

Article

Lightweight Network for Corn Leaf Disease Identification Based on Improved YOLO v8s

Rujia Li ¹, Yadong Li ¹, Weibo Qin ² , Arzlan Abbas ² , Shuang Li ¹, Rongbiao Ji ¹, Yehui Wu ¹, Yiting He ¹ and Jianping Yang ^{1,*} 

¹ School of Big Data, Yunnan Agricultural University, Kunming 650201, China; 18988097525@163.com (R.L.); 14787825720@163.com (Y.L.); 15200724120@163.com (S.L.); jirb4532@163.com (R.J.); wuyehui0511@163.com (Y.W.); hytlrj123123@163.com (Y.H.)

² College of Plant Protection, Jilin Agricultural University, Changchun 130118, China; qinweibo@mails.jlau.edu.cn (W.Q.); zain514786@gmail.com (A.A.)

* Correspondence: yangjp@ynau.edu.cn

Abstract: This research tackles the intricate challenges of detecting densely distributed maize leaf diseases and the constraints inherent in YOLO-based detection algorithms. It introduces the GhostNet_Triplet_YOLOv8s algorithm, enhancing YOLO v8s by integrating the lightweight GhostNet (Ghost Convolutional Neural Network) structure, which replaces the YOLO v8s backbone. This adaptation involves swapping the head's C2f (Coarse-to-Fine) and Conv (Convolutional) modules with C3 Ghost and GhostNet, simplifying the model architecture while significantly amplifying detection speed. Additionally, a lightweight attention mechanism, Triplet Attention, is incorporated to refine the accuracy in identifying the post-neck layer output and to precisely define features within disease-affected areas. By introducing the ECIOU_Loss (EfficientCIoU Loss) function, replacing the original CIoU_Loss, the algorithm effectively mitigates issues associated with aspect ratio penalties, resulting in marked improvements in recognition and convergence rates. The experimental outcomes display promising metrics with a precision rate of 87.50%, a recall rate of 87.70%, and an mAP@0.5 of 91.40% all within a compact model size of 11.20 MB. In comparison to YOLO v8s, this approach achieves a 0.3% increase in mean average precision (mAP), reduces the model size by 50.2%, and significantly decreases FLOPs by 43.1%, ensuring swift and accurate maize disease identification while optimizing memory usage. Furthermore, the practical deployment of the trained model on a WeChat developer mini-program underscores its practical utility, enabling real-time disease detection in maize fields to aid in timely agricultural decision-making and disease prevention strategies.

Keywords: corn leaf disease; ECIOU_Loss; GhostNet networks; Triplet Attention; WeChatlets; YOLO v8s



Citation: Li, R.; Li, Y.; Qin, W.; Abbas, A.; Li, S.; Ji, R.; Wu, Y.; He, Y.; Yang, J. Lightweight Network for Corn Leaf Disease Identification Based on Improved YOLO v8s. *Agriculture* **2024**, *14*, 220. <https://doi.org/10.3390/agriculture14020220>

Academic Editors: Taha Mansouri and Ali Alameer

Received: 27 December 2023

Revised: 19 January 2024

Accepted: 27 January 2024

Published: 29 January 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Corn is China's largest food crop, and the healthy development of the corn seed industry is of great economic significance to ensure national food security and an effective supply of agricultural products. Emphasizing the need for intensified research into corn origin, self-sufficient exploration of core breeding techniques, innovation in superior germplasm, and the cultivation of novel varieties, there is a push to integrate and propel high-quality advancements in the corn seed industry. However, despite this focus, the expanding cultivation zones have witnessed an increased in corn diseases, posing a substantial threat to both quantity and quality of corn [1–3]. Consequently, the swift and precise identification of plant diseases and pests becomes paramount in enhancing agricultural productivity. Conventional methods relying on human expertise to diagnose plant anomalies arising from diseases, pests, nutrient deficiencies, or extreme weather are expensive, time-intensive, and impractical. Leveraging the rapid evolution of computer vision and deep learning, there has been a growing spotlight on disease and pest identification methodologies based on image

processing and machine learning. Specifically, the integration of target detection technology within the agricultural domain has aroused substantial attention among researchers.

The growth patterns and visual attributes of corn leaves exhibit significant variation across different developmental stages, which are influenced by factors like illumination, angle, occlusion, etc., leading to intricate and dynamic characteristics in areas affected by pests and diseases. This complexity renders target detection notably challenging. Moreover, there are many types of corn leaf diseases, and insect pests often share morphological similarities, increasing the risk of misidentification. At present, extensive research by both domestic and international scholars has explored plant disease target identification methods. For instance, Yang et al. introduced Maize-YOLO, a high-precision, real-time corn pest detection method, achieving a 76.3% mAP and 77.3% recall rate [4]. Li et al. also proposed a convolutional neural network (CNN) with multi-scale feature fusion for corn leaf disease detection, demonstrating its effectiveness [5]. Zhang et al. presented SwinT-YOLO as a new detection model with comparable parameters and FLOPs to YOLOv4, achieving a 95.11% detection accuracy in their corn tassel remote sensing dataset [6]. Additionally, the YOLO-Tea model showcases better performance compared to Faster R-CNN and SSD in terms of AP@0.5, AP@TLB, and AP@GMB by margins of 5.5%, 1.8%, 7.0%, and 7.7%, 7.8%, 5.2%, respectively [7]. Furthermore, Yang et al. proposed the Maize-YOLO model, achieving a 76.3% mAP and 77.3% recall rate in corn pest detection. Sun et al. incorporated an attention mechanism into a convolutional neural network for identifying soybean aphids [8], achieving promising results.

Currently, research on identifying corn leaf diseases faces limitations due to the under-performance of traditional target detection algorithms [9–11] and an increased challenge in detection owing to the evolving regional characteristics of corn leaf diseases [12]. This study aims to introduce a novel method for corn disease detection using an enhanced YOLO v8s algorithm. This algorithm is based on YOLO v8s, which integrates the lightweight GhostNet [13] network structure to replace the backbone network of YOLO v8s. Notably, it replaces all C2f and Conv modules in the head with C3Ghost and GhostNet modules, significantly reducing model degree complexity while enhancing detection capabilities. Furthermore, it incorporates a lightweight attention mechanism (Triplet Attention) [14] post-neck layer to amplify the detection efficacy of the output after the neck layer, thereby improving the model's recognition rates. Finally, it adopts the ECIoU_Loss [15] loss function, replacing the original CIoU_Loss to rectify issues related to aspect ratio changes in CIoU, consequently enhancing network convergence rate. This refined and lightweight model is deployed on the WeChat developer applet for practical implementation in monitoring leaf diseases on actual corn planting fields.

2. Materials and Methods

2.1. Dataset

There are various corn leaf diseases that cause serious damage. This study selected four publicly available corn disease datasets from Kaggle, with a total of 1921 images, as shown in Figure 1.

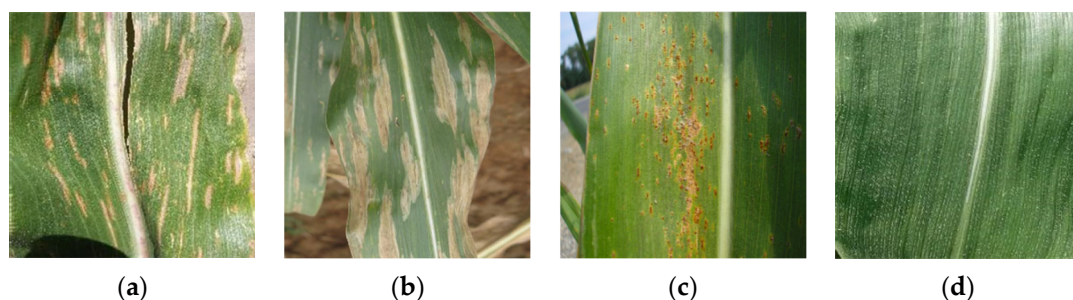


Figure 1. Original dataset. (a) *Cercospora* leaf spot, (b) northern leaf blight, (c) rust, (d) health.

