applied sciences

MDPI

Article

Study on Lightweight Model of Maize Seedling Object Detection Based on YOLOv7

Kai Zhao, Lulu Zhao, Yanan Zhao and Hanbing Deng *

College of Information and Electrical Engineering, Shenyang Agricultural University, Shenyang 110866, China; syauzhaokai@163.com (K.Z.); zhaolulu_zll@163.com (L.Z.); 15589766261@163.com (Y.Z.)
* Correspondence: denghanbing@syau.edu.cn

Abstract: Traditional maize seedling detection mainly relies on manual observation and experience, which is time-consuming and prone to errors. With the rapid development of deep learning and object-detection technology, we propose a lightweight model LW-YOLOv7 to address the above issues. The new model can be deployed on mobile devices with limited memory and real-time detection of maize seedlings in the field. LW-YOLOv7 is based on YOLOv7 but incorporates GhostNet as the backbone network to reduce parameters. The Convolutional Block Attention Module (CBAM) enhances the network's attention to the target region. In the head of the model, the Path Aggregation Network (PANet) is replaced with a Bi-Directional Feature Pyramid Network (BiFPN) to improve semantic and location information. The SIoU loss function is used during training to enhance bounding box regression speed and detection accuracy. Experimental results reveal that LW-YOLOv7 outperforms YOLOv7 in terms of accuracy and parameter reduction. Compared to other object-detection models like Faster RCNN, YOLOv3, YOLOv4, and YOLOv5l, LW-YOLOv7 demonstrates increased accuracy, reduced parameters, and improved detection speed. The results indicate that LW-YOLOv7 is suitable for real-time object detection of maize seedlings in field environments and provides a practical solution for efficiently counting the number of seedling maize plants.

Keywords: YOLOv7; seedling maize; detection model; lightweight; attention models

check for updates

Citation: Zhao, K.; Zhao, L.; Zhao, Y.; Deng, H. Study on Lightweight Model of Maize Seedling Object Detection Based on YOLOv7. *Appl. Sci.* **2023**, *13*, 7731. https://doi.org/10.3390/app13137731

Academic Editor: Stéfano Frizzo Stefenon

Received: 8 June 2023
Revised: 27 June 2023
Accepted: 28 June 2023
Published: 29 June 2023

1. Introduction

Maize is a strategic crop with the largest planting area and production in China. It provides an important guarantee for food security [1]. Rapid calculation of the seedling-emergence rate during seedling stage is crucial for predicting maize yield [2]. The traditional method is to manually calculate in the field, which requires a huge labor cost. To solve this problem, researchers attempt to use visual sensors and computer vision methods to achieve object detection. Yu et al. [3] proposed a new crop-segmentation method (AP-HI) based on computer vision using spatial distribution characteristics to judge whether crops have reached the emergence stage. The average accuracy rate of AP-HI is 96.68%, which is higher than other detection algorithms by 2.37%. Zhao et al. [4] employed the conventional Otsu thresholding approach to segment rapeseed plant objects. The average relative error is 6.83%, while R2 is 84.6%. Xia et al. [5] proposed a cotton-overlapping plant identification and counting method based on SVM and the maximum likelihood classification method that achieved an accuracy rate of 91.13%. However, using traditional object-detection methods to identify maize seedlings, the detection accuracy is affected seriously by the background of images. The error rate of object detection will increase with the complexity of the background (weeds, light, planting density, etc.).

With the development of deep learning, methods based on Deep Convolutional Neural Networks (DCNN) have gradually replaced traditional methods in the field of object detection. In DCNN, parameters of a multi-layer network are fitted to the features, which can be approximated as expressions by functions. Multi-layer network can be used to extract the features of data and realize the positioning and classification of the objects.

Object-detection models based on DCNN can be divided into two types: single stage and two stage. The two-stage object detection model obtains candidate boxes through an additional method (Selective Search [6], RPN [7], etc.) and inputs each candidate region into the deep network to extract features. Classification and boundary regression are performed for each candidate region. Typical two-stage object-detection models include R-CNN [8], Fast R-CNN [9], Faster R-CNN [10], etc.

