain extent constrains the network's detection ability of chili Phytophthora blight. In contrast, the CBAM [30] attention mechanism achieves a more comprehensive feature extraction. It incorporates both channel-wise focus and spatial awareness, taking into account the prominence of features across various channels and the cruciality of features within their spatial arrangements. Although this mechanism significantly improves the meticulousness of feature extraction, it also correspondingly increases the computational cost. Furthermore, CBAM primarily concentrates on extracting local feature details, potentially lacking somewhat in the integration and comprehension of holistic, global features. At the same time, there may be slight precision losses in the processing of spatial information, which could affect the final detection performance to a certain extent. Figure 7 displays the network architectures of SE and CBAM.



(**a**) SE (**b**) CBAM

Figure 7. The structure of SE (**a**) and CBAM (**b**).

The CoordAtt attention (CA) mechanism [31], takes into account both spatial and channel information. To manage the computational expense, an innovative approach is utilized, which incorporates spatial information within the channels, thereby attaining a more expansive informational vantage point. This attention mechanism abandons the conventional 2D global pooling approach, opting for two separate 1D global pooling processes that efficiently consolidate positional data along the vertical and horizontal axes, respectively. By employing this method, not only does it significantly reduce the computational intricacy, but it also averts the likelihood of detail loss that might arise from the exhaustive amalgamation of spatial data. This method maintains specific traits along the spatial dimension, empowering the model to discern and harness the unique attributes of images or feature maps more proficiently within these two orientations. Consequently, it boosts the network's capacity to understand complex spatial relationships. The CA attention mechanism is illustrated in Figure 8, detailing its specific process.

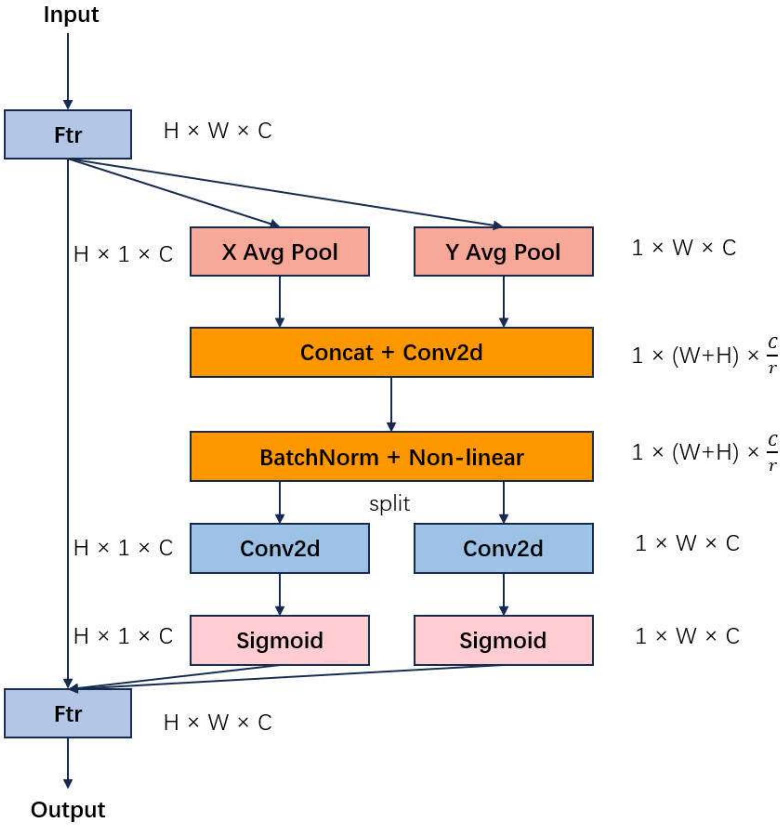


Figure 8. The structure of CA.

At the onset, the incoming feature representation U, which possesses the shape $H' \times W' \times C'$, undergoes the Ftr transformation, yielding an altered feature representation X that has the shape $H \times W \times C$. As a result, the model is able to capture disease characteristics more clearly, effectively preserving the vital information in the lesion areas and making the location of the disease stand out more prominently. Following that, separate encoding processes are applied along the width and height dimensions, utilizing pooling elements with dimensions (H, 1) for the horizontal axis and (1, W) for the vertical axis, respectively. The resulting outputs for the c-th channel at height h and width w are depicted as follows:

$$z_c^h(h) = \frac{1}{W} \sum_{0 \le i < W} x_c(h, i) \tag{5}$$

$$z_c^w(w) = \frac{1}{H} \sum_{0 \le j < H} x_c(j, w) \tag{6}$$

The above formula integrates features from diverse directions and outputs a pair of feature maps with known directions, namely z^h and z^w. By preserving spatial information in one direction while capturing long-range relationships in the other, this attention module stands apart from the global pooling compression method. The positional information is adeptly incorporated into the produced feature maps, enhancing the network's capability to pinpoint the target with greater accuracy. Subsequently, a transposed concatenation operation is performed on z^h and z^w, generating a composite feature map. Afterwards, it undergoes the transformation function F_1 which employs a 1 × 1 convolutional layer to compress the dimensionality and introduce nonlinearity, yielding the feature map $f \in \mathbb{R}^{(H+W) \times 1 \times \frac{C}{r}}$. This enables the model to enhance its focus on small lesion areas without increasing the computational burden. This allows the model to automatically distinguish subtle differences between diseased and normal areas in complex backgrounds, reducing false detections. The formula is presented as follows:

$$f = \delta(F_1([z^h, z^w])) \tag{7}$$

Among them, the symbol [,] signifies the joining of elements along the spatial dimension, while δ denotes the application of a nonlinear activation function. Next, the feature map f is segmented along the spatial dimension, yielding a separation into two distinct feature vectors: $f^h \in \mathbb{R}^{H\times 1\times(\frac{C}{r})}$ and $f^h \in \mathbb{R}^{1\times W\times(\frac{C}{r})}$. To elevate the channel counts of these two feature vectors and revert them back to the initial channel count C, we individually employ two 1 × 1 convolutional kernels, namely F_h and F_w. This type of 1 × 1 convolution can not only increase the channel number without altering the spatial dimensions of the feature map but also incorporate nonlinear combinations via the learning process of the convolutional kernels, subsequently boosting the representational power of the features. This approach not only preserves the spatial dimensions of the feature map but also enhances the nonlinear combination of features through convolutional learning, ensuring that the model has higher discriminative power and detail restoration capabilities when detecting lesion spots. Subsequently, when paired with the sigmoid non-linear activation function, we acquire $g^h \in \mathbb{R}^{H\times 1\times C}$ and $g^w \in \mathbb{R}^{1\times W\times C}$. The formula is presented as follows:

$$g^h = \sigma(F_h(f^h)) \tag{8}$$

$$g^w = \sigma(F_w(f^w)) \tag{9}$$

The Sigmoid function is denoted by σ, and F_h and F_w represent distinct 1 × 1 convolutional operations. We expand g^h and g^w and subsequently utilize them as the attention coefficients for the feature map. Through the attention coefficients, the CA mechanism assigns higher attention to disease-infected areas and integrates this with the original feature map. This enables the model to prioritize the processing of lesion areas and ignore other irrelevant regions in complex field scenarios, thereby enhancing detection accuracy. In practical applications, this implies that regardless of the orientation or location of the lesion, the CA attention mechanism can assist the model in achieving precise identification. Then, the calculation formula of the CA attention mechanism is as follows:

$$y_c(i,j) = x_c(i,j) \times g_c^h(i) \times g_c^w(j) \tag{10}$$

2.2.5. Inner-MPDIoU

YOLOv8 includes the bounding box regression loss function and the classification loss function. The bounding box regression loss function's main purpose is to determine the quantitative discrepancy between the predicted and actual bounding boxes [32]. The accuracy of the regression process for bounding boxes directly influences the overall precision of object detection. The loss function for bounding box regression in YOLOv8 is CIoU and its mathematical formulation for computation is provided below:

$$\mathrm{CIoU} = \mathrm{IoU} - \frac{d^2}{c^2} - \alpha v \tag{11}$$

$$\alpha = \frac{v}{(1 - \mathrm{IoU}) + v} \tag{12}$$

$$v = \frac{4}{\pi^2}\left(\arctan\left(\frac{w^{\mathrm{gt}}}{h^{\mathrm{gt}}}\right) - \arctan\left(\frac{w}{h}\right)\right)^2 \tag{13}$$