To enhance the adaptability of our model for recognizing corn leaf diseases and diversify the dataset, we implemented a range of image augmentation techniques. These techniques encompassed random cropping, blurring, brightness adjustment, noise addition, and flipping; processing images were taken from various angles and lighting conditions, as illustrated in Figure 2. These methods not only heightened visual diversity but also replicated a spectrum of environmental changes encountered in real-world scenarios, including fluctuations in lighting, diverse capture angles, and occlusions. For example, rotation and scaling introduced variability in viewpoint and size, color adjustments assisted the model in acclimating to different lighting and color conditions, and the introduction of occlusions and noise simulated partial information loss and common image quality issues in real-world environments.

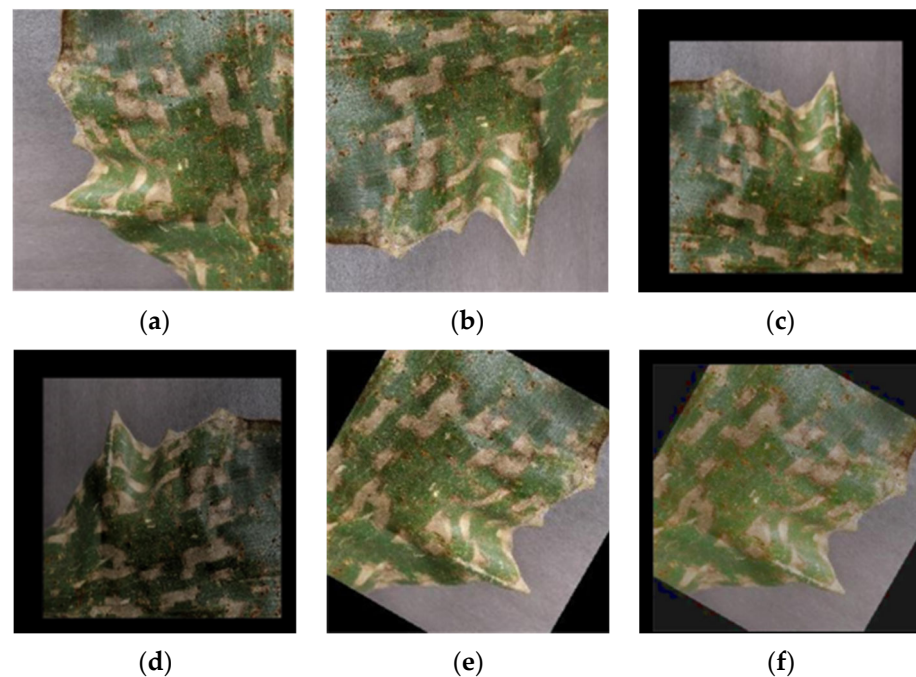


Figure 2. Augmented images: (a) original image, (b) 50% random flip, (c) scaling and flip, (d) random brightness alteration, (e) image scaling and 30° rotation, (f) addition of Gaussian noise.

These augmentation strategies not only enriched the dataset, expanding the number of images from 1407 to 5763, as shown in Table 1, but also effectively mitigated the risk of overfitting, thereby enhancing the model's generalizability. The augmented training image dataset comprised 1407 healthy corn leaf images, 1419 images depicting *Cercospora* leaf spot, 1477 showing northern leaf blight, and 1460 displaying rust. In our experiments, the dataset was partitioned into a training set and a test set in an 8:2 ratio, with all training images standardized to 640×640 pixels. The experimental results, presented in Table 2, demonstrated noteworthy improvements in accuracy, recall, and other performance metrics for the model trained on the augmented dataset, confirming the efficacy of image augmentation techniques in enhancing model performance.

Table 1. Comparison of the dataset before and after augmentation.

Dataset	Number of Pictures/Piece	<i>Cercospora</i> Leaf Spot	Northern Leaf Blight	Rust	Health
Original dataset	1921	448	463	458	552
After data augmentation	5763	1419	1477	1460	1407

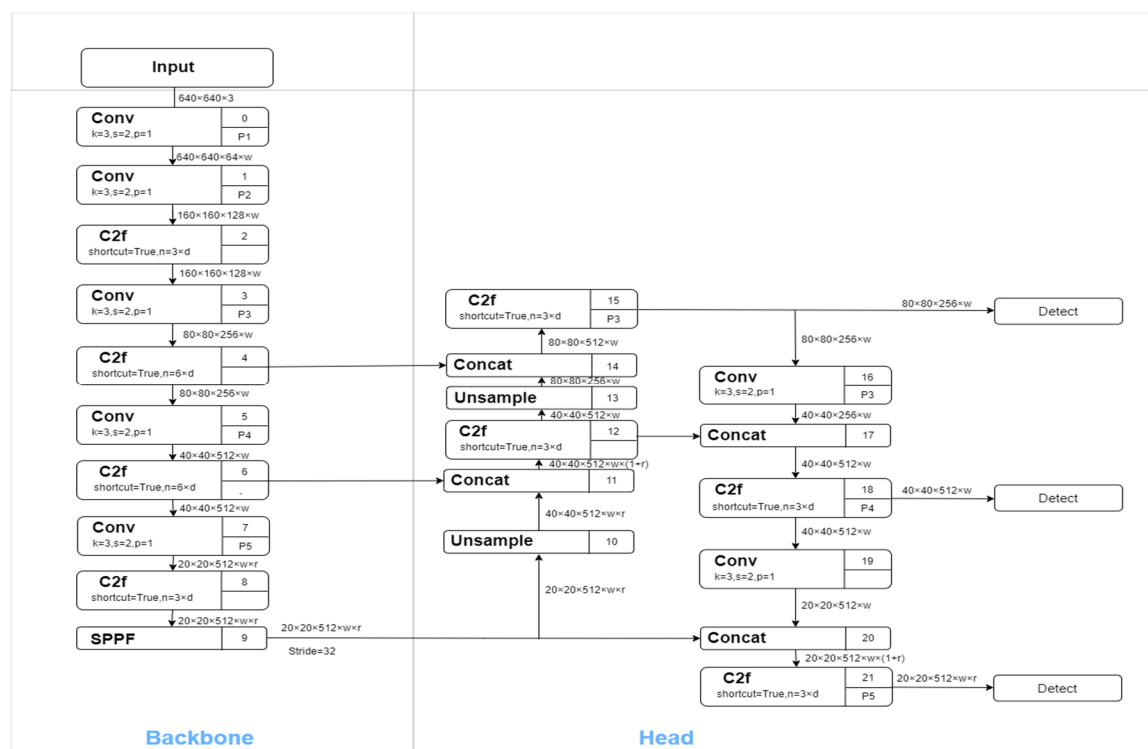
Table 2. Comparison of YOLO v8s model before and after image augmentation processing.

Data Set	P/%	R/%	mAP@0.5%
Original dataset	84.5	88.2	90.8
After data augmentation	86.7	89.3	91.1

In summary, through the application of diverse image augmentation techniques, we not only expanded the training dataset but also enhanced the model's adaptability and robustness to real-world conditions. This holds significant importance for the practical application of corn leaf disease recognition.

2.2. YOLO v8 Network Structures

YOLO v8 is the latest series of YOLO based on the series for target detection models introduced by Ultralytics, and it stands as the cutting-edge state-of-the-art (SOTA) model in this domain [16]. This target detection model is structured into four key components: input, backbone, neck, and prediction, depicted in Figure 3.

**Figure 3.** YOLO v8s network architecture.

Input: scales image size data to suit model training requirements.

Backbone: utilizes the CSPDarkNet-53 network as its primary framework. Through an augmented number of convolutional layers, it extracts feature maps (P1–P5) with varying receptive fields, progressively expanding these fields in sequence.