Now, two-stage object-detection models have been widely used in the field of smart agriculture. Pan et al. [11] presented an enhanced version of the Faster R-CNN method to automatically detect and count sugarcane seedlings with the aid of aerial photography. The sugarcane detector had an average accuracy of 93.67%, and the average accuracy after the use of the seedling deduplication algorithm is 96.83%. Yang et al. [12] introduced Mask-RCNN in strawberry robot detection to visualize the positioning of strawberry picking points. The average detection accuracy is 95.78%, and the recall is 95.41%. Li and colleagues [13] introduced an automatic detection approach for hydroponic lettuce seedlings using the improved Faster R-CNN model. Using the aforementioned enhanced method, the hydroponic lettuce seedling detection accuracy was determined to average 86.2%, surpassing that of other detectors such as RetinaNet, SSD, Cascade RCNN, and FCOS. As reported in their study, Wang and colleagues [14] suggested a Faster RCNN-based approach for detecting various types of tomato diseases. Their experiments demonstrated that this method can effectively identify and differentiate different categories of tomato diseases with high accuracy. Jiang and colleagues [15] proposed a method for counting cotton seedlings in field environments that involves using the Faster R-CNN model with an Inception ResNetv2 feature extractor and a Kalman filter.

Although two-stage object-detection models have achieved good results in the fields of smart agriculture, there are still many problems remaining. Firstly, two-stage detection models are complex in structure and have more parameters than single-stage models that make models generally unable to be deployed on mobile devices. Secondly, two-stage detection models usually have a slow detection speed, which is limited by candidate box-generation algorithms. They fail to meet the requirements of real-time detection.

Due to the issues of two-stage models, single-stage object-detection models are gradually receiving more attention from researchers. In single-stage object-detection models, data are input directly into the backbone network for feature extraction. Compared to two-stage models, the prediction accuracy of single-stage models may be affected due to the generation mode of prediction boxes. However, single-stage models have no feature redundancy extraction and information transmission bottleneck, while the processing speed is faster than two-stage object-detection models. Typical single-stage object-detection models mainly include SSD [16], RetinaNet [17], EfficientDet [18], YOLO [19–23], etc. Especially with the continuous updates of YOLO models, single-stage object-detection has achieved excellent detection results in the agricultural fields. In their study, Yang and colleagues [24] introduced a novel approach for real-time pest detection using DCNN. Their work demonstrated that this method could achieve higher accuracy with greater efficiency, as well as reduced computational demands. Sekharamantry and colleagues [25] proposed a deep-learning scheme for detecting apples in apple farms. Their findings reveal that this method achieved high accuracy, with an accuracy rate of 97%, a recall rate of 99%, and an F0 score of 98.1%. Zhou et al. [26] introduced ConvNext and transformer to design the C3CNTR module. The experimental results showed that on the MAR20 dataset, the mAP increased by 3.5%. Li et al. [27] proposed a lightweight convolutional neural network. The new network model can achieve a much smaller model size and faster detection speed. Gao et al. [28] proposed a new method based on YOLOv4. The new method has high recognition accuracy, fast recognition speed, and low model complexity. Liu [29] and colleagues proposed a maize weed detection method based on YOLOv4-tiny. The new method introduced an attention mechanism module and a spatial pyramid pooling structure. The experiment results indicate that the mAP value was 86.69% and the detection speed was 57.33 f/s. Kaya [30] and their colleagues designed a novel method for detecting

place diseases. The novel method developed a multi-headed DenseNet-based architecture and improved detection accuracy through image fusion. The improved method achieved an average accuracy, recall, precision, and f1 score of 98.17%, 98.17%, 98.16%, and 98.12%, respectively. Zhao et al. [31] proposed a convolutional neural network based on inception and residual structure. The experiment results indicate that the overall accuracy is 99.55% in identifying three diseases of corn, potato, and tomato. Song et al. [32] proposed a corn tassel pose estimation method based on computer vision and directional object detection. The evaluation metrics indicate that the proposed method has a correct estimation rate of 88.56% and 29.57 Giga floating-point operations (GFLOPs).