The IoU in this formula signifies the ratio of the overlapping area to the combined area of the predicted bounding box and the actual bounding box. d^2 signifies the squared distance between the centers of the predicted box and the true box. Meanwhile, c^2 denotes the square of the diagonal length of the smallest bounding rectangle that encompasses both boxes. The balance parameter, α, serves to adjust the weight given to the effect of the aspect ratio in the equation. v is the parameter that measures the aspect ratio of the predicted box and the true box. $w^g t$ and $h^g t$ represent the dimensions of the target box in terms of width and height, whereas w and h signify the corresponding dimensions of the true box.

For the detection of chili Phytophthora blight or similar intricate object detection endeavors, achieving the accurate detection of minute targets holds paramount significance. However, the traditional CIoU and its improved versions such as SIoU may have constraints when dealing with certain specific situations. For instance, if the aspect ratio of the predicted bounding box matches that of the true bounding box yet they vary in size, while their center points coincide, these loss functions might fail to adequately discern and refine this discrepancy, ultimately limiting the object detection's efficacy. To overcome the shortcomings of CIoU, MPDIoU [33] was proposed, and its calculation formula is as follows:

$$\text{MPDIoU} = \text{IoU} - \frac{d_1^2}{w^2 + h^2} - \frac{d_2^2}{w^2 + h^2} \tag{14}$$

$$d_1^2 = (x_1 - x_1^{\text{gt}})^2 + (y_1 - y_1^{\text{gt}})^2 \tag{15}$$

$$d_2^2 = (x_2 - x_2^{\text{gt}})^2 + (y_2 - y_2^{\text{gt}})^2 \tag{16}$$

In the formula, w and h represent the width and height of the input image. d_1^2 is the square of the distance calculated between (x_1, y_1) and $(x_1^g t, y_1^g t)$, and d_2^2 is the square of the distance calculated between (x_2, y_2) and $(x_2^{\text{gt}}, y_2^{\text{gt}})$. (x_1, y_1) signifies the coordinates of the upper left corner of the predicted box, while (x_2, y_2) denotes the coordinates of its lower right corner. Similarly, $(x_1^{\text{gt}}, y_1^{\text{gt}})$ and $(x_2^{\text{gt}}, y_2^{\text{gt}})$ represent the respective coordinates of the upper left and lower right corners of the true box.

The core of MPDIoU lies in utilizing the distance between the top-left and bottom-right key points to evaluate the degree of matching between the predicted bounding box and the ground truth bounding box. Compared to other methods that require separate calculations of multiple indicators such as IoU (Intersection over Union), center point distance, aspect ratio, and more, MPDIoU focuses solely on the distance between these two corner points. By combining the coordinates of the top-left and bottom-right corners with the image's size information, MPDIoU implicitly covers several key pieces of information, including non-overlapping area, center point distance, and deviations in width and height. MPDIoU cleverly represents all these factors through the corner point distance, making the calculation process more concise and clear. It is worth noting that when the aspect ratios of the predicted bounding box and the ground truth bounding box are the same, the LMPDIoU value for a predicted bounding box located inside the ground truth bounding box will be lower than that for a predicted bounding box located outside. This characteristic ensures the accuracy of bounding box regression, resulting in more compact and less redundant predicted bounding boxes.

However, both IoU and MPDIoU exhibit relatively slow convergence rates. In this paper, the MPDIoU will be improved by adopting the Inner IoU [34] approach. Inner IoU incorporates supplementary bounding boxes as a means to compute the Intersection over Union loss. These bounding boxes serve as intermediate media for calculating the IoU loss, providing additional information and guidance for the optimization process. Instead of directly calculating the IoU between the predicted bounding box and the ground truth bounding box, Inner IoU provides a more intricate assessment of the localization precision of the bounding box by analyzing the extent of the overlap between an auxiliary bounding box and either the ground truth bounding box or the predicted bounding box. For different detection tasks, a scaling factor is introduced to adjust the dimensions of the auxiliary bounding box and determine the loss. The calculation formula is as follows:

$$b_l^{\text{gt}} = x_c^{\text{gt}} - \frac{w^{\text{gt}} \cdot \text{ratio}}{2}, b_r^{\text{gt}} = x_c^{\text{gt}} + \frac{w^{\text{gt}} \cdot \text{ratio}}{2} \tag{17}$$

$$b_t^{\mathrm{gt}} = y_c^{\mathrm{gt}} - \frac{h^{\mathrm{gt}} \cdot \mathrm{ratio}}{2}, b_b^{\mathrm{gt}} = y_c^{\mathrm{gt}} + \frac{h^{\mathrm{gt}} \cdot \mathrm{ratio}}{2} \tag{18}$$

$$b_l = x_c - \frac{w \cdot \mathrm{ratio}}{2}, b_r = x_c + \frac{w \cdot \mathrm{ratio}}{2} \tag{19}$$

$$b_t = y_c - \frac{h \cdot \mathrm{ratio}}{2}, b_b = y_c + \frac{h \cdot \mathrm{ratio}}{2} \tag{20}$$

$$\mathrm{inter} = \left(\min(b_r^{\mathrm{gt}}, b_r) - \max(b_l^{\mathrm{gt}}, b_l)\right) \times \left(\min(b_b^{\mathrm{gt}}, b_b) - \max(b_t^{\mathrm{gt}}, b_t)\right) \tag{21}$$

$$\mathrm{union} = (w^{\mathrm{gt}} \cdot h^{\mathrm{gt}}) \cdot (\mathrm{ratio})^2 + (w \cdot h) \cdot (\mathrm{ratio})^2 - \mathrm{inter} \tag{22}$$

$$\mathrm{IoU}^{\mathrm{inner}} = \frac{\mathrm{inter}}{\mathrm{union}} \tag{23}$$

Among them, the ratio is the scale factor, and its usual value range is [0.5, 1.5]. When the ratio is less than 1, the auxiliary bounding box is smaller than the actual bounding box, causing the effective regression range to concentrate in the overlapping area with a high Intersection over Union (IoU) while ignoring the parts outside the boundary. This narrowed scope allows the loss function to focus on finely aligning the overlapping regions of the two boxes. In this scenario, even minor prediction deviations can lead to significant changes in the loss, resulting in larger gradient values. This enables the model to accelerate the alignment adjustment for high IoU samples and improve convergence speed. Conversely, when the ratio is greater than 1, the auxiliary bounding box is larger than the actual bounding box, thereby expanding the effective regression range. In this way, even if the overlap between the predicted box and the ground truth box is small, the loss function can still capture the overall offset between them, allowing the model to focus on broader positional information and improve the regression effect for low IoU samples. Additionally, larger auxiliary bounding boxes reduce sensitivity to minor deviations, leading to relatively smoother gradient changes. This helps steadily bring the predicted box closer to the ground truth box, thereby smoothly optimizing significantly offset boxes and avoiding excessive fluctuations. This adjustment allows the model to balance the regression effects of different IoU samples, and through dynamic adjustment of the ratio, it can effectively optimize the overall accuracy of low IoU samples while improving the convergence speed of high IoU samples. As this paper focuses on detecting chili Phytophthora blight, it necessitates a heightened level of image detail. Consequently, a scaling factor of 1.2 has been chosen for this purpose. The central coordinates of the ground truth box (gt) and its auxiliary bounding box are denoted as $(x_c^{\mathrm{gt}}, y_c^{\mathrm{gt}})$, respectively, while those of the anchor box and its auxiliary bounding box are labeled (x_c, y_c). The height and width of the ground truth box are $h^g t$ and $w^g t$, respectively, and the height and width of the anchor box are h and w, respectively. The auxiliary bounding box of the ground truth box is defined by its upper boundary $b_t^g t$, lower boundary $b_b^g t$, left boundary $b_l^g t$ and right boundary $b_r^g t$, respectively. Correspondingly, b_t, b_b, b_l, b_r are, respectively, the upper, lower, left, and right boundaries of the auxiliary bounding box of the anchor box. By incorporating Inner IoU into MPDIoU, we derive Inner-MPDIoU, and its computational formula is stated as follows:

$$\mathrm{MPDIoU}^{\mathrm{inner}} = \mathrm{IoU}^{\mathrm{inner}} - \frac{d_1^2}{w^2 + h^2} - \frac{d_2^2}{w^2 + h^2} \tag{24}$$

In summary, the Inner-MPDIoU loss function significantly enhances the performance of object detection tasks by integrating the advantages of both the MPDIoU and Inner-IoU methods.

2.2.6. YOLOv8-GDCI Model

This paper makes improvements based on YOLOv8n and proposes a network model named YOLOv8-GDCI for detecting chili Phytophthora blight, whose network structure is shown in Figure 9. We introduce the RepGFPN feature aggregation network and reconstruct the original PAN structure. The CSPStage module, devised for the purpose of feature fusion, utilizes varying numbers of channels tailored to features of diverse scales, thereby enhancing the accuracy of detecting chili Phytophthora blight without imposing undue computational overhead. Then, the lightweight dynamic upsampling operator Dysample based on point sampling is introduced, which brings a larger receptive field and enables it to identify and locate the dense disease areas of chilies more accurately, significantly enhancing the practicability and accuracy of the model. Next, the CA attention mechanism is appended, considering both spatial and channel information, which significantly improves the network's capacity for identifying and extracting critical features. Finally, in view of the poor performance of CIoU in dealing with overlapping objects, Inner-MPDIoU is adopted to quantitatively calculate the difference between the estimated bounding box and the actual bounding box and improve the efficiency of model regression. In summary, the YOLOv8-GDCI model, by undergoing a series of refinements, has successfully elevated both the detection precision of chili Phytophthora blight and its adaptability as well as resilience within intricate disease settings. This achievement offers pivotal technical backing for intelligent oversight, prevention, and management strategies in smart agricultural environments.

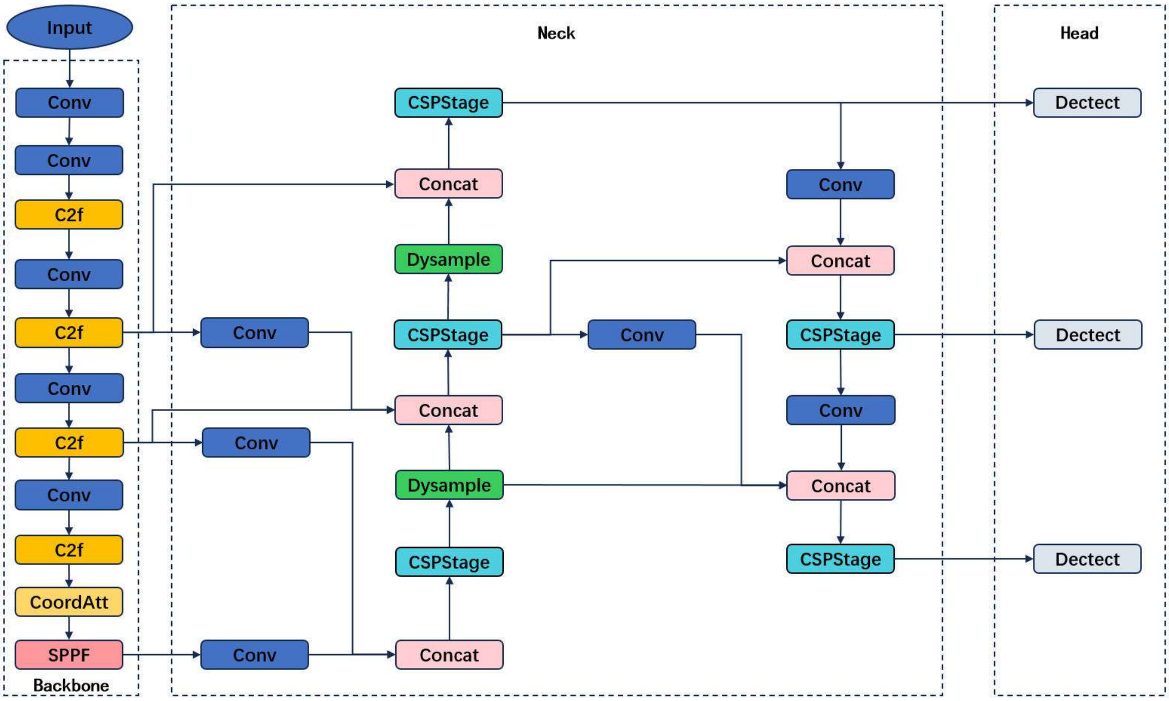


Figure 9. The network structure diagram of our YOLOv8-GDCI algorithm.

3. Results

3.1. Experimental Environment and Model Evaluation Metrics

The GPU selected for this experiment environment is an NVIDIA GeForce RTX 4060 Laptop 8G. The device is manufactured by NVIDIA and procured from Nanjing, China. The software environment is Pytorch 2.3.1, Python 3.9.19, and Cuda 12.1. The initial parameters are outlined in Table 2 as follows:

This article conducts model evaluation based on the precision (P), recall (R), mean average precision (mAP), parameter quantity (Params), and frames per second (FPS). The relevant formulas are as follows:

$$P = \frac{TP}{TP + FP} \times 100\% \tag{25}$$

$$R = \frac{TP}{TP + FN} \times 100\% \tag{26}$$

$$\mathrm{AP} = \int_0^1 P(R)\,\mathrm{d}R \tag{27}$$

$$\mathrm{mAP} = \frac{1}{n}\sum_{i=1}^{n} \mathrm{AP}_i \times 100\% \tag{28}$$

$$\mathrm{FPS} = \frac{1}{T} \tag{29}$$

Specifically, TP denotes the count of samples where both the prediction and the actual label are positive. FP represents the number of samples where the prediction is positive, but the actual label is negative. FN, on the other hand, signifies the count of samples incorrectly predicted as negative while the actual label is positive. Precision gauges the model's accuracy in predicting positive samples, while recall measures the extent of the model's positive predictions covering the actual positive samples. The variable n signifies the total number of distinct label categories, and T denotes the duration the model takes to analyze a single image.

Table 2. Experimental parameter setting.

Parameter	Value
Epochs	400
Workers	6
Optimizer	SGD
Lr0	0.01
Lr1	0.01
momentum	0.937

3.2. Ablation Experiment

To validate the effectiveness of the refined approach chosen, we performed ablation experiments on the test set. The experimental indicators are precision, recall, mAP0.5, Params, and FPS. As depicted in Table 3 below, the results are presented. For further detailed analysis and in-depth exploration of the accuracy of the model when targeting specific detection objects, the value of map0.5% for each specific detection object is given in Table 4.

Table 3. The ablation experiment of YOLO-GDCI. '-' represents that this part is not used, and '✓' represents that this part is used.

