agronomy

Article

Deep-Learning-Based Rice Disease and Insect Pest Detection on a Mobile Phone

Jizhong Deng [1,2,3,†], Chang Yang [1,3,†], Kanghua Huang [3,4], Luocheng Lei [1,3], Jiahang Ye [1,3], Wen Zeng [1,3], Jianling Zhang [1,3], Yubin Lan [2,3,4] and Yali Zhang [1,2,3,*]

[1] College of Engineering, South China Agricultural University, Guangzhou 510642, China; jz-deng@scau.edu.cn (J.D.); changyang@stu.scau.edu.cn (C.Y.); llc@stu.scau.edu.cn (L.L.); yjh@stu.scau.edu.cn (J.Y.); zengwen@scau.edu.cn (W.Z.); jlingzhang@scau.edu.cn (J.Z.)
[2] Guangdong Laboratory for Lingnan Modern Agriculture, Guangzhou 510642, China; ylan@scau.edu.cn
[3] National Center for International Collaboration Research on Precision Agricultural Aviation Pesticides Spraying Technology, Guangzhou 510642, China; kh_huang@stu.scau.edu.cn
[4] College of Electronic Engineering, South China Agricultural University, Guangzhou 510642, China
* Correspondence: ylzhang@scau.edu.cn
† These authors contributed equally to this work.

Abstract: The realization that mobile phones can detect rice diseases and insect pests not only solves the problems of low efficiency and poor accuracy from manually detection and reporting, but it also helps farmers detect and control them in the field in a timely fashion, thereby ensuring the quality of rice grains. This study examined two Improved detection models for the detection of six high-frequency diseases and insect pests. These models were the Improved You Only Look Once (YOLO)v5s and YOLOv7-tiny based on their lightweight object detection networks. The Improved YOLOv5s was introduced with the Ghost module to reduce computation and optimize the model structure, and the Improved YOLOv7-tiny was introduced with the Convolutional Block Attention Module (CBAM) and SIoU to improve model learning ability and accuracy. First, we evaluated and analyzed the detection accuracy and operational efficiency of the models. Then we deployed two proposed methods to a mobile phone. We also designed an application to further verify their practicality for detecting rice diseases and insect pests. The results showed that Improved YOLOv5s achieved the highest F1-Score of 0.931, 0.961 in mean average precision (mAP) (0.5), and 0.648 in mAP (0.5:0.9). It also reduced network parameters, model size, and the floating point operations per second (FLOPs) by 47.5, 45.7, and 48.7%, respectively. Furthermore, it increased the model inference speed by 38.6% compared with the original YOLOv5s model. Improved YOLOv7-tiny outperformed the original YOLOv7-tiny in detection accuracy, which was second only to Improved YOLOv5s. The probability heat maps of the detection results showed that Improved YOLOv5s performed better in detecting large target areas of rice diseases and insect pests, while Improved YOLOv7-tiny was more accurate in small target areas. On the mobile phone platform, the precision and recall of Improved YOLOv5s under FP16 accuracy were 0.925 and 0.939, and the inference speed was 374 ms/frame, which was superior to Improved YOLOv7-tiny. Both of the proposed improved models realized accurate identification of rice diseases and insect pests. Moreover, the constructed mobile phone application based on the improved detection models provided a reference for realizing fast and efficient field diagnoses.

Keywords: rice diseases and insect pests; mobile phone application; deep learning; object detection

check for updates

Citation: Deng, J.; Yang, C.; Huang, K.; Lei, L.; Ye, J.; Zeng, W.; Zhang, J.; Lan, Y.; Zhang, Y. Deep-Learning-Based Rice Disease and Insect Pest Detection on a Mobile Phone. *Agronomy* **2023**, *13*, 2139. https://doi.org/10.3390/agronomy13082139

Academic Editors: Spyros Fountas, Georgios Leontidis and Borja Espejo-García

Received: 30 June 2023
Revised: 3 August 2023
Accepted: 14 August 2023
Published: 15 August 2023

1. Introduction

Effective detection and monitoring of rice diseases and insect pests is crucial for ensuring a high and stable rice yield [1]. The traditional manual identification method cannot meet the increasing demand for field disease and pest detection due to its low efficiency and inconsistent diagnostic results. Therefore, it is important to find an objective and efficient automatic detection method for high-quality rice production.

In the last two decades, many scholars have performed research on image-based automatic identification of diseases and insect pests in crops. Traditional machine-learning methods can classify crop diseases and pests through manually designed feature extraction and classification [2–4]. However, due to the difficulty with manually designed features and the poor generalizability of the specific models, many achievements have not been put into actual production [5–7]. With advances of artificial intelligence, the deep-learning method, which has powerful feature learning and extraction capabilities, has been widely used in complex agricultural scenarios [8–10], particularly in disease and insect pest identification in crops such as tomatoes [11], apples [12], cassava [13] and rice. In research on rice diseases and insect pests, Rahman et al. (2020) [14] suggested a real-time classification approach to classify 9 types of rice disease and pest images (e.g., false smut, brown plant hopper, and background leaf blight) conducted by a deep convolutional neural network (CNN). Joshi et al. (2022) [15] proposed a mobile phone application named RiceBioS to identify biotic stress in rice crops (rice blast and bacterial leaf blight). Deng et al. (2021) [16] developed a smartphone rice disease identification application that combined deep-learning classification methods. Using the deep CNN model deployed on an online server, it classified 6 types of rice diseases and insect pests such as rice leaf blast, false smut, and neck blast. The above research categorized crop diseases and insect pests by using deep-learning image classification, but this by itself cannot accurately and effectively detect areas of crop diseases and pests in images. Although the online deployment of deep-learning models and mobile–client interaction can to some extent alleviate the problem of high consumption of computing resources because of the complicated structure deep-learning models, it increases the response time of model inference and identification, which is not conducive to offline detection and identification of rice diseases and insect pests.

Deep-learning object detection is an end-to-end method that accurately identifies categories of diseases and insect pests and extracts occurrence information from images. The current dominant object detection algorithms are mainly two types: two-stage and one-stage. The Faster R-CNN introduced by Ren et al. (2015) [17] is a representative two-stage algorithm. However, it has not been able to achieve rapid identification due to its two-stage detection method of feature extraction and region-proposal generation by a convolutional network. Redmon et al. (2016) [18] introduced the one-stage object detection algorithm You Only Look Once (YOLO), which achieves end-to-end output detection of objects through a single CNN feature extraction network. Unlike Faster R-CNN, this algorithm combines traditional object detection with target location and classification, thereby greatly increasing the model reasoning speed, which makes it useful for detection tasks in complex farmland environments.

The purpose of this study is to achieve offline detection of rice diseases and insect pests based on mobile phone devices. By combining deep-learning object detection with a mobile phone terminal, we proposed two improved detection and identification models. First, the Improved YOLOv5s model was constructed by introducing Ghost modules as the basic extraction feature to replace the standard convolution modules of the YOLOv5s model. The Improved YOLOv7-tiny model was constructed by introducing the Convolutional Block Attention Module (CBAM) into the YOLOv7-tiny model. In addition, SIoU was applied to replace the original model loss function. Then we compared and analyzed the detection accuracy and operational efficiency of the models. Finally, the improved models were transplanted to an Android mobile phone platform, and a rice disease and pest identification application was built to realize offline mobile phone detection of diseases and insect pests.

2. Materials and Methods

2.1. Image Acquisition and Enhancement

2.1.1. Image Acquisition

Due to the diversity of rice diseases and insect pests as well as the high randomness of disease occurrence, there are currently few open-source rice disease databases, so it is difficult to gather a large amount of data on diseases and insect pests. This study carried out research on six diseases and insect pests as shown in Figure 1: Cnaphalocrocis medinalis, rice smut, rice blast, streak disease, sheath blight, and chilo suppressalis. First, we used a mobile phone to obtain visible light images of the diseases and pests in multiple experimental areas. Then we expanded the image dataset through data enhancement methods. Finally, we trained the disease and pest detection models by using the above-enhanced dataset. This research was carried out for the experiments of rice disease and insect pest data photography and collection in three experimental bases in Guangdong, China in 2020 and 2021: Zengcheng District in Guangzhou, Guangdong, China (113.643906 N, 23.242985 E), Gaoyao District in Zhaoqing, Guangdong, China (112.184548 N, 23.134322 E) and Xinhui District in Jiangmen, Guangdong, China (112.828022 N, 22.441844 E). The data collection situation is shown in Table 1.

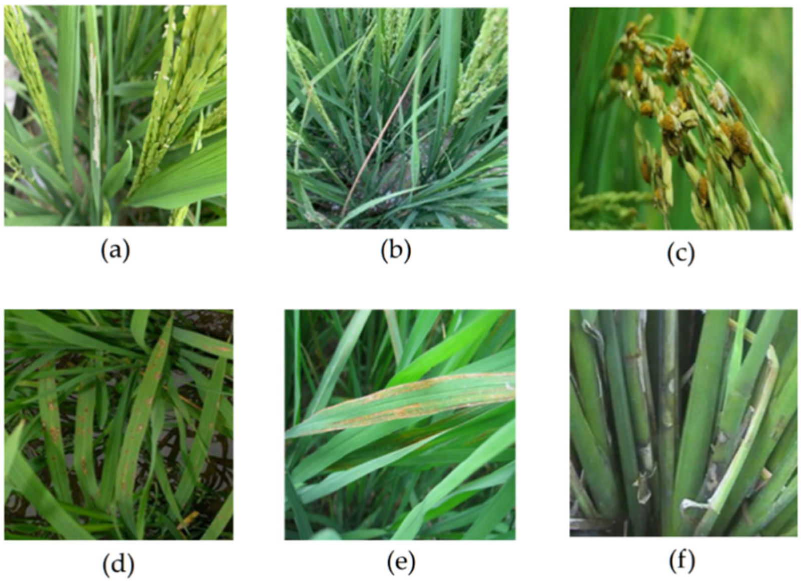


Figure 1. Sample images of rice diseases and insect pests. (**a**) Cnaphalocrocis medinalis; (**b**) Chilo suppressalis; (**c**) Rice smut; (**d**) Rice blast; (**e**) Streak disease; (**f**) Sheath blight.

Table 1. Rice disease and insect pest images dataset.