Neck: introduces FPN (Feature Pyramid Network) and PAN (Path Aggregation Network) structures. FPN employs multiple scales for detecting targets of diverse sizes, while PAN acts as a bottom–up feature pyramid. This combined operation allows FPN to convey potent semantic features from top to bottom, while the feature pyramid handles robust positional features from bottom to top. Together, they aggregate features from distinct backbone layers to different detection layers.

Prediction: constitutes the model's head, which is responsible for the final prediction output. As the receptive field grows from P3 > P4 > P5, the predicted targets transition from small > medium > large.

2.3. Backbone Network Optimization

The widely used lightweight networks like ShuffleNet [17], MobileNet [18,19], and GhostNet are prominent in current applications. Among these, MobileNetv3, a notable iteration, combines different convolution techniques and references a lightweight attention mechanism called SE (Squeeze and Excitation), enhancing detection speed without significant accuracy loss. GhostNet, another impactful network, has demonstrated approximately 1% improved accuracy compared to MobileNetv3 while maintaining similar computational complexity in ImageNet tasks. Thus, this article adopts GhostNet as the replacement backbone extraction network for YOLO v8, aiming to reduce computational demands in deep neural networks. The Ghost module in GhostNet partitions the standard convolution layer into two parts (one part is used for standard convolution while the other part performs a series of simple linear operations on the inherent feature map generated by the first part to generate the final output of the feature map), combining linear transformation with standard convolution to reduce model parameters and calculation amounts. This module generates a complete output feature map with reduced computational cost while maintaining its original size. Illustrations of the standard convolution and Ghost module can be seen in Figure 4.

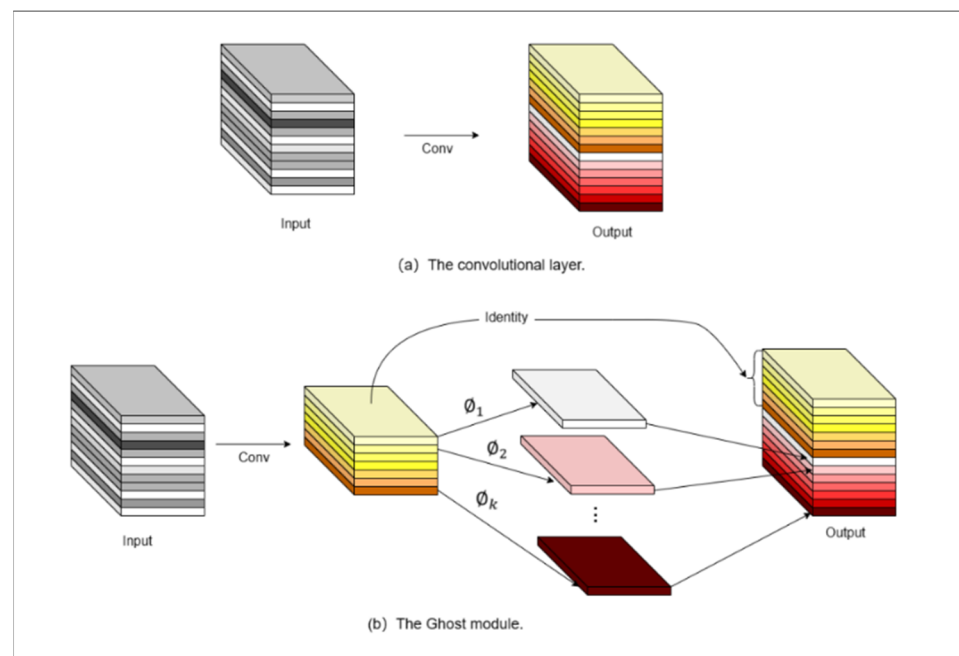


Figure 4. Ghost module.

For standard convolution operations, given an input feature $X \in R^{h \times w \times c}$, where h is the height of the input feature map, w is the width of the feature map, and c is the number of input channels, the operation of any convolution layer that generates n feature maps is as follows (1):

$$Y = X * f + b \quad (1)$$

Among them, $f \in R^{c \times k \times k \times n}$ is the convolution kernel, b is the bias term, and $Y \in R^{h' \times w' \times n}$ is the feature map of the output channel. The calculation formula FLOPs in re-convolution is depicted in Formula (2):

$$FLOPs = c \times k \times k \times n \times h' \times w' \quad (2)$$

Among them, h' and w' are the height and width of the output feature map respectively, and k is the size of the convolution kernel f .

The Ghost module performs a convolution operation on part of the feature maps, that is, using a standard convolution on m original output feature maps $Y' \in R^{h' \times w' \times m}$, where $m \leq n$; the operation is as shown in Formula (3):

$$Y' = X * f' \quad (3)$$

where $f' \in R^{c \times k \times k \times m}$ is the convolution kernel used by the feature layer and does not include the bias term. In order to further obtain the required n feature maps, a series of simple linear changes are performed on the obtained m -dimensional feature map, and s similar feature maps have been generated. The operation is as shown in Equation (4):

$$y_{ij} = \phi_{i,j}(y'_i) \quad \forall i = 1, \dots, m, j = 1, \dots, s \quad (4)$$

where y'_i is the i -th original feature map in Y' , and $\phi_{i,j}$ is the j -th linear calculation used to generate the j -th similar feature map y_{ij} . The calculation amount *FLOPs* are given in formula mentioned below (5):

$$FLOPs = c \times k \times k \times \frac{n}{s} \times h' \times w' + \frac{n}{s} \times d \times d \times (s - 1) \times h' \times w' \quad (5)$$

where d is the average kernel size per linear operation. It can be seen from Formula (5) that the Ghost module divides the calculation into two parts: one part is an ordinary convolution operation, and the other part is a linear transformation operation. Combined with Formula (2), the compression ratio of the model is about s , which greatly reduces the number of parameters.

To build GhostNet, the Ghost Bottleneck module is designed, which is similar to the basic residual block in ResNet and is stacked by two Ghost modules, as shown in Figure 5.

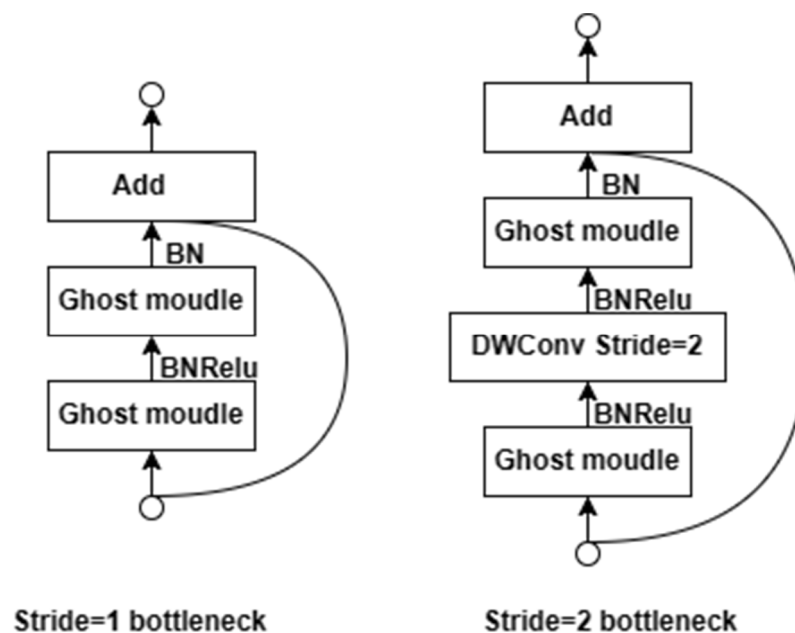


Figure 5. Ghost bottleneck module.

2.4. Triplet Attention

In recent years, the attention mechanism has received extensive attention, particularly within various computer vision domains. While common attention mechanisms enhance model performance, increasing network depth typically intensifies computational requirements. Generally, the amount of calculation that lightweight networks can withstand is limited, so the conventional attention mechanism added to the recognition of lightweight network applications will be affected.