RepGFPN	Dysample	CA	Inner-MPDIoU	P (%)	R (%)	mAP0.5 (%)	Params(M)	FPS
-	-	-	-	0.905	0.824	0.872	3.15	232.8
✓	-	-	-	0.913	0.832	0.881	3.25	209.3
✓	✓	-	-	0.916	0.830	0.883	3.26	133
✓	✓	✓	-	0.910	0.844	0.886	3.27	156
✓	✓	✓	✓	0.914	0.842	0.889	3.27	174

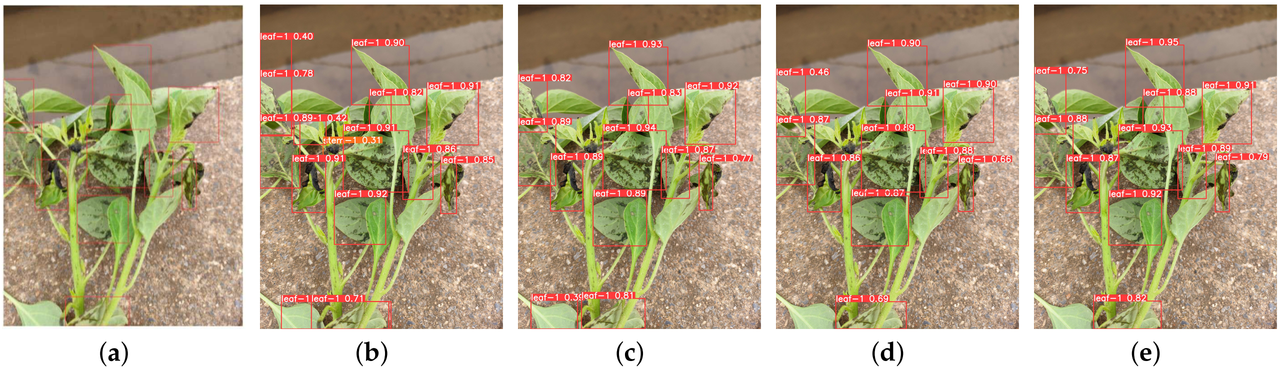
Table 4. The mAP0.5 (%) value of each category. '-' represents that this part is not used, and '✓' represents that this part is used.

RepGFPN	Dysample	CA	Inner-MPDIoU	Leaf-1	Fruit-1	Stem-1
-	-	-	-	0.953	0.817	0.847
✓	-	-	-	0.953	0.833	0.857
✓	✓	-	-	0.959	0.843	0.846
✓	✓	✓	-	0.960	0.82	0.868
✓	✓	✓	✓	0.960	0.835	0.871

Firstly, we substituted the neck network of the YOLOv8 base model with the RepGFPN network, allowing the model to manipulate the feature expressions of different scales more flexibly and leading to enhancements of 0.8%, 0.8%, and 0.9% in detection precision, recall, and mAP0.5, respectively. Particularly, for the two key detection targets of fruit-1 and stem-1, the mAP0.5 values increased by 1.6% and 1%, respectively, strongly substantiating the superiority of the RepGFPN module in enhancing the multi-scale feature processing capability. We then substituted the Upsample module with the Dysample upsampling module. This modification effectively improved the model's precision and mAP0.5 by 0.3% and 0.2%, respectively, while keeping the number of parameters almost unchanged. Although the recall slightly decreased, this change was within an acceptable range. It is worth noting that the Dysample module brought improvements of 0.7% and 1%, respectively, in the mAP0.5 of leaf-1 and fruit-1, indicating its excellent performance in retaining the details of the feature map. Subsequently, we integrated the CA attention mechanism into the model. This mechanism significantly enhanced the model's precise positioning ability for the target of interest by ingeniously embedding spatial information into the channels. With little change in the number of parameters and FPS, the recall and mAP0.5 increased by 1.4% and 0.3%, respectively. Particularly, the mAP0.5 value of stem-1 increased by 2.2%, reflecting the huge potential of the CA attention mechanism in improving detection accuracy. Finally, we replaced the loss function from CIoU to Inner-MPDIoU, which not only improved the model's precision and mAP0.5 by 0.4% and 0.3%, respectively, but also significantly increased the detection speed by 41. Especially, in the detection tasks of fruit-1 and stem-1, the mAP0.5 values increased by 1.5% and 0.3%, respectively, further verifying the effectiveness of the Inner-MPDIoU loss function in handling overlapping prediction boxes and adding auxiliary boxes, offering a more precise and streamlined approach for identifying chili Phytophthora blight. The experimental results prove that each improvement point adopted in this paper is significant. When the four improvements are applied simultaneously, the improvement effect is the most optimal.

3.3. Experimental Comparison of Different Feature Aggregation Networks

To validate the rationality and superiority of replacing the original neck network with RepGFPN, in the context of detecting chili Phytophthora blight, this paper conducted comparative experiments with mainstream feature fusion networks such as PAN, GFPN, and BiFPN [35]. The experimental findings are showcased in Table 5. The results reveal that replacing the neck network with RepGFPN achieved the best results. Its precision increased by 0.8%, 1.6%, and 1.2%, respectively, compared to PAN, GFPN, and BiFPN. The recall increased by 0.8% and 0.9%, respectively, compared to PAN and BiFPN. The mAP0.5 increased by 0.9%, 0.4%, and 1.6%, respectively, compared to PAN, GFPN, and BiFPN. Despite the marginal rise in parameter count and the slight dip in FPS due to the integration of RepGFPN, the experimental findings unequivocally establish that the FPS index comfortably satisfies the thresholds for real-time detection. To better demonstrate the effectiveness of RepGFPN, we have visualized the experimental results as shown in the Figure 10. To more intuitively demonstrate the advantages of RepGFPN, we have visualized the experimental results and presented them in the following figures. From the figures, it can be seen that the basic PAN and BiFPN structures perform poorly, with misdetections and relatively low accuracy. Although the accuracy of GFPN is similar to that of RepGFPN, it also experiences misdetections. In contrast, RepGFPN exhibits the best performance. It can accurately detect targets in complex scenarios while minimizing misdetections, demonstrating its robustness and effectiveness. Through visualization, we have a clearer view of the advantages of RepGFPN and the importance of our refined improvements to the feature network.



(**a**) (**b**) (**c**) (**d**) (**e**)

Figure 10. Visual Demonstration Using Different Feature Networks. (**a**) Truth (**b**) PAN (**c**) GFPN (**d**) BiFPN (**e**) RepGFPN

Table 5. The results of the feature fusion comparison experiment.

Model	P (%)	R (%)	mAP0.5 (%)	Params (M)	FPS
PAN	0.905	0.824	0.872	3.15	232
GFPN	0.897	0.841	0.877	3.36	188
BiFPN	0.901	0.823	0.865	1.99	211.5
RepGFPN	0.913	0.832	0.881	3.25	209.8

3.4. Comparative Experiments of Different Upsampling Algorithms

On the basis of replacing the neck network with RepGFPN, we further improved the upsampling operator and replaced the nearest neighbor upsampling algorithm of the base YOLOv8 with the Dysample algorithm. We conducted comparative experiments with the Nearest and CARAFE algorithms, aiming to highlight the advantages of the Dysample algorithm. The detailed data of the experimental results are listed below and summarized in Table 6.

Table 6. The comparative experimental results of the upsampling algorithms.