Disease Name	Acquisition Time	Data Source	Photograph	Data Enhancement	Total
Cnaphalocrocis medinalis	October 2020	Zengcheng District Experimental Base and Gaoyao District Test Base	487	63	550
Chilo suppressalis	October 2020	Zengcheng District Experimental Base	276	274	550
Rice smut	April 2021	Zengcheng District Experimental Base	209	341	550
Rice blast	March 2021	Zengcheng District Experimental Base and Xinhui District Experiment Base	236	314	550
Streak disease	March 2021	Zengcheng District Experimental Base	184	366	550
Sheath blight	March 2021	Zengcheng District Experimental Base	162	388	550

2.1.2. Image Enhancement

Different image enhancement methods were applied to expand the rice diseases and insect pests image dataset to reduce the impact of quantitative differences between the different dataset categories: image scaling, rotation, vertical and horizontal flipping, translation, and HSV adjustment. The data enhancement effect is shown in Figure 2. From Table 1, the total number of expanded image datasets was increased to 3300, and the image samples were divided into training and test sets. In the training set, there were 3000 images, and the number of images of each rice disease and insect pest category was 500. In the test set, there were 300 images, and the number of images of each rice disease and pest category was 50. The test set consisted of original images of rice diseases and insect pests without data enhancement. In this study, the constructed image dataset was labeled by the object detection labeling method. The distribution of labeling boxes of each category in the training and test sets is shown in Figure 3.

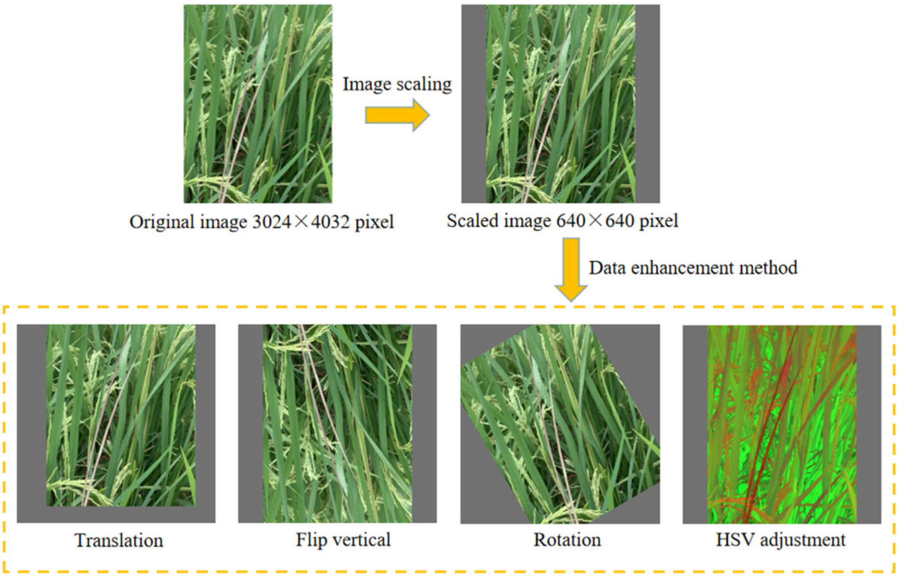


Figure 2. Image enhancement process.

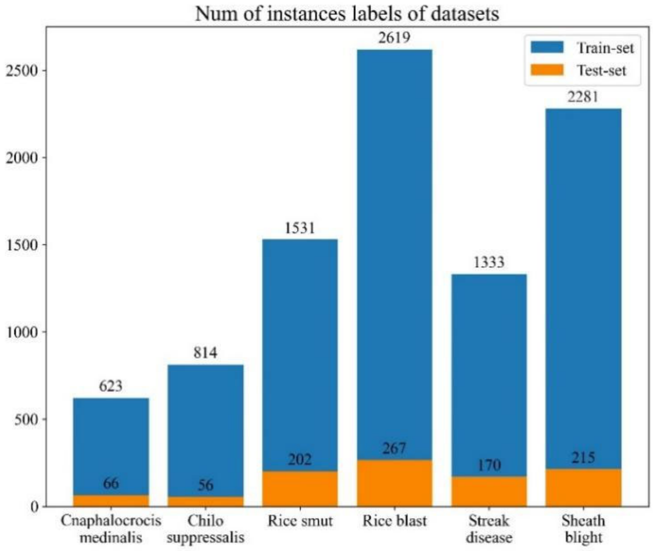


Figure 3. The distribution of labeling boxes of each category in the training set and test set.

2.2. Construction of Rice Disease and Pest Detection Model

2.2.1. Construction of Improved YOLOv5s Rice Diseases and Pests Detection Model

YOLOv5 is an object detection algorithm based on a one-state anchor detection method released by Uitralytics LLC [19]. YOLOv5s is a lightweight model structure of YOLOv5. Its network structure includes three parts: the backbone network, neck, and head structures. Among them, the backbone network includes the CBS, C3, and SPPF modules, the main function of which is to extract image features through down-sampling operations. The neck structure adopts a structure that combines FPN [20] and PAN [21], mainly achieving the fusion of shallow graphic features in the backbone layer and deep semantic features in the detection layer, which enables the model to obtain richer feature information. The head structure outputs the detection results of the target objects in different scales through three detection layers: category probability, probability score, and position information of the target boundary box.

Compared to traditional object detection networks, YOLOv5 has advantages such as higher accuracy and faster inference speed. However, because they use many standard convolutions in the model structure, these modules are not conducive to operation on mobile phones or embedded devices with limited memory and computing resources. Although lightweight models such as the MobileNet [22–24] and ShuffleNet series [25,26] can achieve good performance with very few floating point operations per second (FLOPs), the correlation and redundancy between their feature maps have not been well used. The Ghost module proposed by Han, K. et al. [27] is a model compression method that can generate more feature maps with fewer parameters to ensure redundancy of feature information, reduce network parameters and calculation and ensure network accuracy, thereby improving calculation speed and reducing the latency of the model. The feature extraction process of the Ghost module shown in Figure 4b is as follows: first, it reduces input feature map channels through a convolutional operation to obtain the intrinsic feature map. Then the Ghost feature maps are obtained by using the cheap operations of linear transformation with different convolution kernel sizes (usually 4 × 4 or 5 × 5) to process each channel of the intrinsic feature map. Finally, the intrinsic feature map and all the Ghost feature maps are spliced and output by the concatenating operation.

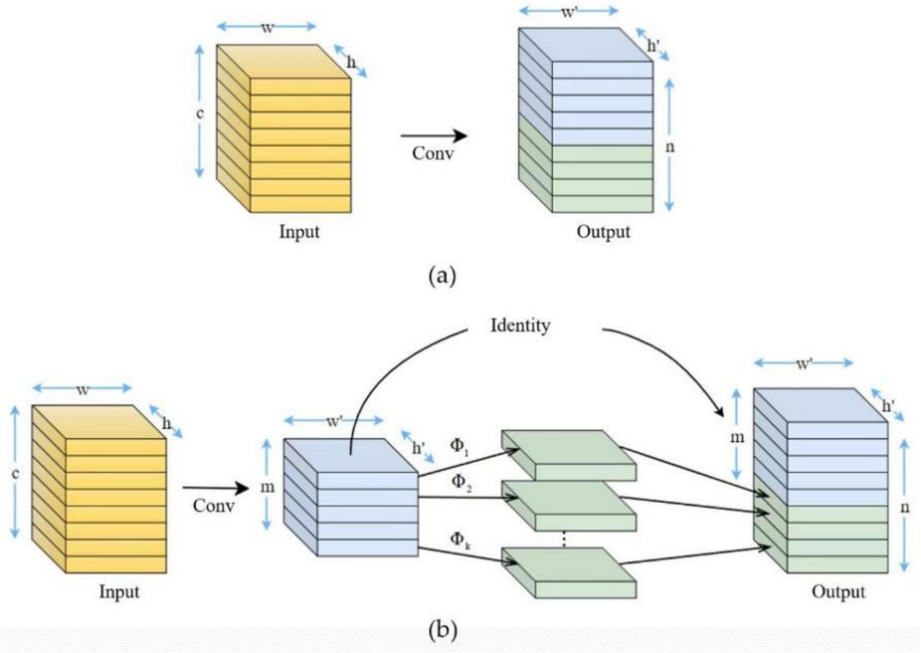


Figure 4. Feature extraction process of Ghost module and standard convolution module. (**a**) The feature extraction process of standard convolution module; (**b**) the feature extraction process of Ghost module.

Assume that the shapes of the input and output maps are $h \times w \times c$ and $h' \times w' \times n$. The convolutional filter size (represented by Conv in Figure 4) is $k \times k$., In the linear transformation process for the Ghost module, it is assumed that the intrinsic feature map channel quantity is m; Φ_k represents the operations of linear transformation; and the transformation quantity is s. Since the Ghost transformation process includes an identity transformation, the effective transformation quantity is $s - 1$. Therefore, the mathematical expression of the parameter calculation-amount ratio between the Ghost module and the standard convolution module is as follows. Generally, s is much smaller than the quantity of input feature map channels c:

$$\frac{\frac{n}{s} \cdot h' \cdot w' \cdot c \cdot k \cdot k + (s-1) \cdot \frac{n}{s} \cdot h' \cdot w' \cdot k \cdot k}{n \cdot h' \cdot w' \cdot c \cdot k \cdot k} = \frac{s + c - 1}{s \cdot c} \approx \frac{1}{s} \tag{1}$$

From Formula (1), the parameter calculation amount of the Ghost module is about $1/s$ of that of the standard convolution. In this study, referring to the parameter settings of GhostNet, the hyperparameter s of the Ghost module was set to 2. Therefore, compared to standard convolution, the Ghost module reduced the parameter calculation amount by about half, and it can be seen that replacing the standard convolution with the Ghost module led to a significant reduction in model computational complexity.

To build a rice disease and insect pest detection model suitable for mobile platform deployment, YOLOv5s was improved by combining the Ghost module and Ghost Bottle neck structure proposed by Han, K. et al. [27]. The Ghost Bottleneck was similar to the basic residual block in ResNet [28], which is mainly composed of two Ghost modules stacked to achieve the expansion and compression of the feature map channels. In this study, the Ghost module was used to replace the convolution module in the backbone and head structure of the original model, and the Ghost bottleneck structure was used to improve the C3 module of the model. The structures of the original YOLOv5s model and the Improved YOLOv5s model are shown in Figure 5a,b.