Although YOLO has achieved great success in various domains, there is little research on crop seedling detection. In the field of smart agriculture, research primarily focuses on pest and disease detection, as well as the detection of fruits and vegetables, while seedling detection has received limited attention. Moreover, there is a lack of publicly available datasets specifically designed to train deep-learning models for maize seedling detection. The complex background of maize seedlings in real environments presents a significant challenge for accurate detection using deep-learning models. Furthermore, the large number of parameters in the model contribute to slow inference speed and excessive memory usage. To address these challenges, we have collected a substantial amount of image data from field environments. Our focus has been on reducing the model parameters in the backbone network, enhancing the performance of the feature fusion network, and resolving the issue of position loss. By making improvements to the YOLOv7 model, we have enhanced its capability and efficiency in maize seedling detection. These efforts have been aimed at overcoming the limitations posed by the lack of datasets, the complexity of the background, and the computational demands of deep-learning models.

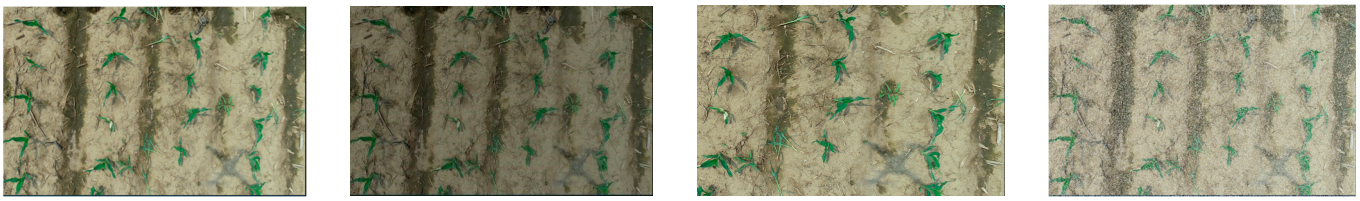
2. Materials and Methods

2.1. Data Acquisition, Augmentation, and Annotation

In this research paper, a dataset of maize seedling images was collected in the northeastern region of China, specifically using the Xianyu 335 maize variety. The images were acquired using a Dajiang drone (4 RTK) and had an original size of 5472 × 3648 pixels. The image acquisition period spanned from the emergence stage to the jointing stage of maize growth. During the data collection process, the drone followed a predetermined flight path and captured images from four different flight heights: 1.6 m, 2 m, 3 m, and 5 m, respectively. The camera was positioned to capture top-view images, and approximately 250 images were collected at each height. As a result, the total number of original images in the dataset amounted to 1000.

To mitigate the risk of overfitting during the training process [33], we increase training samples through data augmentation. General data-augmentation methods include Coutout [34], Random Erasing [35], Mixup [36], Mosaic [37], salt and pepper noise, etc. Considering the factors such as environment complexity, plant morphology, and planting density, Coutout and Random Erasing will result in a decrease in the number of positive samples that is not suitable for small objects. Therefore, we use augmentation methods with random brightness, cropping, and salt and pepper noise to reduce the loss of positive sample. The effect of data augmentation is shown in Figure 1. The total number of images after data augmentation is 2000; we divide all images into 8:1:1 rate and obtain 1600 training images, 200 validation images, and 200 test images, respectively. The distributions of the dataset are shown in Table 1.

During the data annotation process, the open-source tool LabelImg was utilized, as depicted in Figure 2. Each image was labeled using a single-category bounding-box format. In the case of top-view maize seedlings, each seedling was assigned a corresponding bounding box, ensuring that all the pixels of the seedling were fully encompassed within the rectangular area.



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

Figure 1. Data-augmentation methods. (**a**) Original image; (**b**) Random brightness; (**c**) Cropping; (**d**) Salt and pepper noise.

Table 1. Distributions of the dataset.