Model	P (%)	R (%)	mAP0.5 (%)	Params (M)	FPS
Nearest	0.905	0.824	0.872	3.25	232
Carafe	0.907	0.828	0.879	3.56	133
Dysample	0.916	0.83	0.883	3.26	194.9

The Nearest algorithm performed poorly in the metrics of precision, recall, and mAP0.5. This is because the Nearest algorithm only considers the information of adjacent pixels, which may result in image distortion. The CARAFE algorithm performed better than the Nearest algorithm, with its precision, recall, and mAP0.5 increasing by 0.2%, 0.4%, and 0.5%, respectively. However, the performance improvement of the CARAFE algorithm was accompanied by a significant increase in complexity, which directly led to a substantial increase in the number of model parameters and a significant decrease in frames per second (FPS), posing challenges to the computational efficiency and real-time performance in practical applications. While the Dysample algorithm achieved good results in various metrics while maintaining high efficiency. Its precision, recall, and mAP0.5 values increased by 0.9%, 0.2%, and 0.4%, respectively, compared to the CARAFE. The Dysample algorithm performed well in the control of the number of parameters, did not bring much increase compared to the Nearest algorithm, and the decrease in its FPS also remained within an acceptable range, achieving a good balance between performance and efficiency.

3.5. The Contrast Experiment of the Attention Mechanism

Utilizing the aforementioned enhancements, we incorporate the CA attention mechanism to facilitate the model's ability to discern and capture the pivotal features of the target under scrutiny. Concurrently, to ascertain the most effective placement for the CA attention mechanism, we executed experiments under the subsequent three circumstances:

a. Only added at the end of the backbone network
b. Only added at the end of the neck network
c. It is added, respectively, at the end of the backbone network and the neck network.

Table 7 displays the experimental data. It is discernible that the model attains its peak performance when the CA attention mechanism is appended to the terminus of the backbone network. Its recall and mAP indicators all show the highest level. Although the precision value slightly decreases, this change has little impact on the overall performance of the model. This is because the backbone network is responsible for extracting both low-level and high-level features from the original input image, including spatial information and semantic information. By applying the CA (Channel Attention or Contextual Attention, as applicable) mechanism to the backbone network, it can better focus on important spatial and channel features, thereby establishing clear disease-infected area features for the model at the early stage. Introducing CA attention in the latter part of the backbone network enables the model to gradually weight and highlight disease information across different feature levels, reducing interference from unimportant regions and providing cleaner features for subsequent feature fusion. Since the neck network focuses on feature fusion rather than initial feature extraction, adding attention here is helpful but less significant than adding it at the end of the backbone. If the CA attention mechanism is applied to both the backbone and neck networks, it may lead to the neck already receiving features processed by the backbone, causing the model to over-weight the same features, resulting in feature redundancy. This redundancy can degrade the model's generalization ability because re-weighting already enhanced features can cause the model to bias towards specific features while ignoring other potentially important information, leading to a decrease in detection performance. Therefore, in YOLOv8, adding the CA attention mechanism solely to the backbone network allows for more efficient focusing on key features, improving the detection accuracy of the YOLOv8 model for chili Phytophthora blight.

Table 7. The contrast experimental results of different positions of the CA attention mechanism.

Model	P (%)	R (%)	mAP0.5 (%)	Params (M)
Base	0.916	0.830	0.883	3.26
Base-a	0.910	0.844	0.886	3.27
Base-b	0.901	0.818	0.877	3.27
Base-c	0.913	0.832	0.881	3.27

To gauge the effect of diverse attention mechanisms on the aforementioned model's performance, and contrast the performance of the CA attention mechanism against other attention mechanisms specifically in detecting chili Phytophthora blight, we incorporated comparative experiments by appending the channel attention mechanism (SE), spatial and channel attention mechanisms (CBAM, EMA [36]), and the parameter-free attention mechanism (SiMAM [37]) to the backbone's end. The outcomes of the experimental analysis are depicted in Table 8.

Specifically, the CA attention mechanism surpassed other compared attention mechanisms in both the recall and the mAP0.5, two key indicators. Its recall achieved improvements of 1.4%, 0.8%, 0.6%, and 2.2%, respectively, compared to None, SE, CBAM, EMA, and SiMAM, while the mAP0.5 was, respectively, 0.3%, 0.2%, 0.7%, 1.5%, and 0.5% higher. This result not only demonstrates the outstanding ability of the CA attention mechanism to accurately capture the features related to the disease but also further validates its effectiveness in improving the comprehensiveness and accuracy of detection. While the integration of the CA attention mechanism did marginally elevate the model's parameter count, this increment remained at a relatively low level in comparison to alternative attention mechanisms. This indicates that while improving performance, the CA attention mechanism also maintains good model efficiency, which is conducive to achieving faster inference speed and lower resource consumption in practical applications.

Table 8. The contrastive experimental results of different attention mechanisms.

Model	P (%)	R (%)	mAP0.5 (%)	Params (M)
None	0.916	0.830	0.883	3.26
SE	0.906	0.836	0.884	3.27
CBAM	0.912	0.838	0.879	3.33
EMA	0.926	0.822	0.871	3.29
SiMAM	0.911	0.843	0.881	3.26
CA	0.910	0.844	0.886	3.27

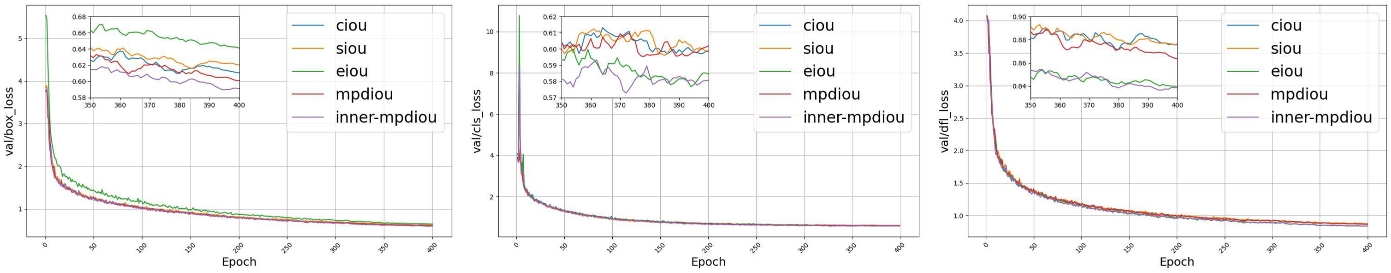
3.6. Contrast Experiments of Different Loss Functions

We replaced the original loss function of YOLOv8 with Inner-MPDIoU. Taking the model after the above improvement as the benchmark, multiple loss functions were conducted in multiple groups of experiments to compare their performances in terms of precision, recall, and mAP0.5; the results are shown in the Table 9. From the Table 9, we know that the Inner-MPDIoU loss function performs most outstandingly in the comprehensive evaluation metric of mAP0.5, achieving improvements of 0.3%, 1.1%, 0.7%, and 0.3%, respectively, compared to traditional or variant loss functions such as CIoU, SIoU, EIoU, and MPDIoU, fully verifying its significant advantage in improving the overall performance of the model.

Table 9. The contrastive experimental results of different loss functions.

Model	P (%)	R (%)	mAP0.5 (%)
CIoU	0.910	0.844	0.886
SIoU	0.903	0.853	0.878
EIoU	0.928	0.823	0.882
MPDIoU	0.917	0.839	0.886
Inner-MPDIoU	0.914	0.842	0.889

To delve deeper into the performance disparities among various loss functions, we analyzed from three aspects: box_loss, cls_loss, and dfl_loss. box_loss reflects the proximity between the predicted bounding box and the actual bounding box. A smaller value indicates a higher degree of overlap, thereby indicating more accurate positioning; cls_loss gauges the model's performance in classifying, and its reduction means more accurate category prediction; dfl_loss, as a measure of feature point loss, directly correlates with the model's proficiency in capturing and learning pertinent feature information, determining its optimization level. Furthermore, as evident from Figure 11, all the loss functions involved in the comparison are able to converge effectively, but Inner-MPDIoU exhibits the lowest values among the three indicators of box_loss, cls_loss, and dfl_loss. This intuitive data comparison confirms the excellent ability of Inner-MPDIoU in bounding box positioning, category classification, and feature learning.