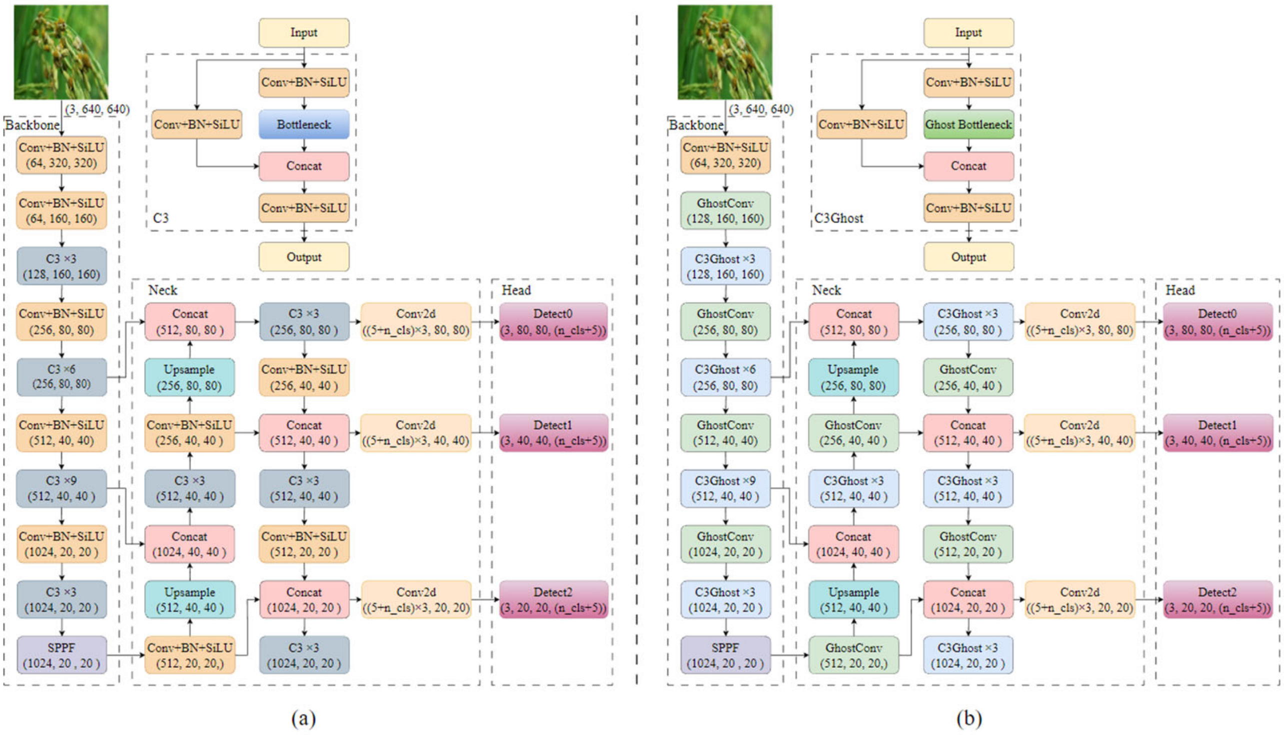

Input
Conv+BN+SiLU
Conv+BN+SiLU
Bottleneck
Concat
Conv+BN+SiLU
C3
Output
(3, 640, 640)
Backbone
Conv+BN+SiLU (64, 320, 320)
Conv+BN+SiLU (64, 160, 160)
C3 ×3 (128, 160, 160)
Conv+BN+SiLU (256, 80, 80)
C3 ×6 (256, 80, 80)
Conv+BN+SiLU (512, 40, 40)
C3 ×9 (512, 40, 40)
Conv+BN+SiLU (1024, 20, 20)
C3 ×3 (1024, 20, 20)
SPPF (1024, 20 , 20)
Neck
Concat (512, 80, 80)
Upsample (256, 80, 80)
Conv+BN+SiLU (256, 40, 40)
C3 ×3 (512, 40, 40)
Concat (1024, 40, 40)
Upsample (512, 40, 40)
Conv+BN+SiLU (512, 20, 20,)
C3 ×3 (256, 80, 80)
Conv+BN+SiLU (256, 40, 40)
Concat (512, 40, 40)
C3 ×3 (512, 40, 40)
Conv+BN+SiLU (512, 20, 20)
Concat (1024, 20, 20)
C3 ×3 (1024, 20, 20)
Conv2d ((5+n_cls)×3, 80, 80)
Conv2d ((5+n_cls)×3, 40, 40)
Conv2d ((5+n_cls)×3, 20, 20)
Head
Detect0 (3, 80, 80, (n_cls+5))
Detect1 (3, 40, 40, (n_cls+5))
Detect2 (3, 20, 20, (n_cls+5))
(a)
Input
Conv+BN+SiLU
Conv+BN+SiLU
Ghost Bottleneck
Concat
Conv+BN+SiLU
C3Ghost
Output
(3, 640, 640)
Backbone
Conv+BN+SiLU (64, 320, 320)
GhostConv (128, 160, 160)
C3Ghost ×3 (128, 160, 160)
GhostConv (256, 80, 80)
C3Ghost ×6 (256, 80, 80)
GhostConv (512, 40, 40)
C3Ghost ×9 (512, 40, 40)
GhostConv (1024, 20, 20)
C3Ghost ×3 (1024, 20, 20)
SPPF (1024, 20 , 20)
Neck
Concat (512, 80, 80)
Upsample (256, 80, 80)
GhostConv (256, 40, 40)
C3Ghost ×3 (512, 40, 40)
Concat (1024, 40, 40)
Upsample (512, 40, 40)
GhostConv (512, 20, 20,)
C3Ghost ×3 (256, 80, 80)
GhostConv (256, 40, 40)
Concat (512, 40, 40)
C3Ghost ×3 (512, 40, 40)
GhostConv (512, 20, 20)
Concat (1024, 20, 20)
C3Ghost ×3 (1024, 20, 20)
Conv2d ((5+n_cls)×3, 80, 80)
Conv2d ((5+n_cls)×3, 40, 40)
Conv2d ((5+n_cls)×3, 20, 20)
Head
Detect0 (3, 80, 80, (n_cls+5))
Detect1 (3, 40, 40, (n_cls+5))
Detect2 (3, 20, 20, (n_cls+5))
(b)

Figure 5. The structure of the YOLOv5s model and the Improved YOLOv5s model. (**a**) The structure of the YOLOv5s model; (**b**) the structure of the Improved YOLOv5s model.

2.2.2. Construction of Improved YOLOv7-Tiny Rice Diseases and Pests Detection Model

The main contribution of YOLOv7 proposed by Wang, C.Y. et al. [29] lies in the introduction of structural reparameterization and dynamic label allocation strategies for model optimization, as well as the introduction of expansion and composite scaling methods, which not only improves the efficiency of model parameters and calculation, but also significantly reduces model parameters, thereby improving inference speed and model detection accuracy. YOLOv7-tiny is also a lightweight model structure of YOLOv7. Its network structure includes two main sections: backbone and head structures. The functions of each section are the same as those of YOLOv5s. The difference is that the neck and head structures are merged into the head structure.

To make the constructed rice disease and insect pest identification model suitable for mobile phone platform deployment, we used YOLOv7's lightweight model YOLOv7-tiny for experimental analysis. To improve the model accuracy, YOLOv7-tiny was combined with the CBAM [30] (as shown in Figure 6). It uses a combination of the Channel Attention Module (CAM) and the Spatial Attention Module (SAM). The feature map is first input into the CAM to complete channel attention recalibration of the original feature through processing of Global Average Pooling (GAP), Global Max Pooling (GMP) and Shared Multilayer Perceptron (MLP) layers. Therefore, the mathematical expression of the channel attention module is as follow:

$$CAM(F) = \sigma(W_1(W_0(GAP(F))) + W_1(W_0(GMP(F)))) \tag{2}$$

where F denotes the input feature map with the shape of $H \times W \times C$; σ denotes the sigmoid function; and W_0 and W_1 denote the weights of the MLP layers. The output of the CAM module is a 1D channel attention map of the shape $1 \times 1 \times C$.

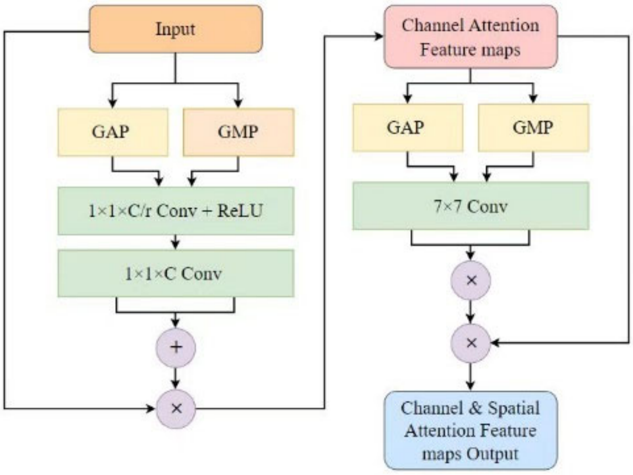

Input
GAP
GMP
1×1×C/r Conv + ReLU
1×1×C Conv
+
×
Channel Attention Feature maps
GAP
GMP
7×7 Conv
×
×
Channel & Spatial Attention Feature maps Output

Figure 6. The CBAM module.

The channel attention feature map was then input into the SAM module and processed by the GAP, GMP and a convolution layer. This module can be expressed by the following mathematical expression:

$$SAM(F) = \sigma\left(f^{7\times7}(GAP(F); GMP(F))\right) \tag{3}$$

where $f^{7\times7}$ denotes a convolution layer with a 7×7 filter size, and the output of the SAM module is a 2D attention map of the shape of H × W × 1. Therefore, the $CBAM$ module can be expressed as follows:

$$CBAM(F) = SAM(CAM(F) \otimes F) \otimes (CAM(F) \otimes F) \tag{4}$$

where $\otimes$ denotes the element-wise multiplication operation.

After processing the CBAM module, the new feature maps get the attention weight in the channel and space dimensions, which greatly improves the relationship between each feature in the channel and space, so that it is more conducive to extracting the effective features of the target from the model. In this study, the last three convolutional layers of the head structure of YOLOv7-tiny were improved, of which the first two layers retained the RepConv structure [31] of YOLOv7, and the last layer was replaced by the CBAM module. The structures of the original YOLOv7-tiny and the Improved YOLOv7-tiny models are shown in Figure 7a,b.