In order to improve model accuracy and trim model size, this article introduces a lightweight and effective attention mechanism—Triplet Attention, which is a new method of calculating attention weights by using a three-branch structure to capture cross-dimensional interactions. Through rotation operations and residual transformations, Triplet Attention establishes inter-dimensional dependencies for input tensors, encoding inter-channel and spatial information with minimal computational overhead. This approach is simple and efficient, and it seamlessly integrates as an additional module within classic backbone networks and the neck output.

Unlike CBAM and SENet, which necessitate learnable parameters for channel dependencies, Triplet Attention models channel spatial attention via an almost parameterless mechanism, as depicted in Figure 6.

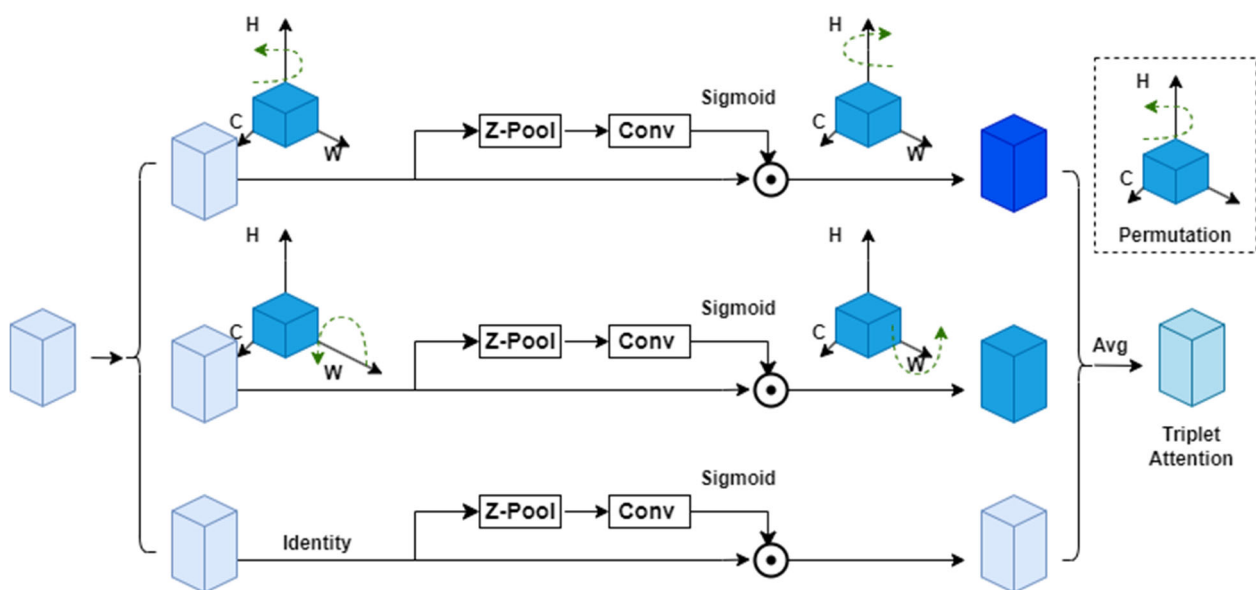


Figure 6. Schematic diagram of Triplet Attention.

The schematic depiction of Triplet Attention reveals a structure with three branches (Figure 6). The top and middle branches compute attention weights between the channel dimension (C) and spatial dimensions, while the bottom branch captures spatial dependencies (H and W). In the first two branches, rotation operations establish a linkage between the channel dimension and one of the spatial dimensions. The resulting weights are ultimately aggregated through straightforward averaging.

2.5. *EfficiC*Loss Loss Function

In YOLO v8s, the application of the CIoU loss function introduces limitations during regression when predicting simultaneous changes in frame width and height. This is attributed to its inability to effectively optimize similarity, particularly when handling variations in aspect ratios between predicted and actual boxes. To overcome this challenge, the EIoU_Loss, an improved variant based on the CIoU_Loss (Complete Intersection over Union Loss) concept, seeks to precisely calculate length–width differences by isolating the aspect ratio factor. This refined approach aids in better understanding the similarity between bounding boxes, thereby optimizing their position and size more effectively.

Expanding on this improvement, the ECIoU_Loss is introduced as a further advancement. This enhanced loss function integrates components from CIoU and EIoU in its computation. Initially, CIoU adjusts the aspect ratio of the predicted box until it falls within an appropriate range, which is followed by EIoU fine-tuning each edge until it converges

to the correct value. Equation (6) illustrates this refined calculation, enhancing the accuracy of bounding box positions and sizes.

$$ECIoU_Loss = 1 - IOU + \alpha v + \frac{\rho^2(b^{gt}, b)}{c^2} + \frac{\rho^2(h^{gt}, h)}{c_h^2} + \frac{\rho^2(w^{gt}, w)}{c_w^2} \quad (6)$$

In the Formula (6), IoU —traditional regression loss
Definition of penalty term:

$$\mathfrak{R}_{CIoU} = \frac{\rho^2(b^{gt}, b)}{c^2} + \alpha v \quad (7)$$

$$\alpha = \frac{v}{(1 - IoU) + v} \quad (8)$$

$$v = \left(e^{\frac{w^{gt}}{h^{gt}}} - e^{\frac{w}{h}} \right)^2 \quad (9)$$

As shown in the formula above, ρ^2 represents the square of the Euclidean distance between two rectangular boxes. c^2 represents the square of the diagonal distance between two rectangular boxes. b and b^{gt} represent the center point of two rectangular boxes. α represents the weight coefficient. v is used to measure the consistency of the relative proportions of two rectangular frames.

For w^{gt} , h^{gt} , w and h are the width and height of the two boxes, respectively. c_h , c_w represent the height and width of the smallest outer rectangle of the prediction box and the target box.

Derived from IoU , the $ECIoU_Loss$ integrates length and width information from both the target and prediction frames. This effectively resolves the penalty failure observed in $CIoU$ related to changes in aspect ratio, leading to a notable improvement in network convergence.

2.6. GhostNet_Triplet_YOLO v8s Target Detection Algorithm

The YOLO v8 target detection model comprises four key components: input, backbone, neck, and prediction. However, its heavy reliance on standard convolutions leads to memory-intensive operations, posing challenges for deployment on smaller programs and edge devices.

This paper introduces Ghost-Net_Triplet_YOLOv8s, which is a maize leaf disease identification model that leverages the lightweight GhostNet architecture to replace the YOLO v8s network's backbone. The Ghost modules in this architecture generate a portion of the feature maps with fewer convolution operations, completing them through simple linear operations. This approach reduces parameters and computational load while preserving feature map richness. Consequently, the model, despite being lightweight, maintains the capability to capture sufficient features for identifying various disease types. Additionally, the C2f and Conv modules in the network's head are replaced with C3Ghost and Ghost-Net modules, significantly reducing model complexity and simultaneously enhancing detection accuracy.

A noteworthy addition is the incorporation of the Triplet Attention mechanism in the head. This mechanism effectively models channel and spatial dimensions with minimal additional parameters, enabling nuanced recognition and emphasizing critical features within the image while maintaining computational efficiency. This is crucial for detection efficacy, allowing the model to distinguish between visually similar diseased areas and healthy plant tissue. Finally, the original $CIoU_Loss$ is replaced with the $ECIoU_Loss$ loss function, rectifying the penalty failure related to aspect ratio changes in $CIoU$ and enhancing network convergence. This methodology significantly reduces the model's memory footprint while maintaining accuracy.

In summary, the integration of these three enhancement modules has significantly improved the maize leaf disease identification model's detection efficiency and accuracy while maintaining or even reducing computational resource consumption. This optimization is achieved through intelligent simplification of the model architecture, effective emphasis on key features, and precise adjustment of the loss function. These techniques are particularly well suited for resource-constrained devices, such as mobile devices and WeChat mini-programs, providing a technological guarantee for real-time and accurate disease monitoring. The schematic diagram depicting the corn disease identification model is showcased in Figure 7.