(**a**) box_loss (**b**) cls_loss (**c**) dfl_loss

Figure 11. Contrast experiments of different loss functions in loss values.

3.7. Comparative Experiments of Different Models

To thoroughly showcase the enhanced algorithm's efficacy in detecting chili Phytophthora blight, this paper, conducts a comparative analysis with prevalent object detection algorithms including Faster R-CNN, SSD [38], YOLOv5n, YOLOv6n, YOLOv8n, YOLOV8s, and YOLOv8m, utilizing the dataset compiled for this study. The outcomes of this comparison are tabulated in Table 10.

Table 10. The comparison test results of different models.

Model	P (%)	R (%)	mAP0.5 (%)	Params (M)	FPS
Faster R-CNN	0.931	0.823	0.879	72.0	23
SSD	0.912	0.838	0.881	41.1	52.14
YOLOv5n	0.906	0.812	0.862	1.76	254.6
YOLOv6n	0.898	0.798	0.853	4.23	277
YOLOv8n	0.905	0.824	0.872	3.15	232.8
YOLOv8s	0.921	0.847	0.890	11.2	160
YOLOv8m	0.950	0.826	0.892	25.9	81
YOLOv8-GDCI	0.914	0.842	0.889	3.27	219.5

Two-stage object detection methods, exemplified by Faster R-CNN and SSD, often struggle to achieve ideal real-time performance in complex or rapidly changing detection tasks due to their high computational complexity and large model size. Despite its speed, YOLOv5n, with the lowest number of parameters, has a limitation in chili Phytophthora blight disease detection, as its mAP0.5 is only 0.862. Similarly, YOLOv6n performs poorly in this task, achieving a mAP0.5 of only 0.853, with both precision and recall at relatively low levels. While striving for extreme lightweightness, it sacrifices some detection performance. Although YOLOv8n can balance accuracy and computational cost, there remains ample scope for enhancing its detection precision. On the other hand, while YOLOv8s and YOLOv8m have shown improved detection accuracy, their accompanying higher number of parameters and increased computational cost limit their applicability in real-time chili Phytophthora blight disease detection scenarios. In comparison with the aforementioned models, the model presented in this paper has undergone various improvements. Considering various indicators such as detection accuracy, missed detection rate, mAP0.5, number of parameters, and FPS, our model achieves the best results. Specifically, this model performs outstandingly in terms of accuracy, reaching a high level of 0.914. This value is second only to those models specifically designed for high accuracy rather than real-time performance, such as Faster R-CNN, YOLOv8s, and YOLOv8m. The recall is second only to YOLOv8s, which is 0.842, effectively reducing missed detections and further enhancing the comprehensiveness and reliability of detection. The mAP0.5 score of 0.889, second only to YOLOv8s and YOLOv8m, demonstrates that its object detection capability in complex scenarios is also formidable, enabling it to accurately identify and locate multiple targets. In spite of the remarkable gains in performance efficiency, this model boasts a comparably modest number of parameters, marginally surpassing those of lightweight models like

YOLOv5n and YOLOv8n, thereby enabling it to function proficiently even in settings with limited resources. At the same time, although its FPS index is slightly lower than a few models such as YOLOv5n, YOLOv6n, and YOLOv8n, it consistently sustains a notably high frame rate, guaranteeing real-time detection capabilities and fulfilling the necessity for prompt monitoring in smart farming environments.

3.8. Visualization

To illustrate the comparative performance of this algorithm in a more intuitive manner, this paper selects image samples covering multiple dimensions for comparative experiments, specifically including leaf scenes, chili fruit scenes, and stem bifurcation scenes under simple and complex disease conditions. In a simple scene, the detection targets are clearly visible, and unobstructed by other parts, and their boundaries are easy to identify, making it easier to accurately locate and assess the severity of the disease. In a complex scene, the detection targets may overlap with each other, making it difficult to instantly recognize clear boundaries, and the targets are more densely packed. The comparison of model performance is based on the accuracy of detecting chili Phytophthora blight. Figure 12 illustrates the results.