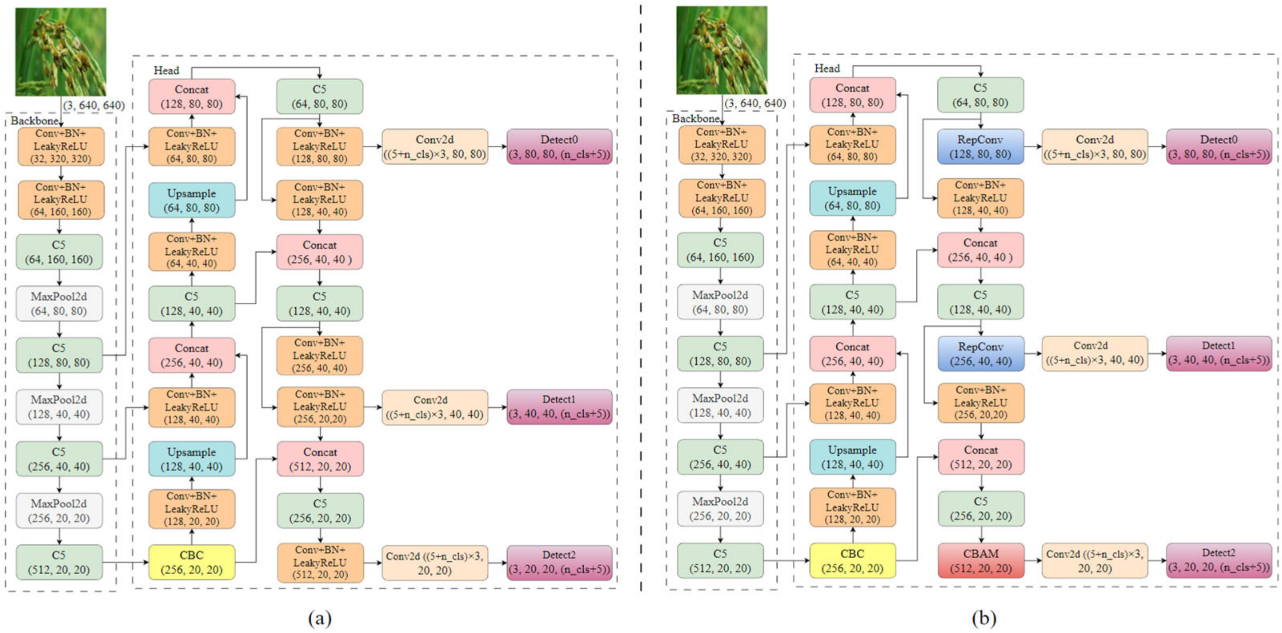

(3, 640, 640)
Backbone
Conv+BN+ LeakyReLU (32, 320, 320)
Conv+BN+ LeakyReLU (64, 160, 160)
C5 (64, 160, 160)
MaxPool2d (64, 80, 80)
C5 (128, 80, 80)
MaxPool2d (128, 40, 40)
C5 (256, 40, 40)
MaxPool2d (256, 20, 20)
C5 (512, 20, 20)
Head
Concat (128, 80, 80)
Conv+BN+ LeakyReLU (64, 80, 80)
Upsample (64, 80, 80)
Conv+BN+ LeakyReLU (64, 40, 40)
C5 (128, 40, 40)
Concat (256, 40, 40)
Conv+BN+ LeakyReLU (128, 40, 40)
Upsample (128, 40, 40)
Conv+BN+ LeakyReLU (128, 20, 20)
CBC (256, 20, 20)
C5 (64, 80, 80)
Conv+BN+ LeakyReLU (128, 80, 80)
Conv+BN+ LeakyReLU (128, 40, 40)
Concat (256, 40, 40)
C5 (128, 40, 40)
Conv+BN+ LeakyReLU (256, 40, 40)
Conv+BN+ LeakyReLU (256, 20,20)
Concat (512, 20, 20)
C5 (256, 20, 20)
Conv+BN+ LeakyReLU (512, 20, 20)
Conv2d ((5+n_cls)×3, 80, 80)
Conv2d ((5+n_cls)×3, 40, 40)
Conv2d ((5+n_cls)×3, 20, 20)
Detect0 (3, 80, 80, (n_cls+5))
Detect1 (3, 40, 40, (n_cls+5))
Detect2 (3, 20, 20, (n_cls+5))
(a)
(3, 640, 640)
Backbone
Conv+BN+ LeakyReLU (32, 320, 320)
Conv+BN+ LeakyReLU (64, 160, 160)
C5 (64, 160, 160)
MaxPool2d (64, 80, 80)
C5 (128, 80, 80)
MaxPool2d (128, 40, 40)
C5 (256, 40, 40)
MaxPool2d (256, 20, 20)
C5 (512, 20, 20)
Head
Concat (128, 80, 80)
Conv+BN+ LeakyReLU (64, 80, 80)
Upsample (64, 80, 80)
Conv+BN+ LeakyReLU (64, 40, 40)
C5 (128, 40, 40)
Concat (256, 40, 40)
Conv+BN+ LeakyReLU (128, 40, 40)
Upsample (128, 40, 40)
Conv+BN+ LeakyReLU (128, 20, 20)
CBC (256, 20, 20)
C5 (64, 80, 80)
RepConv (128, 80, 80)
Conv+BN+ LeakyReLU (128, 40, 40)
Concat (256, 40, 40)
C5 (128, 40, 40)
RepConv (256, 40, 40)
Conv+BN+ LeakyReLU (256, 20,20)
Concat (512, 20, 20)
C5 (256, 20, 20)
CBAM (512, 20, 20)
Conv2d ((5+n_cls)×3, 80, 80)
Conv2d ((5+n_cls)×3, 40, 40)
Conv2d ((5+n_cls)×3, 20, 20)
Detect0 (3, 80, 80, (n_cls+5))
Detect1 (3, 40, 40, (n_cls+5))
Detect2 (3, 20, 20, (n_cls+5))
(b)

Figure 7. The structure of the YOLOv7-tiny and the Improved YOLOv7-tiny models. (**a**) The structure of the YOLOv7-tiny model; (**b**) the structure of the Improved YOLOv7-tiny model.

In addition, we introduced SIoU [32] to improve the regression loss function of the original model. Based on the characteristics of the traditional regression box loss, SIoU introduces the angle cost to assist the distance calculation between the ground true box and the predicted box of the target to improve convergence speed and feature learning ability of model training. Specifically, the SIoU loss function includes four parts: angle, distance, shape, and IoU costs.

The angle cost describes the convergence process of the minimum angle of the ground true box and the predicted one, which can be expressed as Formula (5).

$$\wedge = 1 - 2 \times sin^2\left(\arcsin(x) - \frac{\pi}{4}\right),\ x = \frac{c_h}{\sigma} = sin(\alpha) \tag{5}$$

where $\wedge$ represents the angle cost; c_h denotes the difference of the two boxes center points in height; σ denotes the distance of the two boxes; and α denotes the radian of the angle of the line connecting the two boxes center points and the Cartesian coordinate x-axis. When α equals 0 or $\pi/2$, the angle loss obtains the minimum value of 0; when α equals $\pi/4$, the angle loss obtains the maximum value of 1.

The distance cost describes the convergence process of the minimum distance between the ground true and the predicted boxes, and the distance loss can be expressed mathematically by Formula (6).

$$\Delta = \sum\nolimits_{t=x,y}\left(1 - e^{-(2-\wedge)\rho_t}\right) \tag{6}$$

where

$$\rho_x = (\frac{b^{gt}_{c_x} - b^{p}_{c_x}}{bb_w})^2,\ \rho_y = (\frac{b^{gt}_{c_y} - b^{p}_{c_y}}{bb_h})^2 \tag{7}$$

where $(b^{gt}_{c_x}, b^{gt}_{c_y})$ and $(b^{p}_{c_x}, b^{p}_{c_y})$ denote the center coordinate points of the two boxes, respectively; and bb_w and bb_h denote the width and height of the bounding box of the two boxes, respectively. When combined with Formula (5), it can be seen that the distance cost correlates positively with the angle cost.

The shape cost describes the overall shape convergence of the minimum difference in border height and width between the ground true box and the predicted box. And the shape cost can be expressed by Formula (8).

$$\Omega = \sum\nolimits_{t=w,h} \left(1 - e^{-w_t}\right)^{\theta} \tag{8}$$

where

$$\mathrm{w_w} = \frac{\left|\mathrm{w^p} - \mathrm{w^{gt}}\right|}{\max\left(\mathrm{w^p}, \mathrm{w^{gt}}\right)},\ \mathrm{w_h} = \frac{\left|\mathrm{h^p} - \mathrm{h^{gt}}\right|}{\max\left(\mathrm{h^p}, \mathrm{h^{gt}}\right)} \tag{9}$$

where $\left(\mathrm{w^{gt}}, \mathrm{h^{gt}}\right)$ and $(\mathrm{w^p}, \mathrm{h^p})$ denote the width and height of the ground true and predicted boxes, respectively; θ is an adjustable variable to adjust the contribution of shape cost to the overall loss.

Therefore, the expression of the SIoU Loss function can be expressed by Formula (10), where IoU denotes the overlap ratio of the two boxes; Δ denotes the distance cost; and Ω denotes the shape cost.

$$Loss_{box} = 1 - IoU + \frac{\Delta + \Omega}{2} \tag{10}$$

2.3. The Comparison Methods Used in This Study

The classic two-stage object detection network Faster-RCNN was improved based on the R-CNN [33] and Fast-RCNN [34] networks. The process by which the algorithm achieved object detection comprised mainly two stages. In the first stage, the model used the layer stacking method of the VGGNet [35] network to perform feature extraction on the input image. In the second stage, it generated a series of region proposals on the last convolution output feature maps through the region proposal network (RPN); then the generated region proposals were mapped to the feature maps extracted by the convolutional layer through the region-of-interest pooling layer (Roi pooling); finally, the output proposed feature map was used for classification and position regression prediction by using a softmax classifier and the bounding box regression algorithm.

The Single Shot MultiBox Detector (SSD) is a one-stage object-detection algorithm proposed by Liu, W. et al. (2016) [36]. The model is mainly composed of three parts: the VGG-Base backbone, the Extra layers, and the Pred-layers. First, the features of the input image were extracted through the VGG-Base backbone. Then the extracted effective feature information was sent to the extra-layers with different down-sampling operations to obtain a plurality of feature maps with different scales to form a pyramidal feature map set. The extracted multi-scale feature information was then sent to the Pred-layers part of the SSD model for classification prediction and bounding box regression. Finally, the Non-Maximum Suppression (NMS) algorithm was used to filter the prediction result with the best score for the detection and location of the target object. In this study, we compared the Faster-RCNN and SSD models with the improved object detection networks and analyzed the detection effects of different types of object detection models on the rice diseases and insect pests datasets.

2.4. Development Platform for Rice Diseases and Insect Pests Identification Application

Mobile phones have great advantages as image acquisition terminals and devices for loading intelligent identification algorithms to detect rice diseases and insect pests quickly in the field. To verify the practicability of the improved detection models, we developed a mobile phone application to detect of diseases and pests, as shown in Figure 8. The specific development process and experimental platform of the system are as follows.