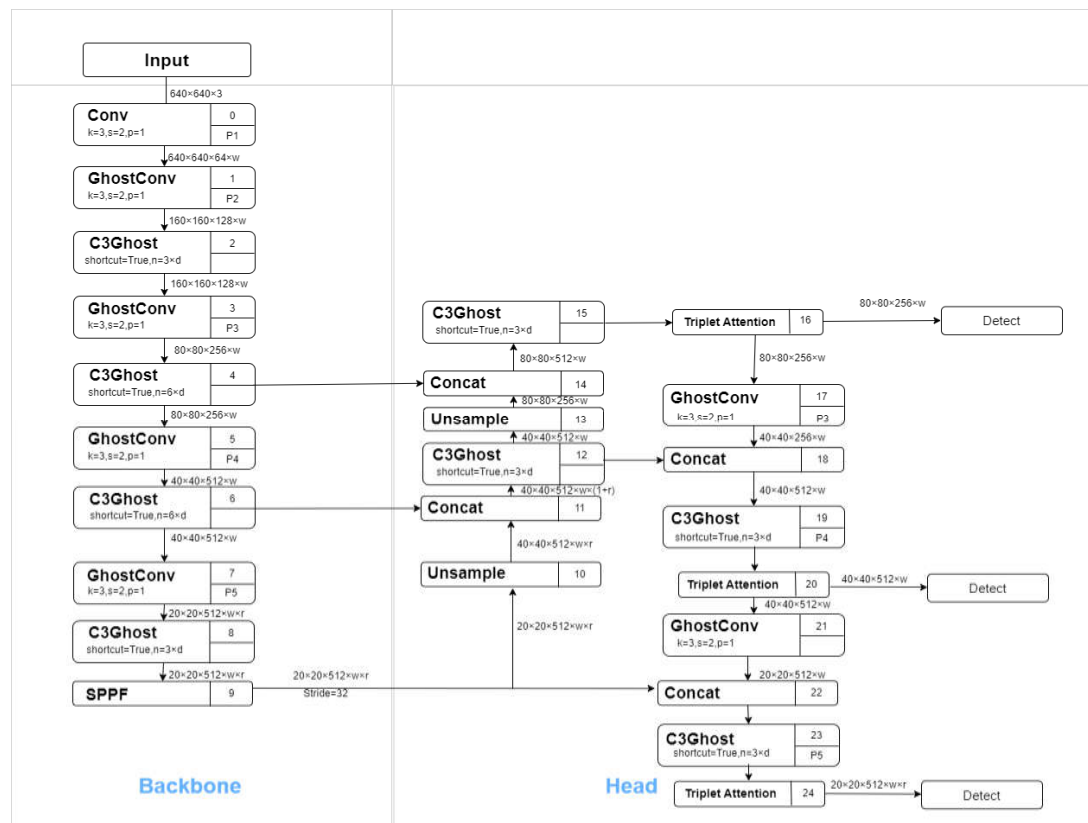


Figure 7. Overall schematic diagram of improved YOLO-v8s.

3. Results

3.1. Comprehensive Test Platform and Training Evaluation

Our research utilizes the PyTorch deep learning framework as the foundation for our model training platform. The laboratory server is equipped with an NVIDIA GeForce RTX 4090 GPU, boasting 32 GB of GPU memory, and it operates on a Windows 10 system. The optimization and enhancement of the algorithm are performed using Python 3.8, PyTorch 1.11.0, and the CUDA 13.0 deep learning framework. Throughout the training process, we employ an SGD optimizer with a momentum of 0.937. The learning rate starts at an initial value of 0.01 and gradually decreases to a final value of 0.0001. The training regimen involves a batch size of 16 and spans a maximum of 150 training epochs. Additionally, an IoU threshold of 0.7 guides the training protocol. This configuration effectively facilitates the training and evaluation of the model, ensuring its performance and accuracy across various tasks.

To select the optimal model, this article employs mAP (mean average precision), Precision, and Recall (recall rate) as key metrics to evaluate the detection performance of the model comprehensively. Formulas (10)–(12) govern these assessments. mAP is the average AP across each category, serving as a measure of the target detection algorithm's

overall performance. Precision represents the ratio of correctly classified positive examples to the total classified positive examples, assessing the model's accuracy. Meanwhile, Recall measures the number of correctly identified positive examples out of all actual positive examples, gauging the model's effectiveness in capturing all positives within the dataset.

$$mAP = \frac{\sum_{k=1}^N PR}{N} \quad (10)$$

$$Precision = \frac{TP}{TP + FP} \quad (11)$$

$$Recall = \frac{TP}{TP + FN} \quad (12)$$

where N —number of detection sample categories; TP —the number of real positive samples (the number of positive samples detected as positive samples), which is measured by the number of detection frames with $IoU > 0.5$, that is, the number of correct detections; FP —the number of false positive samples (the number of negative samples that are detected as positive samples), which is measured by the detection box of $IoU \leq 0.5$, that is, the number of detection errors; FN —the number of false negative samples (the number of positive samples that are detected as negative samples), which is measured by the number of GTs that are not detected—that is, the number of missed detections.

3.2. Improvement of Model Detection Efficiency

To validate the efficacy of the proposed GhostNet_Triplet_YOLOv8s method in identifying various corn leaf diseases (*Cercospora* Leaf Spot, northern leaf blight, and rust) alongside healthy leaves, a comparative analysis was conducted against the original YOLO v8s model. The detailed findings are listed in Table 3.

Table 3. Comparative results of average precision mean for various target classes.

Corn Disease Categories	mAP@0.5/%	
	Methods of This Article	YOLO v8s
<i>Cercospora</i> Leaf Spot	88.4	88.6
Northern Leaf Blight	95.2	94.5
Rust	82.8	81.9
Health	99.4	99.4
All diseases	91.4	91.1

The findings from the table above indicate that this method achieves a mean average precision (mAP@0.5) of 99.4% for identifying healthy corn leaves, matching YOLO v8s performance. Notably, in detecting northern leaf blight and rust on corn leaves, the mAP@0.5 scores demonstrate a 0.7% and 0.9% improvement, reaching 95.2% and 82.8%, respectively, surpassing the YOLO v8s model. Overall disease identification experienced a 0.3% increase in mean average precision (mAP@0.5) compared to the original model. These results emphasize the enhanced efficacy of this article's method in effectively identifying various corn diseases. A selection of test results are visualized in Figure 8.

The method proposed in this article employs grid division and predefined multiple anchor boxes within each grid, allowing the model to predict both category and location information for targets within these anchor boxes simultaneously. This strategy of gridding and anchor box generation enables the model to effectively capture objects of varying scales and shapes. In Figure 8, the detection results of our model on the test set (depicted in subgraphs (a–d) corresponding to *Cercospora* leaf spot, northern leaf blight, rust, and healthy leaves, respectively) are showcased. Notably, the model exhibits relatively high confidence in predicting northern leaf blight and healthy leaves due to their distinct characteristics. Conversely, *Cercospora* leaf spot and rust typically manifest as smaller lesions,

leading to multiple prediction boxes and relatively lower detection confidence. Nevertheless, the model demonstrates good adaptability across different disease types, accurately pinpointing the disease locations and exhibiting commendable generalization capabilities.

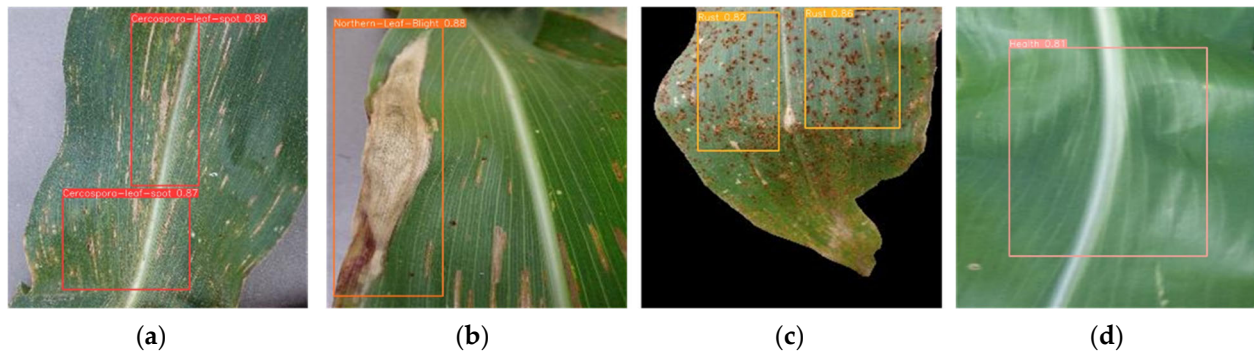


Figure 8. The test results of the method in this paper where (a) *Cercospora* leaf spot, (b) northern leaf blight, (c) rust, (d) health.