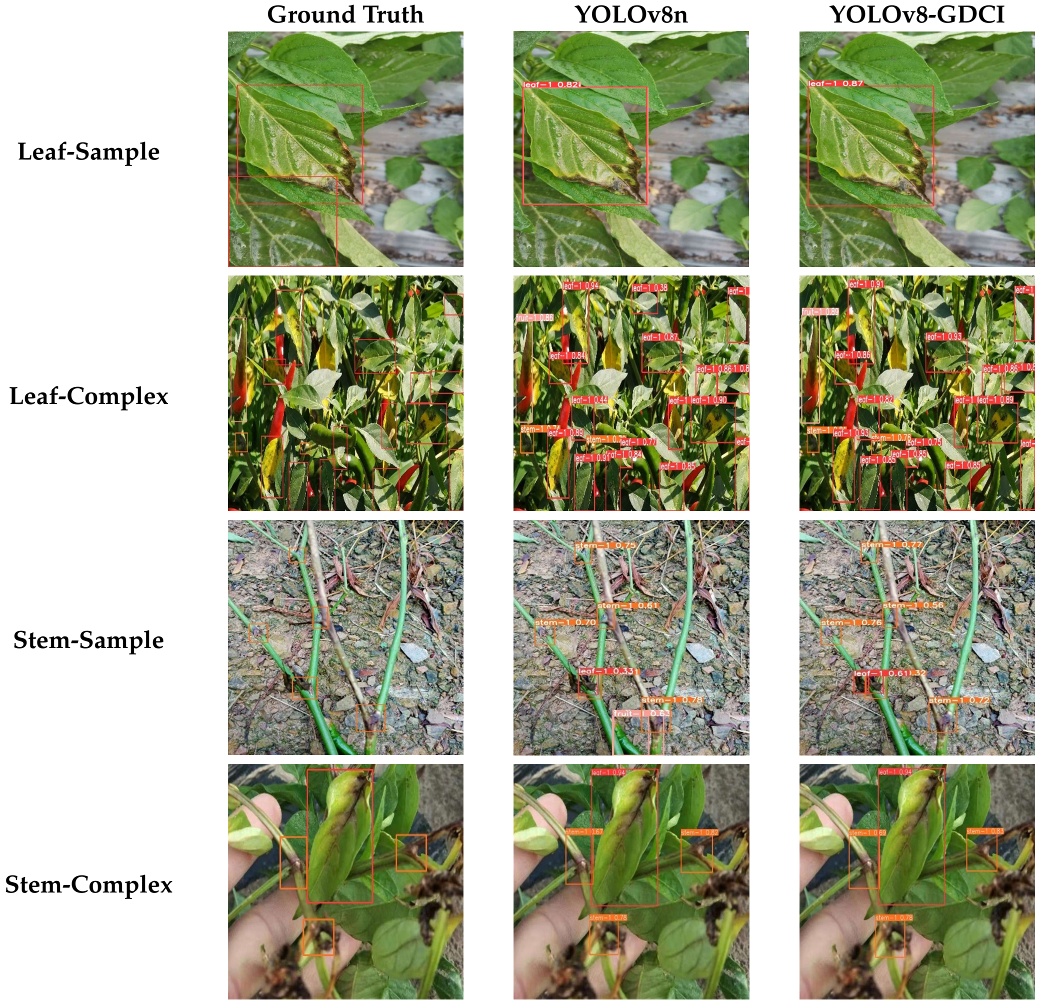


Figure 12. *Cont.*

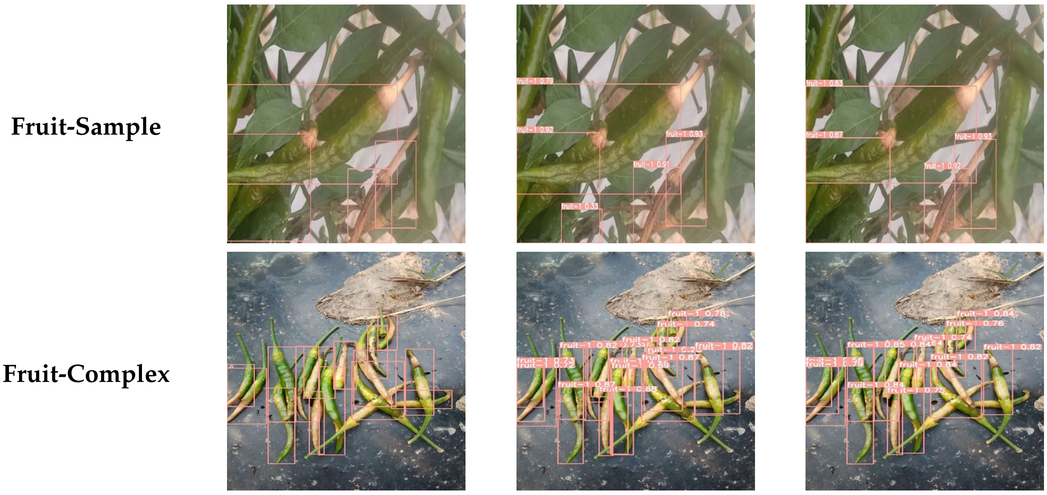


Figure 12. The visualization results in different environments.

When dealing with simple disease scenes, the YOLOv8n model performs well, but its prediction confidence is slightly inferior to that of YOLOv8-GDCI. However, when facing more complex disease scenarios, such as challenges like complex background information and dense target distribution, the YOLOv8n model exposes more obvious limitations; specifically an increase in missed detections and false detections. Missed detection refers to the situation where the model mistakenly identifies a diseased area as healthy. False detection refers to the situation where the model erroneously identifies a healthy area as diseased.

In comparison, the refined algorithm, YOLOv8-GDCI, presented in this study, demonstrates marked improvements in tackling intricate and challenging situations. It can not only accurately identify and locate the target in a complex background, but also maintain a high accuracy rate when facing dense targets, while effectively minimizing the rates of missed detections and spurious detections.

4. Discussion

The present study introduces a novel object detection algorithm, YOLOv8-GDCI, specifically tailored for identifying chili Phytophthora blight. The YOLOv8-GDCI algorithm can accurately identify the different characteristics of Phytophthora capsici on leaves, fruits, and stems: on leaves, it often appears as circular or irregular water-soaked spots; on fruits, it manifests as irregular water-soaked patches and on stems, it presents as dark, moist, irregular plaques. Based on the recognition of these characteristics, YOLOv8-GDCI can efficiently detect chili Phytophthora blight. In comparison to existing algorithms such as YOLOv5n, YOLOv6n, and YOLOv8n, this approach proficiently addresses challenges posed by dense target clusters and complex, cluttered surroundings, while adhering to the necessity for real-time detection. The mAP of YOLOv8-GDCI reached 0.889, which is 1.7% higher than the base model YOLOv8n, 2.7% higher than YOLOv5n, and 3.6% higher than YOLOv6n. The precision is 0.914, 0.9% higher than YOLOv8n, 0.8% higher than YOLOv5n, and 1.6% higher than YOLOv6n. The recall rate is 0.842, which is 1.8% higher than YOLOv8n, 3% higher than YOLOv5n, and 4.4% higher than YOLOv6n. Meanwhile, the FPS of YOLOv8-GDCI exceeds 60, meeting the requirements for real-time scenarios. Its core contributions encompass the following aspects:

- Addressing the potential limitation of inadequate multi-scale feature fusion within the neck network of the conventional YOLOv8n, this paper presents the RepGFPN network architecture as a solution. This improvement, while efficiently managing the computational overhead, fosters the interplay and integration of information across feature representations of varying scales, markedly amplifying the model's proficiency in identifying multi-scale objects within intricate settings, thereby bolstering its detection performance.

- The Dysample upsampling operator is used to substitute the original upsampling operator, enhancing the model's receptive field. This operator constructs the upsampling process through a unique point sampling strategy, effectively enlarging the feature receptive field of the model; thereby achieving more delicate and accurate feature reconstruction.
- We integrated the CA attention mechanism into the final stage of the backbone network, which cleverly combines channel information with spatial position information, achieving more comprehensive and accurate target detection.
- Aimed at the issue of the inadequate performance of the CIoU loss function in small object detection tasks, this paper proposes to adopt the Inner-MPDIoU loss function as an alternative. It addresses the issue of limited accuracy in detecting small targets and, to a certain extent, improves the overall detection efficiency and performance.

Despite demonstrating certain potential in detecting chili Phytophthora blight, the YOLOv8-GDCI model still faces significant challenges. In practical application scenarios, the detection of chili Phytophthora blight lesions is often influenced by various factors, with shading being a notable issue that cannot be ignored. Shading can obscure or alter the characteristics of lesions, leading to misjudgments by the model and consequently reducing detection accuracy. Although the improved model has made progress in addressing the occlusion issue, there is still room for further improvement. Furthermore, the current YOLOv8-GDCI model may lack the capability to detect chili plants at night, which limits the comprehensiveness and practicality of the model to a certain extent. To further enhance the practicality of this model, we can collect nighttime images of chili Phytophthora disease for training, thereby improving the model's ability to detect chili plants at night. Additionally, considering the similarity in symptoms between chili wilt and chili Phytophthora blight, we can attempt to apply the YOLOv8-GDCI model to the detection of chili wilt. By comparing and analyzing the symptomatic features of these two diseases, we can utilize the model to differentiate between them, achieving precise identification of chili diseases. In the future, we will continue to optimize the network architecture, considering enhancements to the backbone network and convolutions to improve feature extraction capabilities. Furthermore, we will continue to explore new attention mechanisms to further enhance the application effectiveness of the model in smart farms.

5. Conclusions

This study presents a novel object detection algorithm, YOLOv8-GDCI, specifically designed for detecting chili Phytophthora blight. The algorithm can accurately identify the various disease characteristics of Phytophthora capsici on chili leaves, fruits, and stems, and can efficiently handle complex and dense target scenarios. Compared to the YOLOv8n algorithm, YOLOv8-GDCI demonstrates superior performance across multiple metrics. Specifically, the mAP of YOLOv8-GDCI reaches 0.889, which is 1.7% higher than YOLOv8n; its precision is 0.914, which is 0.9% higher than YOLOv8n and its recall rate is 0.842, which is 1.8% higher than YOLOv8n. Additionally, the FPS of YOLOv8-GDCI exceeds 60, meeting the requirements for real-time detection. These improvements result in a model that exhibits strong accuracy and efficiency in detecting chili Phytophthora blight. The main innovations of this algorithm are multifaceted. First, the RepGFPN network architecture is proposed to address the insufficient multi-scale feature fusion in the neck network of the traditional YOLOv8n model. This improvement effectively manages computational overhead while promoting the interaction and integration of information across feature representations at different scales, significantly enhancing the model's ability to identify targets in complex environments. Second, the Dysample upsampling operator is introduced to replace the original upsampling method, which strengthens the model's receptive field, leading to finer and more accurate feature reconstruction. Third, the integrated CA attention mechanism cleverly combines channel and spatial position information, improving the comprehensiveness and accuracy of target detection. Lastly, to address the limitations of the CIoU loss function in small object detection, the Inner-MPDIoU loss function is proposed, which significantly improves the accuracy of detecting small targets. YOLOv8-GDCI performs excellently in detecting chili Phytophthora blight, but still faces some challenges, such as occlusion issues, where progress has been made but further improvements are needed, and limited night-time detection capabilities. Future research will focus on collecting night-time images and expanding the dataset to enhance detection capabilities, as well as exploring its application in detecting chili wilt. Additionally, efforts will be made to optimize the network architecture, improve feature extraction capabilities, and explore new attention mechanisms to further enhance the model's effectiveness in smart farming applications.