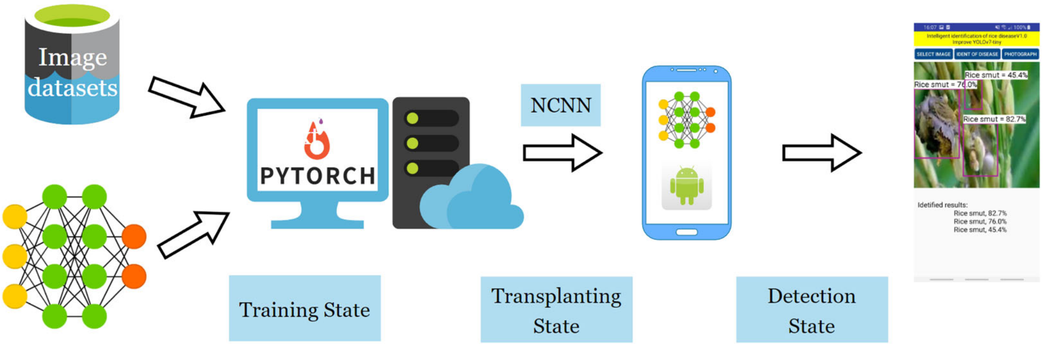


Figure 8. The specific development process of the rice diseases and insect pests identification application.

The experiment was first conducted on a 64-bit server computer with an Ubuntu 16.04.6 system. The configuration is shown in Table 2. The server was configured with two P100 graphics cards with 16G memory, and the system was installed with version 10.2 of Cuda and 7.6.5 Cudnn deep-learning environments. The deep-learning open-source library Pytorch was applied to train and test the object detection models. The NCNN framework was used to optimize the parameters and structure of the trained models, which enabled the models to be deployed and run on a mobile phone. NCNN is a high-performance neural network forward-computing framework suitable for mobile phone deployment, which enables the deep-learning model to achieve efficient reasoning on a mobile phone platform.

Table 2. Experimental platform.

Platform	System	Configuration	Framework/Architecture
Server platform for model training	Ubuntu 16.04.6	P100 16G × 2	Cuda 10.2, Cudnn 7.6.5, Pytorch
Application development platform	Android 9.0	Memory 4G	Android

In this study, the improved object detection models were transplanted to an Android mobile phone platform, and a rice disease and insect pest identification application was developed to verify the practicability of the models. The configuration of the platform is shown in Table 2. The mobile phone was installed with the Android 9.0 version system, and the running memory was 4 GB, which met the running and reasoning of the deep-learning model. The application included two parts: the front-end interaction interface and back-end program. The front-end interaction interface was composed mainly of the tools provided by the Android UI interface to design, including Button, ImgView, TextView. The back-end development processing achieved image-selection and photograph identification by calling the object detection model, which realized the real-time image collection and identification processing of the disease and insect pest image data in the field.

2.5. Evaluation Indicators

In this article, we evaluated constructed models from two aspects: object detection accuracy and model operational efficiency. The accuracy of the models was evaluated mainly using Precision, Recall, F1-score, mAP (0.5), and mAP (0.5:0.9), where mAP (0.5) was the AP value at an IoU threshold of 0.5, which was obtained by integrating the PR curve. mAP (0.5:0.9) was the average of the mAP values at different IoU thresholds. Then the evaluation indicators were as shown in Formulas (11–14), where TP (True Positive) represents the number of originally positive samples that were correctly predicted to be positive samples; FP (False Positive) represents the number of originally negative samples that were incorrectly predicted to be positive samples; FN (False Negative) represents the number of positive samples that were incorrectly predicted to be negative samples.

$$Precision = \frac{TP}{TP + FP} \tag{11}$$

$$Recall = \frac{TP}{TP + FN} \tag{12}$$

$$F1\ score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{13}$$

$$AP = \int_0^1 P(r)dr \tag{14}$$

The operational efficiency of the model was evaluated mainly by the parameter quantity, model size, FLOPs, and inference speed.

3. Results

3.1. Analysis of Experimental Results of Different Models

3.1.1. Analysis of Training Processes of Different Model

The models mentioned in Sections 2.2 and 2.3—Faster-RCNN, VGG16-SSD, YOLOv5s, and YOLOv7-tiny—were used for comparison with the Improved YOLOv5s model and the Improved YOLOv7-tiny models. For model training, we used the rice diseases and insect pests image dataset on the server computer. To achieve better training results, the SGD gradient descent algorithm with momentum was applied to update the gradient during model training, and the hyperparameter momentum was set to 0.937. The purpose of adding momentum gradient optimization was to suppress the oscillation generated by the gradient descent and accelerate the convergence of the model. The LambdaLR learning-rate adjustment strategy was adopted, and the initial learning rate and the weight decay were 0.01 and 0.0005, respectively. Regarding model evaluation, the NMS method was used to optimize the output of the model detection results, in which the NMS-confidence and IOU thresholds of the prediction results were set to 0.25 and 0.6, respectively.

For the training results of different models on the rice diseases and insect pests dataset, the training batch size was set to 32, and the model was trained for a total of 300 epochs. We set it to validate the models every 10 epochs in the training process to record the loss value and mAP (0.5) result on the validation set and save the model training files. From Figure 9a,b, the loss values of each model decreased continuously during training, and they tended to be stable after 300 iterations, indicating that the constructed rice diseases and pests detection models fit the data distribution of the training set samples well. The loss values of Improved YOLOv5s and Improved YOLOv7-tiny were slightly lower than those of the original models, and had better convergence effects. It can be seen from the mAP (0.5) results change curve of the validation set (Figure 9c) that the curves of the improved models were higher than those of the original models, which indicated better detection accuracy.

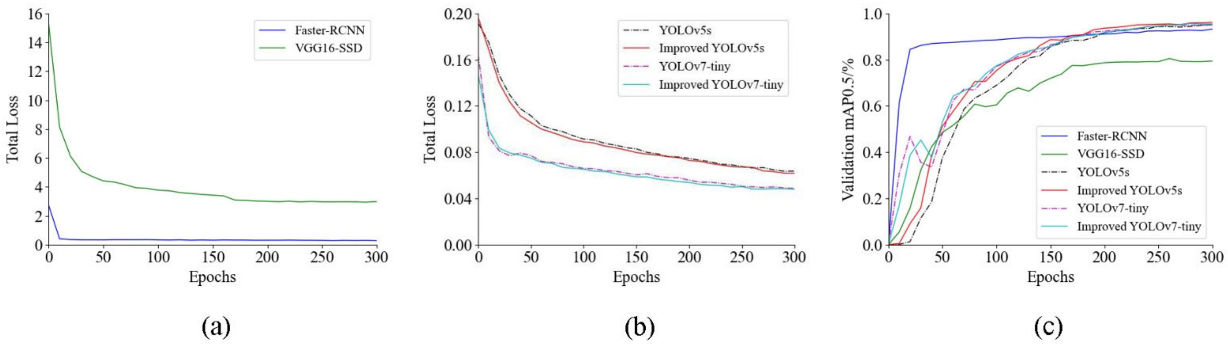


Figure 9. Change curves of loss value and mAP (0.5) of different models. (**a**) Loss value curve of Faster-RCNN and VGG16-SSD; (**b**) loss value curve of the improve models; (**c**) mAP (0.5) curve of different models.

3.1.2. Comparison of Detection Accuracy of Different Models

In this study, we used the test set of the image dataset constructed in Section 2.1 to test the detection accuracy of each model. From Table 3, it can be seen that the Improved YOLOv5s outperformed the original model YOLOv5s in all evaluation indicators. Its precision and recall value improved by 3.2 and 0.3%, respectively, while its F1-Score, mAP (0.5), and mAP (0.5:0.9) improved by 1.7, 0.9, and 3.7%, respectively, and it achieved the highest scores among the six experimental methods with values of 0.931, 0.961, and 0.648. Although the Improved YOLOv7-tiny had a 1.2% decrease in recall compared to the original model, it had improved precision, F1-Score, mAP (0.5), and mAP (0.5:0.9) by 3.5, 1.1, 0.3 and 1.1%, respectively. In addition, the Improved YOLOv7-tiny was second only to Improved YOLOv5s in F1-Score, mAP (0.5), and mAP (0.5:0.9), which are three comprehensive indicators for evaluating the overall identification performance of the model. Therefore, from the above analysis, the detection accuracy of the Improved YOLOv5s and Improved YOLOv7-tiny in this study was better than that of their original models.

Table 3. Comparison of detection accuracy indicators of different models.

Network	**Precision**	**Rcall**	**F1-Score**	**mAP**	
				IOU = 0.5	**IOU = 0.5:0.95**
Faster-RCNN	0.888	0.944	0.915	0.933	0.584
VGG16-SSD	0.806	0.939	0.872	0.820	0.476
YOLOv5s	0.908	0.923	0.915	0.952	0.625
YOLOv7-tiny	0.895	0.924	0.909	0.951	0.620
Improved YOLOv5s	0.937	0.926	0.931	0.961	0.648
Improved YOLOv7-tiny	0.926	0.913	0.919	0.954	0.627

3.2. Analysis of Experimental Results of Improved Models

3.2.1. Comparison of Operational Efficiency before and after Model Improvement

In this section, we evaluate the operational efficiency of the improved models using parameter quantity, model size, the FLOPs, and model inference speed. From Table 4 below, it can be seen that in the comparative analysis before and after model improvement the parameters, model size, and FLOPs of the Improved YOLOv5s decreased by 47.5, 45.7, and 48.7%, respectively, whereas the inference speed increased by 38.6% compared to the original model. Therefore, the application of the Ghost module significantly improved the structure of YOLOv5s, thereby greatly improving its operational efficiency. Compared to the original model, Improved YOLOv7-tiny had increased parameter quantity, model size, and the FLOPs by 0.5, 1.7, and 3.1%, respectively, whereas the inference speed decreased by 4.9%. Although the application of the CBAM module improved the accuracy of the model, it increased its complexity to some extent. In a comparative analysis of the performance of the two improved models, the parameter quantity, model size, and FLOPs of Improved YOLOv5s were all lower by 61, 62.9, and 60%, respectively. And the inference speed of Improved YOLOv5s was 18.6% faster. Therefore, it outperformed Improved YOLOv7-tiny in operational efficiency.

Table 4. Comparison of operation performance indicators of different models.