3.3. Ablation Experiment

To evaluate the efficacy of the corn disease identification method introduced in this article using GhostNet_Triplet_YOLOv8s and its enhancement over the original algorithm, an ablation experiment was devised.

3.3.1. Comparative Experiments on Backbone Networks

Distinct backbone networks can exert varying degrees of impact on the detection accuracy and memory of the model, as depicted in Table 4 below.

Table 4. Comparison of different backbone networks.

Network Model	Backbone Network	P/%	R/%	mAP@0.5%	FLOPs/G	Weight Size/MB
YOLO v8s	CSPDarknet-53 (C2f)	86.7	89.3	91.1	28.8	22.5
YOLO v8s	RepVGGBlock	85.3	82.6	88.4	8.3	5.5
YOLO v8s	FasterNeXt	84.5	82.0	88.0	8.3	5.5
YOLO v8s	PP-LCNet	83.3	81.8	86.6	17.4	12.2
YOLO v8s	Vanillanet	80.4	81.5	85.3	10.6	7.6
YOLO v8s	MobileNetv3	82.5	82.4	87.1	18.2	13.8
YOLO v8s	GhostNet	84.5	82.8	88.4	8.3	5.0

Table 4 illustrates that upon replacing lightweight networks like RepVGGBlock [20], FasterNeXt, PP-LCNet, Vanillanet [21], MobileNetv3 and GhostNet, there was a decline in the mean average precision (mAP). However, this decline was accompanied by a significant reduction in the number of model parameters, indicating the potential trade-off between network depth reduction and precision in detecting corn leaf lesions. Among the five lightweight backbone networks assessed, RepVGGBlock and GhostNet displayed the highest mean average precision, reaching 88.4%. The model featuring GhostNet as the backbone consistently demonstrates superior performance in accuracy, recall, and mean average precision compared to MobileNetv3. Notably, GhostNet stands out as the smallest model, with only 5.0 MB and 8.3 G FLOPs, indicating superior mobile performance compared to other backbone networks, offering an advantageous edge in deployment. Consequently, this article opts for GhostNet as the chosen backbone network.

3.3.2. Combination Experiment of Backbone Network and Network Neck

Based on GhostNet backbone network selection, various neck architectures were chosen for comparative testing, as detailed in Table 5 below.

Table 5. Comparison of network necks.

Backbone Network	Neck	P/%	R/%	mAP@0.5%	FLOPs/G	Weight_Size/MB
GhostNet	GhostNet	83.7	83.7	88.5	16.3	3.7
GhostNet	LADH	83.2	82.5	87.2	20.8	5.2
GhostNet	AFPN	83.4	81.8	86.4	16.4	5.3
GhostNet	P2	84.4	82.8	88.4	15.5	5.0

The ablation experiment conducted to assess the impact of incorporating different neck architectures into the target detection network, using GhostNet as the backbone, yielded insightful results. Combinations were tested with the GhostNet neck framework, LADH, AFPN, and P2 to identify the most optimal approach. As illustrated in Table 5, the combination of the GhostNet backbone with the GhostNet neck framework produces more favorable results compared to the other three combinations. This specific combination showcases precision and recall rates of 83.7%, a mean average precision (mAP) of 88.5%, 16.3 G FLOPs, and a compact model size of 3.7 MB. Despite the reduction in network parameters, there is a notable enhancement in both recall and precision metrics. Hence, GhostNet emerged as the preferred neck network for this study.

3.3.3. Combination Experiment of Backbone Network, Network Neck and Attention Mechanism

Expanding on the preferred combination method highlighted in Table 5, which incorporates GhostNet as the backbone network and the GhostNet neck framework, diverse attention mechanisms were introduced. These attention mechanisms were strategically placed within the network's neck layer, inducing alterations in detection precision and network weight size. Following several experiments, all the attention mechanisms discussed in this article were integrated into the neck layer. The comparative outcomes of these experiments are meticulously outlined in Table 6.

Table 6. Comparison of different attention mechanisms.

Model	Backbone Network	Neck	Attention Mechanism	P/%	R/%	mAP@0.5%	FLOPs/G	Weight_Size/MB
Model 1	GhostNet	GhostNet	-	83.7	83.7	88.5	16.3	3.7
Model 2	GhostNet	GhostNet	EMA	88.6	86.1	90.8	16.7	12.3
Model 3	GhostNet	GhostNet	CBAM	86.8	86.9	90.9	16.9	12.2
Model 4	GhostNet	GhostNet	SimAM	88.4	87.3	90.8	16.4	12.2
Model 5	GhostNet	GhostNet	ECA	87.0	86.6	91.1	16.7	12
Model 6	GhostNet	GhostNet	Triplet Attention	87.8	87.7	91.3	16.39	11.2
Model 7	GhostNet	GhostNet	BiFPN	88.9	85.4	91.0	19.0	11.7

Table 6 compares six attention mechanisms, including EMA [22], CBAM [23], SimAM [24], ECA [25], Triplet Attention, and BiFPN [26], under the conditions of the GhostNet backbone network and the GhostNet neck framework. The findings highlight significant improvements in precision, recall, and mAP attributed to these attention mechanisms. However, their integration led to a noticeable increase in model parameters. Particularly, Triplet Attention stood out as the most promising mechanism which not only improved performance but also effectively reduced the number of parameters and simplified the YOLO v8s model. This reduction in parameters enhances its suitability for deployment on edge devices compared to other mechanisms.

3.3.4. Loss Function Comparison Test

To assess the impact of various bounding box loss functions on network detection accuracy, DIoU, GIoU, SiIoU, MPDIoU, ECIoU, and five other loss functions were compared against the CIoU loss function employed by YOLO v8s. These six loss functions were

integrated within the GhostNet backbone network and the GhostNet neck framework while incorporating Triplet Attention. Furthermore, the YOLO v8s model was tested alongside these mechanisms to identify the most effective loss function. The specific combination method is detailed in Table 7.

Table 7. Comparison of loss functions.

Model	Backbone Network	Neck	Attention	Loss	P/%	R/%	mAP@0.5%	FLOPs/G
Model 1	GhostNet	GhostNet	Triplet Attention	CIoU	87.8	87.5	91.3	16.3
Model 2	GhostNet	GhostNet	Triplet Attention	GIoU	86.4	87.8	90.4	16.3
Model 3	GhostNet	GhostNet	Triplet Attention	SIoU	87.8	86.5	91.4	16.3
Model 4	GhostNet	GhostNet	Triplet Attention	MPDIoU	87.8	87.9	90.8	16.3
Model 5	GhostNet	GhostNet	Triplet Attention	ECIoU	87.5	87.7	91.4	16.3

The experimental findings indicate that Model 5, utilizing the ECIoU loss function, exhibits the most optimal detection performance. Compared to Model 1, Model 5 showcases a 0.2% increase in recall, a 0.1% rise in mean average precision (mAP), and a negligible 0.3% decrease in precision, indicating commendable performance. Notably, when the mean average precision of Model 3 matches that of Model 5, the recall of Model 5 with ECIoU as the loss function is considerably higher than that of Model 3. Therefore, the addition of the ECIoU loss function can effectively improve the detection precision of the network.