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

Funding: This research was funded by Xinjiang Smart Farm Project—National Science and Technology Major Special Project for the New Generation of Artificial Intelligence (2022ZD0115800).

Data Availability Statement: The datasets in this study are available from the corresponding author on reasonable request.

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

References

1. Zou, Z.; Zou, X. Geographical and ecological differences in pepper cultivation and consumption in China. *Front. Nutr.* **2021**, *8*, 718517. [CrossRef] [PubMed]
2. Idoje, G.; Dagiuklas, T.; Iqbal, M. Survey for smart farming technologies: Challenges and issues. *Comput. Electr. Eng.* **2021**, *92*, 107104. [CrossRef]
3. Ozyilmaz, U. Evaluation of the effectiveness of antagonistic bacteria against Phytophthora blight disease in pepper with artificial intelligence. *Biol. Control* **2020**, *151*, 104379. [CrossRef]
4. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
5. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.
6. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
7. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
8. Jocher, G.; Chaurasia, A.; Stoken, A.; Borovec, J.; Kwon, Y.; Michael, K.; Fang, J.; Wong, C.; Yifu, Z.; Montes, D.; et al. Ultralytics/yolov5: V6. 2-yolov5 classification models, apple m1, reproducibility, clearml and deci. ai integrations. *Zenodo* **2022**. Available online: https://zenodo.org/records/7002879 (accessed on 12 July 2023).
9. 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.
10. 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/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 7464–7475.
11. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef]
12. Zhao, J.; Qu, J. Healthy and Diseased Tomatoes Detection Based on YOLOv2. In *Proceedings of the Human Centered Computing*; Tang, Y., Zu, Q., Rodríguez García, J.G., Eds.; Springer: Cham, Switzerland, 2019; pp. 347–353.
13. Liu, J.; Wang, X. Early recognition of tomato gray leaf spot disease based on MobileNetv2-YOLOv3 model. *Plant Methods* **2020**, *16*, 83. [CrossRef] [PubMed]
14. Sangaiah, A.K.; Yu, F.N.; Lin, Y.B.; Shen, W.C.; Sharma, A. UAV T-YOLO-Rice: An Enhanced Tiny Yolo Networks for Rice Leaves Diseases Detection in Paddy Agronomy. *IEEE Trans. Netw. Sci. Eng.* **2024**, 1–16. [CrossRef]
15. Xie, Z.; Li, C.; Yang, Z.; Zhang, Z.; Jiang, J.; Guo, H. YOLOv5s-BiPCNeXt, a Lightweight Model for Detecting Disease in Eggplant Leaves. *Plants* **2024**, *13*, 2303. [CrossRef]
16. Yue, X.; Li, H.; Song, Q.; Zeng, F.; Zheng, J.; Ding, Z.; Kang, G.; Cai, Y.; Lin, Y.; Xu, X.; et al. YOLOv7-GCA: A Lightweight and High-Performance Model for Pepper Disease Detection. *Agronomy* **2024**, *14*, 618. [CrossRef]

17. Yang, S.; Yao, J.; Teng, G. Corn Leaf Spot Disease Recognition Based on Improved YOLOv8. *Agriculture* **2024**, *14*, 666. [CrossRef]
18. Wang, G.; Chen, Y.; An, P.; Hong, H.; Hu, J.; Huang, T. UAV-YOLOv8: A Small-Object-Detection Model Based on Improved YOLOv8 for UAV Aerial Photography Scenarios. *Sensors* **2023**, *23*, 7190. [CrossRef] [PubMed]
19. 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; pp. 8759–8768.
20. Jiang, Y.; Tan, Z.; Wang, J.; Sun, X.; Lin, M.; Li, H. Giraffedet: A heavy-neck paradigm for object detection. In Proceedings of the International Conference on Learning Representations, Vienna, Austria, 4 May 2021.
21. Xu, X.; Jiang, Y.; Chen, W.; Huang, Y.; Zhang, Y.; Sun, X. Damo-yolo: A report on real-time object detection design. *arXiv* **2022**, arXiv:2211.15444.
22. Soudy, M.; Afify, Y.; Badr, N. RepConv: A novel architecture for image scene classification on Intel scenes dataset. *Int. J. Intell. Comput. Inf. Sci.* **2022**, *22*, 63–73. [CrossRef]
23. Cunningham, P.; Delany, S.J. K-nearest neighbour classifiers-a tutorial. *ACM Comput. Surv. (CSUR)* **2021**, *54*, 1–25. [CrossRef]
24. Wang, J.; Chen, K.; Xu, R.; Liu, Z.; Loy, C.C.; Lin, D. Carafe: Content-aware reassembly of features. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 3007–3016.
25. Lu, H.; Liu, W.; Fu, H.; Cao, Z. FADE: Fusing the assets of decoder and encoder for task-agnostic upsampling. In Proceedings of the European Conference on Computer Vision, Tel Aviv, Israel, 23–27 October 2022; Springer: Berlin/Heidelberg, Germany, 2022; pp. 231–247.
26. Lu, H.; Liu, W.; Fu, H.; Cao, Z. FADE: A Task-Agnostic Upsampling Operator for Encoder–Decoder Architectures. *Int. J. Comput. Vis.* **2024**, 1–22. [CrossRef]
27. Liu, W.; Lu, H.; Fu, H.; Cao, Z. Learning to upsample by learning to sample. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Paris, France, 2–6 October 2023; pp. 6027–6037.
28. Niu, Z.; Zhong, G.; Yu, H. A review on the attention mechanism of deep learning. *Neurocomputing* **2021**, *452*, 48–62. [CrossRef]
29. 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; pp. 7132–7141.
30. 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.
31. Hou, Q.; Zhou, D.; Feng, J. Coordinate attention for efficient mobile network design. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 13713–13722.
32. Xiong, C.; Zayed, T.; Jiang, X.; Alfalah, G.; Abelkader, E.M. A Novel Model for Instance Segmentation and Quantification of Bridge Surface Cracks—The YOLOv8-AFPN-MPD-IoU. *Sensors* **2024**, *24*, 4288. [CrossRef]
33. Siliang, M.; Yong, X. MPDIoU: A loss for efficient and accurate bounding box regression. *arXiv* **2023**, arXiv:2307.07662.
34. Zhang, H.; Xu, C.; Zhang, S. Inner-IoU: More effective intersection over union loss with auxiliary bounding box. *arXiv* **2023**, arXiv:2311.02877.
35. Chen, J.; Mai, H.; Luo, L.; Chen, X.; Wu, K. Effective feature fusion network in BIFPN for small object detection. In Proceedings of the 2021 IEEE International Conference on Image Processing (ICIP), Anchorage, AK, USA, 19–22 September 2021; pp. 699–703.
36. Ouyang, D.; He, S.; Zhang, G.; Luo, M.; Guo, H.; Zhan, J.; Huang, Z. Efficient multi-scale attention module with cross-spatial learning. In Proceedings of the ICASSP 2023—2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Rhodes Island, Greece, 4–10 June 2023; pp. 1–5.
37. Yang, L.; Zhang, R.Y.; Li, L.; Xie, X. Simam: A simple, parameter-free attention module for convolutional neural networks. In Proceedings of the International Conference on Machine learning, Online, 18–24 July 2021; pp. 11863–11874.
38. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Proceedings, Part I 14; Springer: Berlin/Heidelberg, Germany, 2016; pp. 21–37.

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