Network	Parameters/M	Model Size/MB	FLOPs/GFLOPs	Inference Speed/ms (b32)
YOLOv5s	7.03	13.8	15.8	11.4
YOLOv7-tiny	6.02	11.7	13.1	8.2
Improved YOLOv5s	3.69	7.49	8.1	7
Improved YOLOv7-tiny	6.05	11.9	13.5	8.6

3.2.2. Comparison of Different Type Rice Diseases and Insect Pests Detection Accuracy before and after Model Improvement

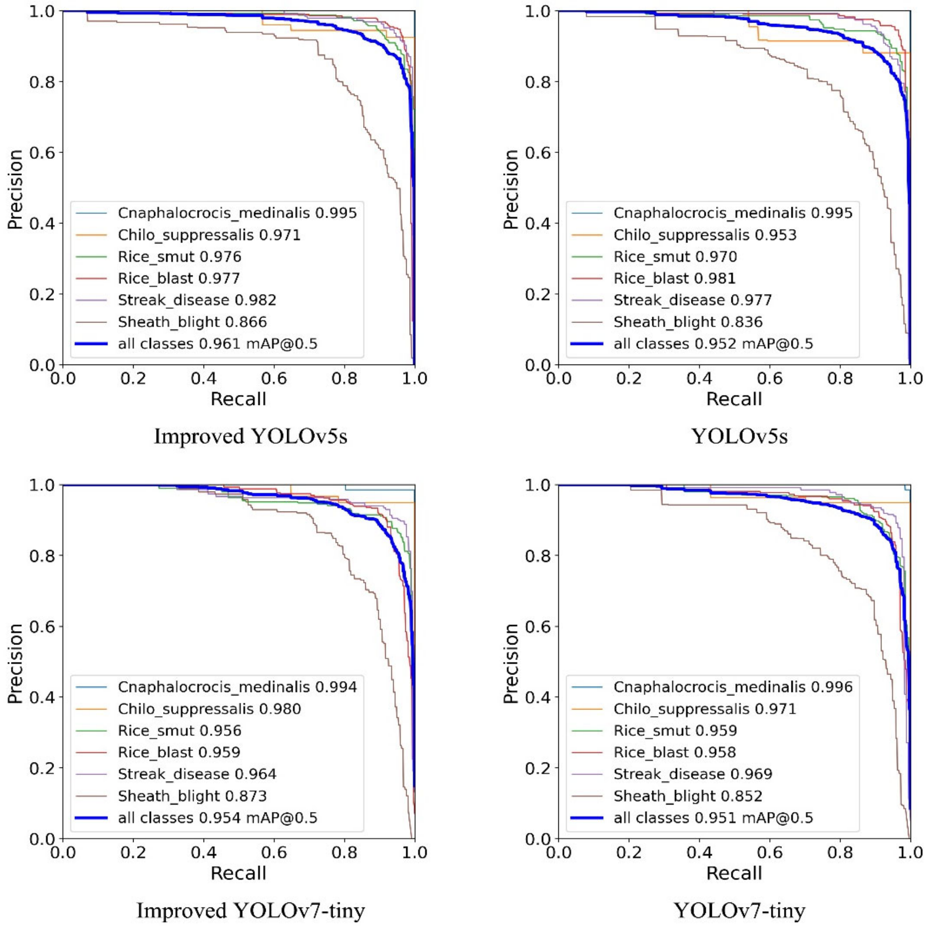
We also evaluated the detection accuracy of the improved models constructed in this study on different categories of rice diseases and insect pests images using average precision (AP) and PR curves. From the data in Table 5 below, it can be seen that the AP of Improved YOLOv5s on Chilo suppressalis, Rice smut, Streak disease, and Sheath light was 1.9, 0.6, 0.5, and 3.6% greater, respectively, than for the original model. The AP of Improved YOLOv7-tiny in Chilo suppressalis, Rice blast, and Sheath light increased by 0.9, 0.1, and 2.5%, respectively, compared to YOLOv7-tiny. For the comparison and analysis of the detection accuracy of the two improved models, the mAP (0.5) of Improved YOLOv5s reached 0.961, which was 0.7% higher than the mAP (0.5) value of 0.954 for Improved YOLOv7-tiny. The overall detection accuracy of Improved YOLOv5s for rice diseases and insect pests was better than that of Improved YOLOv7-tiny. Moreover, the PR curve shown in Figure 10 was used to evaluate model detection accuracy. The enclosed area of the curve was equal to the mAP (0.5). When the curve area was close to 1, the model detection accuracy was the highest.

3.2.3. Comparison of Detection Results of Different Type Rice Diseases and Insect Pests Images before and after Model Improvement

For validation of the detection effect of the improved object detection models, we randomly selected one image from each type of rice disease and insect pest in the test set to demonstrate the detection effect of each model. In Figure 11, the input and label images of different types of rice diseases and insect pests are shown in the first and second columns, while the detection results of these images and the corresponding predicted probability heat maps of Improved YOLOv5s and Improved YOLOv7 tiny are shown in the third to sixth columns. The figure shows that, for the detection results of the rice smut sample, Improved YOLOv5s had better detection accuracy for each target in the image compared to Improved YOLOv7-tiny, while Improved YOLOv7-tiny had a missed detection. For the detection results of the sheath light and Chilo suppressalis samples, the detection accuracy and identification effect of Improved YOLOv7-tiny were better, while the detection results of Improved YOLOv5s for these two types of diseases and insect pests revealed false and missing detection. For the other three types of disease and insect pest, the two improved models displayed little difference in detection accuracy and both effectively detected the target areas of diseases and pests in the image. In addition, the model identification effects were also analyzed through probability heat maps, in which the color change from red to blue indicated that the probability contribution of the region to the prediction results was gradually decreasing. From the heat map results, Improved YOLOv5s performed better in detecting larger target areas in images, as shown in the large red areas in the heat map results of streak disease and sheath blight samples; Improved YOLOv7 tiny was more accurate in detecting smaller target areas in images, such as the small red areas in the heat map results of rice smut and rice blast.

Table 5. Comparison of different type rice diseases and insect pests detection accuracy before and after model improvement.

Models	Average Precision						mAP (IOU = 0.5)
	Cnaphalocrocis Medinalis	Chilo Suppressalis	Rice Smut	Rice Blast	Streak Disease	Sheath Blight	
YOLOv5s	0.995	0.953	0.970	0.981	0.977	0.836	0.952
YOLOv7-tiny	0.996	0.971	0.959	0.958	0.969	0.852	0.951
Improved YOLOv5s	0.995	0.971	0.976	0.977	0.982	0.866	0.961
Improved YOLOv7-tiny	0.994	0.980	0.956	0.959	0.964	0.873	0.954



Improved YOLOv5s YOLOv5s

Improved YOLOv7-tiny YOLOv7-tiny

Figure 10. The PR curves before and after model improvement.

3.3. Analysis of Experimental Results of Rice Diseases and Insect Pests Detection Application on Mobile Phone

3.3.1. Identification Results of Rice Diseases and Insect Pests Detection Mobile Phone Application

For further validation of the practicality of the improved models proposed in this study, we transplanted Improved YOLOv5s and Improved YOLOv7-tiny to the mobile phone Android platform to build a rice disease and insect pest identification application. It was designed with image-selection and photo identification functions on which the two models were tested and analyzed using the above test set. In identification accuracy, from Table 6, when the model accuracy was FP16 on the mobile phone, the precision and recall of Improved YOLOv5s reached 0.925 and 0.939, respectively, which were 1.1 and 2.3% higher, respectively, than those of Improved YOLOv7-tiny. The inference speed of Improved YOLOv5s on the Android mobile phone platform was 374 ms/frame, which was 6.7% faster. In identification results, as shown in Figure 12, both models achieved accurate detection of rice diseases and insect pests in images. And Improved YOLOv5s outperformed Improved YOLOv7-tiny in the detection of rice smut, streak disease, and streak blight, which are three diseases and insect pests sample images that have larger target sizes (as shown in Figure 12a), while Improved YOLOv7-tiny had better detection for rice blast, Cnaphalocrocis medinalis, and Chilo suppressalis, which are three sample images with smaller target sizes (as shown in Figure 12b).