3.3.5. Ablation Test Results

In order to further verify the effectiveness of the GhostNet_Triplet_YOLOv8s model, an ablation experiment was conducted based on YOLO v8s, as shown in Table 8.

Table 8. Ablation experiment results.

Model	P/%	R/%	mAP@0.5%	FLOPs/G	Weight Size/MB
YOLO v8s	86.7	89.3	91.1	28.8	22.5
YOLO v8s + GhostNet	84.5	82.8	88.4	8.3	5.0
YOLO v8s + GhostNet + GhostNet	83.7	83.7	88.5	16.3	3.7
YOLO v8s + GhostNet + GhostNet + Triplet Attention	87.8	87.7	91.3	16.39	11.2
YOLO v8s + GhostNet + GhostNet + Triplet Attention + ECIoU	87.5	87.7	91.4	16.39	11.2

The data in Table 8 showcase four key enhancements. Firstly, the substitution of GhostNet for the original YOLO v8s backbone notably reduces the model complexity and volume of the model, although it leads to a 2.7% decrease in mAP. Secondly, the incorporation of the GhostNet lightweight neck structure reduces model weight by 26%, increases the recall rate by 0.9%, and slightly boosts mAP by 0.1%. Thirdly, the introduction of the Triplet Attention mechanism yields significant improvements, elevating precision by 4.1%, recall rate by 4%, and mAP by 2.8%. Finally, replacing the original model's loss function with ECIoU results in a 0.1% mAP increase while substantially reducing model complexity by 43.1% and volume by 50.2% compared to the original model. Overall, these advancements effectively simplify the model volume and enhance computational efficiency while maintaining a high level of accuracy.

3.4. Comparison of Different Algorithm Types

To further validate the performance of our GhostNet_Triplet_YOLOv8s algorithm relative to current target detection models, we conducted experimental comparisons under the same conditions (including consistent parameter settings and the same dataset). Precision, recall, mAP@0.5, and model parameter volume were the metrics employed for comparison. In the present study, the algorithm was compared with Fast RCNN, SSD, YOLO v5s, YOLO

v7, YOLOv7-tiny, YOLO v8n, YOLOX, YOLO v4-tiny, and YOLO v8s. The outcomes of these experiments are detailed in Table 9.

Table 9. Network model identification accuracy and performance comparison.

Model	P/%	R/%	mAP@0.5/%	Weight Size/MB	Parameters/M	FLOPs/G
Fast RCNN	37.57	64.06	52.38	521.90	-	-
SSD	90.23	86.21	90.19	108.43	-	-
YOLO v5s	87.50	91.00	90.80	14.19	7.20	16.50
YOLO v7	77.70	79.00	81.90	74.90	36.90	104.70
YOLO v7-tiny	91.91	19.15	71.91	24.20	6.20	16.80
YOLO v8n	86.30	84.80	90.00	6.20	3.20	8.70
YOLOX	92.63	88.17	91.10	34.40	9.00	26.80
YOLO v4-tiny	75.48	46.18	62.08	22.40	6.10	16.46
YOLO v8s	86.70	89.30	91.10	22.50	11.20	28.80
Methods of this article	87.50	87.70	91.40	11.20	5.90	16.39

From Table 9, the optimized algorithm showcased a 0.8% precision boost and a 0.3% mAP increase compared to the original YOLO v8s model. Simultaneously, it exhibited a significant 50.2% reduction in weight and size as well as a reduction in parameters by 5.3 M and in FLOPs by 12.41 G. This signifies an enhanced detection accuracy alongside reduced model parameters, memory usage, and complexity relative to the original model. In comparison with Fast RCNN, SSD, YOLO v5s, YOLO v7, YOLO v7-tiny, YOLO v8n, YOLOX, and YOLO v4-tiny, the improved model displayed a substantial increase in mAP ranging from 0.3% to 39.02%, respectively. Furthermore, it considerably reduced parameters and FLOPs compared to these models. Notably, the comprehensive indicators of precision, recall rate, mAP, and the reduced weight size, parameters, and FLOPs reinforce the effectiveness of this enhanced model, especially in edge device applications.

3.5. Corn Leaf Disease Identification Applet

Corn diseases are a significant concern for corn growers in China, and the present study has taken steps to assist by creating a WeChat applet specifically for identifying these diseases. Users can upload images of corn diseases via the app, and these images are sent to a server developed using PyCharm with HTTP for data communication. The server employs GhostNet_Triplet_YOLOv8s for disease identification, providing results within just 1 s on an average network speed. Once processed, users receive details about the disease category, its probability, and recommended prevention methods. Additionally, this tool aims to empower both corn growers and researchers with better insights into corn diseases and their prevention. The applet underwent testing in Yunnan Agricultural University's corn experimental field, yielding promising recognition results, as illustrated in Figure 9.

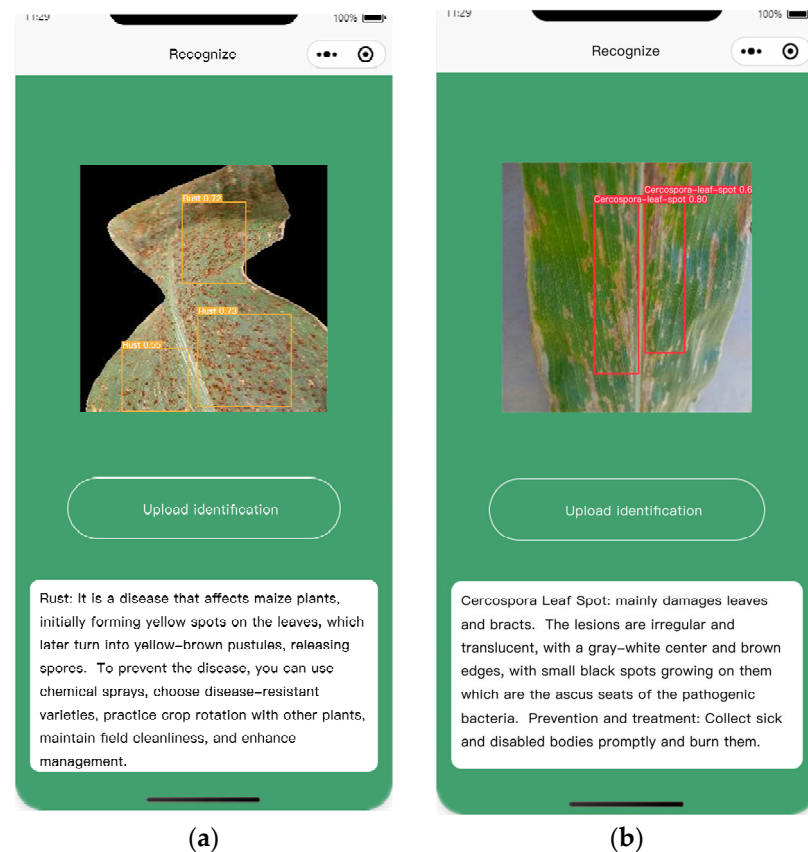


Figure 9. Partial recognition results of corn leaf diseases (a) corn rust disease recognition results (b) corn leaf spot disease recognition results.

4. Discussion

To assess the model's efficacy and suitability, this article validated both the original model and the methods introduced in this study using the public grape dataset for verification. The dataset comprises 3330 images across four diseases: BlackRot, GrapeEsca, GrapeHealthy, and LeafBlight.

The results explained in Table 10 reveal noteworthy enhancements in the proposed model compared to the original YOLO v8s. This enhanced model exhibits a 0.1% increase in precision and in recall rate, while there is also a 0.05% rise in mean average precision (mAP). These improvements extend beyond the corn disease dataset, demonstrating enhanced performance across various grape datasets. This improvement in generalization performance is crucial for the applicability of the model, showcasing its robustness across diverse datasets and scenarios. Thus, the model not only achieves high performance in specific tasks but also exhibits exceptional adaptability across broader field applications.

Table 10. Model test results comparison.