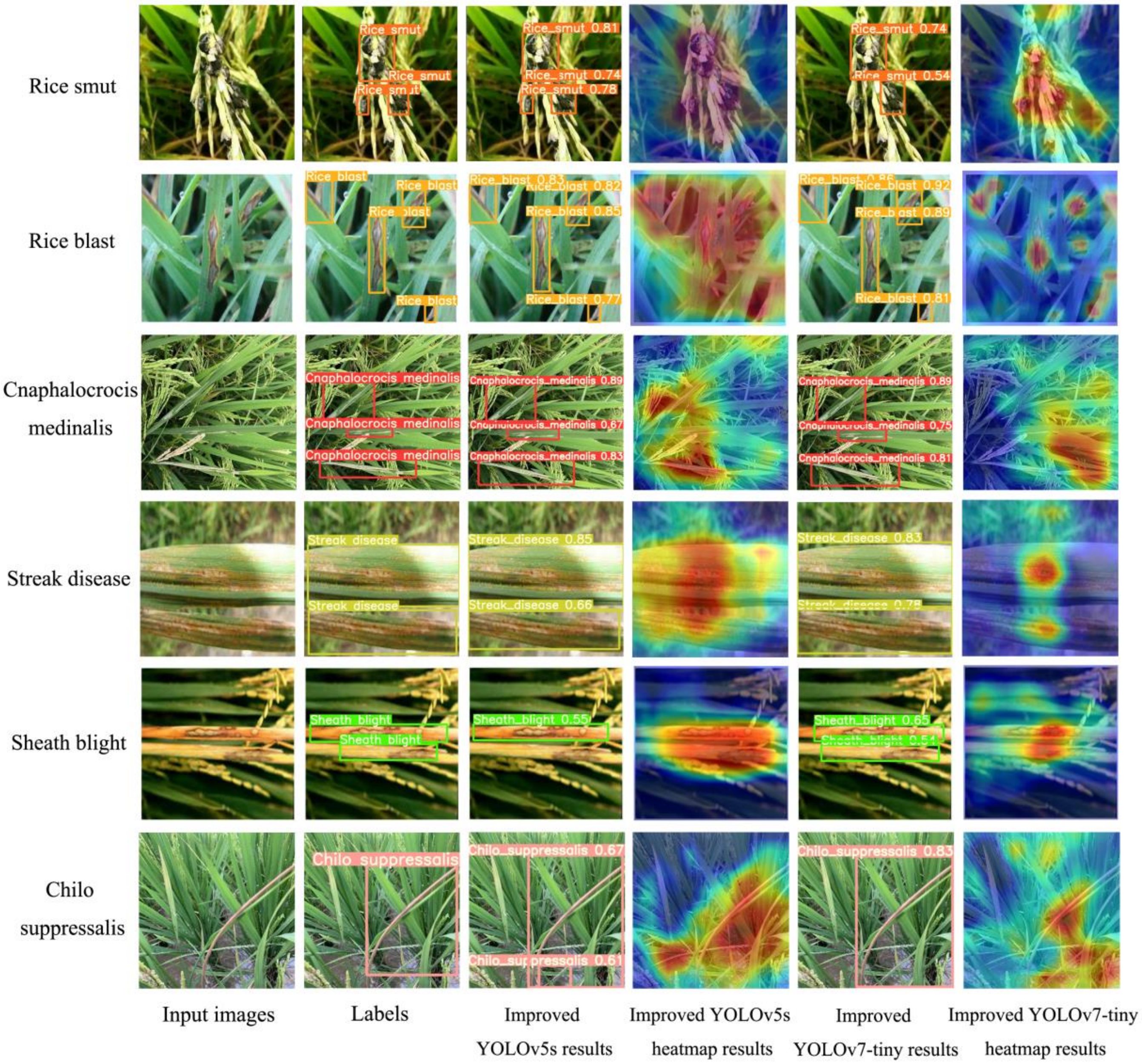
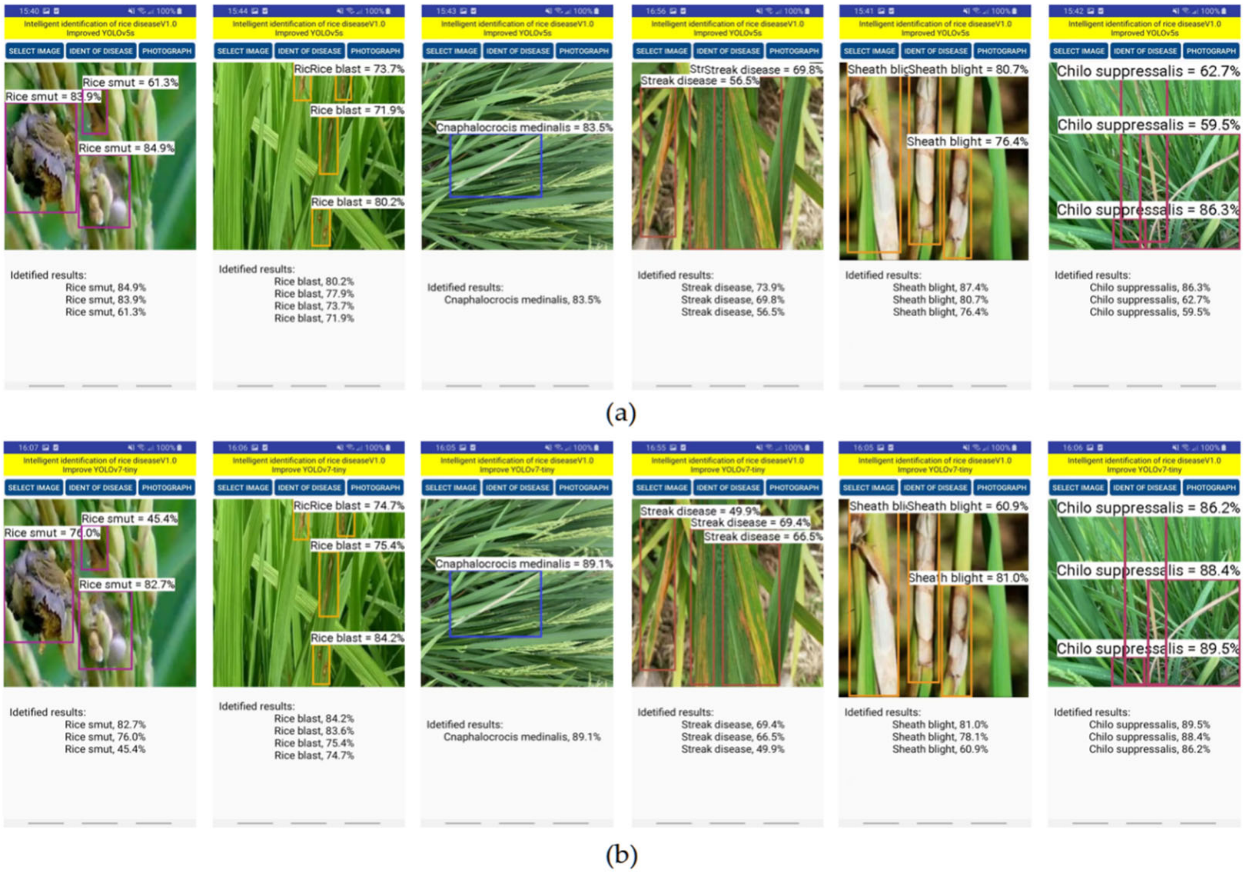


Figure 11. Detection Results of Different Type Rice Diseases and Insect Pests.

Table 6. Identification accuracy of two improved models on the mobile phone platform.

Network	Precision	Recall	Inference Speed/ms
Improved YOLOv5s	0.925	0.939	374
Improved YOLOv7-tiny	0.915	0.918	401



Idetified results:
Rice smut, 84.9%
Rice smut, 83.9%
Rice smut, 61.3%

Idetified results:
Rice blast, 80.2%
Rice blast, 77.9%
Rice blast, 73.7%
Rice blast, 71.9%

Idetified results:
Cnaphalocrocis medinalis, 83.5%

Idetified results:
Streak disease, 73.9%
Streak disease, 69.8%
Streak disease, 56.5%

Idetified results:
Sheath blight, 87.4%
Sheath blight, 80.7%
Sheath blight, 76.4%

Idetified results:
Chilo suppressalis, 86.3%
Chilo suppressalis, 62.7%
Chilo suppressalis, 59.5%

Idetified results:
Rice smut, 82.7%
Rice smut, 76.0%
Rice smut, 45.4%

Idetified results:
Rice blast, 84.2%
Rice blast, 83.6%
Rice blast, 75.4%
Rice blast, 74.7%

Idetified results:
Cnaphalocrocis medinalis, 89.1%

Idetified results:
Streak disease, 69.4%
Streak disease, 66.5%
Streak disease, 49.9%

Idetified results:
Sheath blight, 81.0%
Sheath blight, 78.1%
Sheath blight, 60.9%

Idetified results:
Chilo suppressalis, 89.5%
Chilo suppressalis, 88.4%
Chilo suppressalis, 86.2%

Figure 12. Identification results of rice diseases and insect pests detection mobile phone applications. (**a**) Identification results of Improved YOLOv5s; (**b**) identification results of Improved YOLOv7-tiny.

3.3.2. Runtime Performance of Rice Diseases and Insect Pests Detection Application on a Mobile Phone

To verify the applicability and compatibility of the improved models with mobile phone hardware, we evaluated the runtime performance of the application based on the improved detection models using indicators such as the model size after conversion with NCNN, the CPU usage, and the RAM usage. From Table 7, the results of model size, CPU usage, and RAM usage for Improved YOLOv5s on the phone were 14.3 MB, 49%, and 262.9 MB, respectively, which were 38, 5.8 and 17.4% less than for Improved YOLOv7-tiny. The performance of Improved YOLOv5s was better, but both models can be used for inference on the Android mobile phone platform because the model size, CPU usage, and RAM usage are within a reasonable range of computing resource allocation. Therefore, both proposed models can be applied to most Android mobile phone hardware platforms with similar or better configurations and performance than the platform in this study.

Table 7. Runtime results on the mobile phone platform of two improved models.

Network	Model Size/MB	CPU Usage/%	RAM Usage/MB
Improved YOLOv5s	14.3	49	262.9
Improved YOLOv7-tiny	23.1	52	318.2

4. Discussion

Previous studies have shown that CNN models can be applied to the image classification of rice diseases and insect pests and have achieved good results [37–39]. However, the classification method cannot accurately detect the specific occurrence areas of diseases and pests in the image. In addition to identifying the correct information category when diagnosing rice diseases and insect pests on actual farmland, detecting the occurrence areas of diseases and insect pests is also important. In this regard, the object detection method based on deep-learning used in our research proved to be a feasible method for achieving accurate classification and location detection of diseases and pests.

In this study, based on the object detection models YOLOv5 and YOLOv7, two lightweight rice disease and insect pest detection models, Improved YOLOv5s and Improved YOLOv7-tiny, were constructed to identify the categories and detect the occurrence locations of rice diseases and insect pests in images. From an analysis of the experimental results of different models, the two improved models outperformed their original models in F1-Score, mAP (0.5) and mAP (0.5:0.9), and Improved YOLOv5s achieved the highest score in these indicators compared to the other methods, which showed that the two improved methods had effectively improved detection accuracy. Regarding operational efficiency, Improved YOLOv5s had significantly improved parameter quantity, model size, FLOPs, and inference speed compared to the original model, indicating that the application of the Ghost module not only improved detection accuracy but also optimized model structure and enhanced the operational inference efficiency.

In addition, Improved YOLOv5s was better than Improved YOLOv7-tiny in various evaluation indicators of detection accuracy. Regarding identification of rice diseases and insect pests, Improved YOLOv5s had a higher AP value compared to Improved YOLOv7-tiny. In addition, the AP value of sheath blight was lower than that of the other five rice diseases and insect pests in the detection results of different models. There were two reasons. First, the original image data of this category of rice disease and insect pest were less than those of the other five categories, and the original features for model learning were relatively less, so it was difficult for the model to cover the complete feature values for this image category in fitting training, thereby causing an increase in model recognition error [40]. Second, although certain data enhancement methods reduced the impact of imbalanced distribution of data in various categories on model performance, the image features of the enhanced image data were similar to those of the real image data, so repeated learning of the same features resulted in overfitting. Therefore, when applied to new data, the accuracy and robustness of the model in that category were not as good as those in categories with rich data features [41]. Furthermore, through probability heat map analysis, Improved YOLOv5s performed better in detecting larger rice disease and insect pest target areas in images, while Improved YOLOv7-tiny performed more accurately in detecting smaller target areas in images. On the mobile phone with the Android platform, the applications combined with the improved models achieved accurate identification of rice diseases and insect pests, and the application based on Improved YOLOv5s performed better in detection accuracy, inference efficiency, and runtime performance. However, both models are compatible with most Android mobile phones with similar or better configurations and performance than the hardware platform in this study.

To sum up, both the proposed improved models can be applied to the task of rice disease and insect pest detection and achieved good identification results. The constructed mobile phone application for rice disease and insect pest detection provided a fast and convenient intelligent mobile terminal offline identification method that is worth promoting.

As for the prospect of the work, the improved models proposed in this study are the general and extensible methods for effectively identifying rice diseases and insect pests, which is not limited to the detection and identification of the six common rice diseases and insect pests mentioned above. If more identification of rice diseases and insect pests need to be conducted in subsequent research, it is only necessary to increase the categories and quantities of their image data and retrain the construction models. For disease and pest identification on the mobile phone platform, considering that detection results are affected by shooting angle, distance, and ambient light of the phone, we can adapt the model to different scenarios and improve generalization performance by expanding the image dataset and using a variety of different image augmentation strategies to improve the detection and identification accuracy.

5. Conclusions

1. We proposed two rice disease and insect pest detection models suitable for mobile phone terminals based on deep-learning detection and realized offline detection by intelligent mobile phone terminals, thus providing an efficient and reliable intelligent detection method for farmers and plant protection personnel. By introducing the Ghost module, Improved YOLOv5s significantly improved detection accuracy and operation efficiency compared to YOLOv5s. It had the highest F1-Score and the highest scores for mAP (0.5) and mAP (0.5:0.9) with values of 0.931, 0.961 and 0.648, respectively. Moreover, the parameter quantity, model size, and FLOPs of Improved YOLOv5s were reduced by 47.5, 45.7, and 48.7%, respectively, while the inference speed improved by 38.6% and model detection accuracy and operation efficiency also significantly improved. By introducing the CBAM attention module and SIoU loss, Improved YOLOv7-tiny outperformed YOLOv7-tiny in detection accuracy, and the model accuracy of F1-Score, mAP (0.5), and mAP (0.5:0.9) were second only to those of Improved YOLOv5s.
2. For the detection of different categories of rice diseases and insect pests, the AP value of Improved YOLOv5s was higher than that of Improved YOLOv7-tiny for the identification of all rice diseases and insect pests. The probability heat maps showed that Improved YOLOv5s had better detection in areas of rice disease and insect pests where there were larger image target sizes, and Improved YOLOv7-tiny had better detection accuracy where there were smaller image target sizes.
3. The two improved models were transplanted to an Android mobile phone. Under FP16, the precision and recall of Improved YOLOv5s was 0.925, and 0.939, and the inference speed was 374 ms/frame. Model accuracy, operational efficiency, and runtime performance were better than for Improved YOLOv7-tiny. The mobile phone application constructed by the improved models is compatible with most Android mobile phone hardware platforms for achieving accurate detection of rice diseases and insect pests.

The improved object detection models proposed in this study can realize the accurate detection of rice diseases and insect pests, and the mobile phone application for rice disease and insect pest identification can provide strong support for the rapid diagnosis and intelligent identification of rice diseases and insect pests in the field.

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

Funding: This research was funded by the Guangdong Provincial Innovation Team for General Key Technologies in Modern Agricultural Industry, grant number 2022KJ133, the Laboratory of Lingnan Modern Agriculture Project, grant number NT2021009, and the 111 Project grant number D18019.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to the privacy policy of the organization.

Acknowledgments: Many thanks to South China Agricultural University plant protection experts Maoxin Zhang and Guohui Zhou for helping us to carry out an investigation in the field of the study site and for giving professional guidance in rice diseases and insect pests image data annotation.

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

References

1. Liu, T.; Zhong, X.C.; Sun, C.M.; Guo, W.S.; Chen, Y.Y.; Sun, J. Recognition of Rice Leaf Diseases Based on Computer Vision. *Sci. Agric. Sin.* **2014**, *47*, 664–674. (In Chinese)
2. Wen, C.; Guyer, D.; Li, W. Local feature-based identification and classification for orchard insects. *Biosyst. Eng.* **2009**, *104*, 299–307. [CrossRef]
3. Wen, C.; Guyer, D. Image-based orchard insect automated identification and classification method. *Comput. Electron. Agric.* **2012**, *89*, 110–115. [CrossRef]
4. Liu, T.; Chen, W.; Wu, W.; Sun, C.; Guo, W.; Zhu, X. Detection of aphids in wheat fields using a computer vision technique. *Biosyst. Eng.* **2016**, *141*, 82–93. [CrossRef]
5. Xie, C.; Zhang, J.; Li, R.; Li, J.; Hong, P.; Xia, J.; Chen, P. Automatic classification for field crop insects via multiple-task sparse representation and multiple-kernel learning. *Comput. Electron. Agric.* **2015**, *119*, 123–132. [CrossRef]
6. Espinoza, K.; Valera, D.L.; Torres, J.A.; Lopez, A.; Molina-Aiz, F.D. Combination of image processing and artificial neural networks as a novel approach for the identification of Bemisia tabaci and Frankliniella occidentalis on sticky traps in greenhouse agriculture. *Comput. Electron. Agric.* **2016**, *127*, 495–505. [CrossRef]
7. Wang, Z.; Chu, G.K.; Zhang, H.J.; Liu, S.; Huang, X.; Gao, F.; Zhang, C.; Wang, J. Identification of diseased empty rice panicles based on Haar-like feature of UAV optical image. *Trans. Chin. Soc. Agric. Eng.* **2018**, *34*, 73–82. (In Chinese)
8. Wang, D.F.; Wang, J. Crop disease classification with transfer learning and residual networks. *Trans. Chin. Soc. Agric. Eng.* **2021**, *37*, 199–207. (In Chinese)
9. Li, X.Z.; Ma, B.X.; Yu, G.W.; Chen, J.; Li, Y.; Li, C. Surface defect detection of Hami melon using deep learning and image processing. *Trans. Chin. Soc. Agric. Eng.* **2021**, *37*, 223–232. (In Chinese)
10. Hou, J.L.; Fang, L.F.; Wu, Y.Q.; Li, Y.; Xi, R. Rapid recognition and orientation determination of ginger sprouts based on deep learning. *Trans. Chin. Soc. Agric. Eng.* **2021**, *37*, 213–222. (In Chinese)
11. Agarwal, M.; Singh, A.; Arjaria, S.; Sinha, A.; Gupta, S. ToLeD: Tomato leaf disease detection using convolution neural network. *Proc. Comput. Sci.* **2020**, *167*, 293–301. [CrossRef]
12. Zhong, Y.; Zhao, M. Research on deep learning in apple leaf disease recognition. *Comput. Electron. Agric.* **2020**, *168*, 105146. [CrossRef]
13. Ramcharan, A.; Baranowski, K.; McCloskey, P.; Ahmed, B.; Legg, J.; Hughes, D.P. Deep Learning for Image-Based Cassava Disease Detection. *Front. Plant Sci.* **2017**, *8*, 1852. [CrossRef]
14. Rahman, C.R.; Arko, P.S.; Ali, M.E.; Khan, M.A.I.; Apon, S.H.; Nowrin, F.; Wasif, A. Identification and recognition of rice diseases and pests using convolutional neural networks. *Biosyst. Eng.* **2020**, *194*, 112–120. [CrossRef]
15. Joshi, P.; Das, D.; Udutalapally, V.; Pradhan, M.K.; Misra, S. RiceBioS: Identification of Biotic Stress in Rice Crops Using Edge-as-a-Service. *IEEE Sens. J.* **2022**, *22*, 4616–4624. [CrossRef]
16. Deng, R.L.; Tao, M.; Xing, H.; Yang, X.L.; Liu, C.; Liao, K.; Qi, L. Automatic diagnosis of rice diseases using deep learning. *Plant Sci.* **2021**, *12*, 701038. [CrossRef]
17. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, 91–99. Available online: https://arxiv.org/abs/1506.01497 (accessed on 1 February 2023). [CrossRef]
18. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788. Available online: https://arxiv.org/abs/1506.02640 (accessed on 1 February 2023).
19. YOLOv5. Available online: https://github.com/ultralytics/yolov5 (accessed on 1 February 2023).
20. Lin, T.Y.; Dollar, P.; Girshick, R.; He, K.M.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017. Available online: https://arxiv.org/abs/1612.03144v2 (accessed on 1 February 2023).
21. Liu, S.; Qi, L.; Qin, H.F.; Shi, J.P.; Jia, J.Y. Path Aggregation Network for Instance Segmentation. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018. Available online: https://arxiv.org/abs/1803.01534v4 (accessed on 1 February 2023).
22. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient convolutional neural networks for mobile vision applications. *IEEE Access.* **2017**, *6*, 1–14.
23. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.-C. MobileNet V2: Inverted residuals and linear bottlenecks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520. Available online: https://arxiv.org/1801.04381 (accessed on 1 February 2023).
24. Howard, A.; Sandler, M.; Chu, G.; Chen, L.-C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. Searching for MobileNetV3. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019. Available online: https://arxiv.org/abs/1905.02244 (accessed on 1 February 2023).
25. 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 (CVPR), Salt Lake City, UT, USA, 18–22 June 2018.

26. Ma, N.N.; Zhang, X.; Zheng, H.T.; Sun, J. ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018. Available online: https://arxiv.org/abs/1807.11164 (accessed on 1 February 2023).
27. Han, K.; Wang, Y.; Tian, Q.; Guo, J.; Xu, C. GhostNet: More Features from Cheap Operations. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2019. Available online: https://arxiv.org/abs/1911.11907 (accessed on 1 February 2023).
28. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
29. Wang, C.Y.; Bochkovskiy, A.; Liao, H. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022. Available online: https://arxiv.org/abs/2207.02696 (accessed on 1 February 2023).
30. Woo, S.; Park, J.; Lee, J.-Y.; Kweon, I.S. *CBAM: Convolutional Block Attention Module*; Springer: Cham, Switzerland, 2018; pp. 3–19.
31. Ding, X.H.; Zhang, X.Y.; Ma, N.N.; Han, J.; Ding, G.; Sun, J. RepVGG: Making VGG-style ConvNets Great Again. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022. Available online: https://arxiv.org/2101.03697 (accessed on 1 February 2023).
32. Gevorgyan, Z. SIoU Loss: More Powerful Learning for Bounding Box Regression. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022. Available online: https://arxiv.org/abs/2205.12740 (accessed on 1 February 2023).
33. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
34. Girshick, R. Fast R-CNN. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1440–1448.
35. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *Comput. Sci.* **2014**. Available online: https://arxiv.org/abs/1409.1556 (accessed on 1 February 2023).
36. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. SSD: Single shot multibox detector. *arXiv* **2016**, arXiv:1512.02325.
37. Jiang, F.; Lu, Y.; Chen, Y.; Cai, D.; Li, G.F. Image recognition of four rice leaf diseases based on deep learning and support vector machine. *Comput. Electron. Agric.* **2020**, *179*, 105824. [CrossRef]
38. Lu, Y.; Yi, S.J.; Zeng, N.Y.; Liu, Y.R.; Zhang, Y. Identification of rice diseases using deep convolutional neural networks. *Neurocomputing* **2017**, *267*, 378–384. [CrossRef]
39. Wang, Y.B.; Wang, H.F.; Peng, Z.H. Rice diseases detection and classification using attention based neural network and bayesian optimization. *Expert Syst. Appl.* **2021**, *178*, 114770. [CrossRef]
40. Johnson, J.M.; Khoshgoftaar, T.M. Survey on deep learning with class imbalance. *J. Big Data* **2019**, *6*, 27. [CrossRef]
41. Shorten, C.; Khoshgoftaar, T.M. A survey on Image Data Augmentation for Deep Learning. *J. Big Data* **2019**, *6*, 60. [CrossRef]

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.