Model	P/%	R/%	mAP@0.5/%
YOLO v8s	99.30	99.60	99.25
Methods of this article	99.40	99.70	99.30

5. Conclusions

Addressing the shortcomings of traditional target detection methods in identifying corn leaf diseases and the limitations observed in YOLO series algorithms, this article introduces a corn disease detection method, GhostNet_Triplet_YOLOv8s. This enhanced algorithm swaps out YOLO v8s' primary network with the lightweight GhostNet and

modifies essential modules (C2f and Conv modules in the head with C3Ghost and Ghost-Net modules), boosting detection capabilities. The introduction of Triplet Attention and the ECIoU_Loss loss function significantly enhances detection performance and network convergence while reducing model complexity. Experimental results on a corn disease dataset reveal that the improved algorithm outperforms the original YOLO v8s in precision, mAP and other metrics. Additionally, in comparison with various target detection models, the improved model excels remarkable accuracy and efficiency metrics. Notably, the improved model exhibits superior performance not only on the corn disease dataset but also demonstrates outstanding generalization capabilities across different datasets and scenarios. To validate its practical use, a specialized app for rapid corn leaf disease identification was developed, aiming to reduce economic losses and provide effective solutions. Therefore, this model presents a comprehensive performance balance, promising efficient real-world applications. Furthermore, this research aims to expand validation across broader datasets, optimize the model for improved real-time performance on edge devices explore diverse disease detection methods, enhance interpretability techniques, and extend solutions to other crops, thereby providing a more reliable and efficient solution for agricultural disease detection.

Author Contributions: Supervision: J.Y.; conceptualization: R.L.; methodology: R.L., R.J. and S.L.; software: W.Q. and Y.H.; validation: R.L., Y.H. and Y.L.; research: R.L. and Y.L.; data curation: Y.L., W.Q., A.A., R.J. and Y.W.; writing—preparation of the original draft: R.L. and A.A.; Revised, proofreading and editing the manuscript: A.A. and J.Y. All authors have read and agreed to the published version of the manuscript.

Funding: The authors would like to express their sincere gratitude for the financial support provided by the Major Project of Yunnan Science and Technology, under Project No. 202302AE09002003, entitled “Research on the Integration of Key Technologies in Smart Agriculture”.

Institutional Review Board Statement: Not applicable.

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

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

1. Kai, S.; Zhikun, L.; Hang, S.; Chunhong, G. A research of maize disease image recognition of corn based on BP networks. In Proceedings of the 2011 Third International Conference on Measuring Technology and Mechatronics Automation, Shanghai, China, 6–7 January 2011; IEEE: New York City, NY, USA, 2011.
2. Perkins, J.; Pedersen, W. Disease development and yield losses associated with northern leaf blight on corn. *Plant Dis.* **1987**, *71*, 940–943. [\[CrossRef\]](#)
3. Smith, D.; White, D. Diseases of corn. *Corn Corn Improv.* **1988**, *18*, 687–766.
4. Yang, S.; Xing, Z.; Wang, H.; Dong, X.; Gao, X.; Liu, Z.; Zhang, X.; Li, S.; Zhao, Y. Maize-YOLO: A new high-precision and real-time method for maize pest detection. *Insects* **2023**, *14*, 278. [\[CrossRef\]](#) [\[PubMed\]](#)
5. Li, Y.; Sun, S.; Zhang, C.; Yang, G.; Ye, Q. One-stage disease detection method for maize leaf based on multi-scale feature fusion. *Appl. Sci.* **2022**, *12*, 7960. [\[CrossRef\]](#)
6. Zhang, X.; Zhu, D.; Wen, R. SwinT-YOLO: Detection of densely distributed maize tassels in remote sensing images. *Comput. Electron. Agric.* **2023**, *210*, 107905. [\[CrossRef\]](#)
7. Xue, Z.; Xu, R.; Bai, D.; Lin, H. YOLO-tea: A tea disease detection model improved by YOLOv5. *Forests* **2023**, *14*, 415. [\[CrossRef\]](#)
8. Sun, P.; Chen, G.; Cao, L. Image recognition of soybean pests based on attention convolutional neural network. *Chin. Agric. Mech* **2020**, *41*, 171–176.
9. Gao, F.; Fu, L.; Zhang, X.; Majeed, Y.; Li, R.; Karkee, M.; Zhang, Q. Multi-class fruit-on-plant detection for apple in SNAP system using Faster R-CNN. *Comput. Electron. Agric.* **2020**, *176*, 105634. [\[CrossRef\]](#)
10. Hou, Y.; Shi, G.; Zhao, Y.; Wang, F.; Jiang, X.; Zhuang, R.; Mei, Y.; Ma, X. R-YOLO: A YOLO-Based Method for Arbitrary-Oriented Target Detection in High-Resolution Remote Sensing Images. *Sensors* **2022**, *22*, 5716. [\[CrossRef\]](#) [\[PubMed\]](#)
11. Wu, Z.; Chen, X.; Gao, Y.; Li, Y. Rapid target detection in high resolution remote sensing images using YOLO model. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2018**, *42*, 1915–1920. [\[CrossRef\]](#)

12. Diao, Z.; Guo, P.; Zhang, B.; Zhang, D.; Yan, J.; He, Z.; Zhao, S.; Zhao, C.; Zhang, J. Navigation line extraction algorithm for corn spraying robot based on improved YOLOv8s network. *Comput. Electron. Agric.* **2023**, *212*, 108049. [[CrossRef](#)]
13. Han, K.; Wang, Y.; Tian, Q.; Guo, J.; Xu, C.; Xu, C. Ghostnet: More features from cheap operations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020.
14. Misra, D.; Nalamada, T.; Arasanipalai, A.U.; Hou, Q. Rotate to attend: Convolutional triplet attention module. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 3–8 January 2021.
15. Yu, J.; Wu, T.; Zhang, X.; Zhang, W. An efficient lightweight SAR ship target detection network with improved regression loss function and enhanced feature information expression. *Sensors* **2022**, *22*, 3447. [[CrossRef](#)] [[PubMed](#)]
16. Terven, J.; Cordova-Esparza, D. A comprehensive review of YOLO: From YOLOv1 to YOLOv8 and beyond. *arXiv* **2023**, arXiv:2304.00501.
17. Zhang, X.; Zhou, X.; Lin, M.; Sun, J. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.
18. Li, Y.; Yuan, G.; Wen, Y.; Hu, J.; Evangelidis, G.; Tulyakov, S.; Wang, Y.; Ren, J. Efficientformer: Vision transformers at mobile net speed. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 12934–12949.
19. Michele, A.; Colin, V.; Santika, D.D. Mobilenet convolutional neural networks and support vector machines for palmprint recognition. *Procedia Comput. Sci.* **2019**, *157*, 110–117. [[CrossRef](#)]
20. Tao, B.; Wang, Y.; Qian, X.; Tong, X.; He, F.; Yao, W.; Chen, B.; Chen, B. Photoelastic stress field recovery using deep convolutional neural network. *Front. Bioeng. Biotechnol.* **2022**, *10*, 818112. [[CrossRef](#)] [[PubMed](#)]
21. Chen, H.; Wang, Y.; Guo, J.; Tao, D. VanillaNet: The Power of Minimalism in Deep Learning. *arXiv* **2023**, arXiv:2305.12972.
22. Stone, A.A.; Shiffman, S. Ecological momentary assessment (EMA) in behavioral medicine. *Ann. Behav. Med.* **1994**, *16*, 199–202. [[CrossRef](#)]
23. 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.
24. 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, Virtual Event, 18–24 July 2021.
25. Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; Hu, Q. ECA-Net: Efficient channel attention for deep convolutional neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020.
26. Chen, J.; Mai, H.; Luo, L.; Chen, X.; Wu, K. Effective feature fusion network in BIFPN for small target detection. In Proceedings of the International Conference on Image Processing (ICIP), Anchorage, AK, USA, 19–22 September 2021; IEEE: New York City, NY, USA, 2021.

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.