	Number of Original Images	Number of Random Brightness Images	Number of Cropping Images	Number of Salt and Pepper Noise Images	Number of Total Images
Training	1000	175	255	170	1600
Validation	100	32	30	38	200
Test	120	27	21	32	200

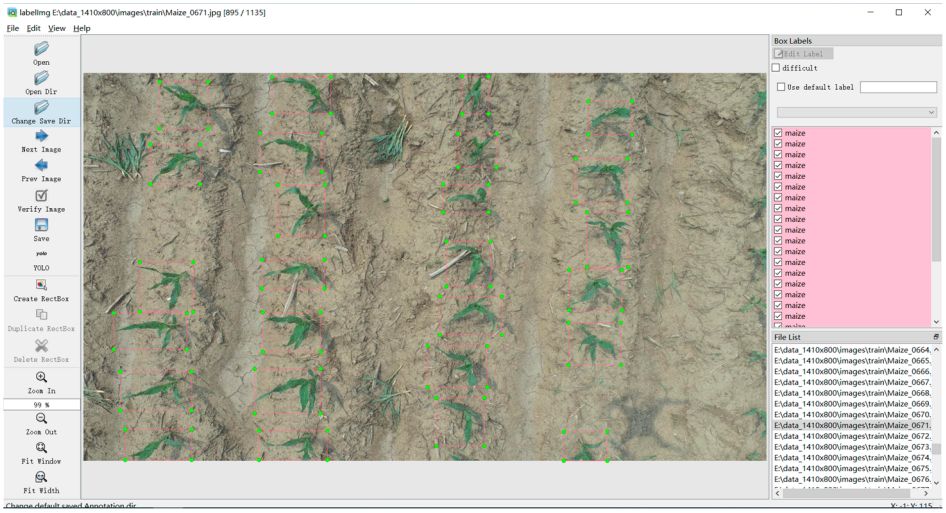


Figure 2. Labeling and annotations of maize seedlings.

2.2. YOLOv7

In the field of object detection, YOLO has always been one of the most popular deep-learning models. YOLO adopts a single neural network structure that divides the entire image into multiple grids and predicts multiple bounding boxes for each grid, which contains object positions and class information. Therefore, for each bounding box, YOLO predicts four coordinate values that represent the coordinates of the upper-left and lower-right corners of the bounding box, as well as the probabilities of belonging to different categories. Since this prediction process involves regression calculation between input data and model parameters, it can be considered a regression problem. Since 2016, YOLO has released multiple versions, each with different improvements and optimizations. In this paper, we utilized YOLOv7 [38], which is considered to be the most stable and reliable among the released versions. Compared to other YOLO models, YOLOv7 has significantly improved in both detection accuracy and speed during the Maize dataset.

Building on its predecessor, YOLOv7 innovatively introduces the extended ELAN architecture that improves the network's self-learning capability without destroying the original gradient path. ELAN is mainly composed of VoVNet combined with CSPNet and optimizes the gradient length of the overall network with the structure of stack in the computational block. By optimizing gradient paths, deeper networks can effectively learn

and converge. In addition, YOLOv7 incorporates a cascade-based model scaling method, which dynamically adjusts the model size to suit the specific detection requirements. The main purpose of model scaling is to adjust some attributes of the model and generate models at different scales to meet the needs of different inference speeds. For example, the scaling model of EfficientDet [39] considers width, depth, and resolution. As for scaled YOLOv4 [40], its scaling model adjusts the number of stages. However, the above methods are mainly used in PlainNet and ResNet. Therefore, it is necessary to propose corresponding composite model scaling methods for cascaded models. In the YOLOv7 model, when we scale the depth factor of a calculation block, we must also calculate the changes in the output channel of that block. Then, we will scale the width factor of the transition layer by an equal amount of variation. The method can maintain the characteristics of the model during the initial design and maintain the optimal structure. These methods ensure that the model is optimized for the task at hand, further enhancing the effectiveness and performance of YOLO. The research paper adopts YOLOv7 as the baseline model and carries out further optimizations to enhance its performance. The main structure of YOLOv7 consist of three key components: input, backbone, and head. These components work together to enable efficient and accurate object detection. The network structure of YOLOv7 is shown in Figure 3.

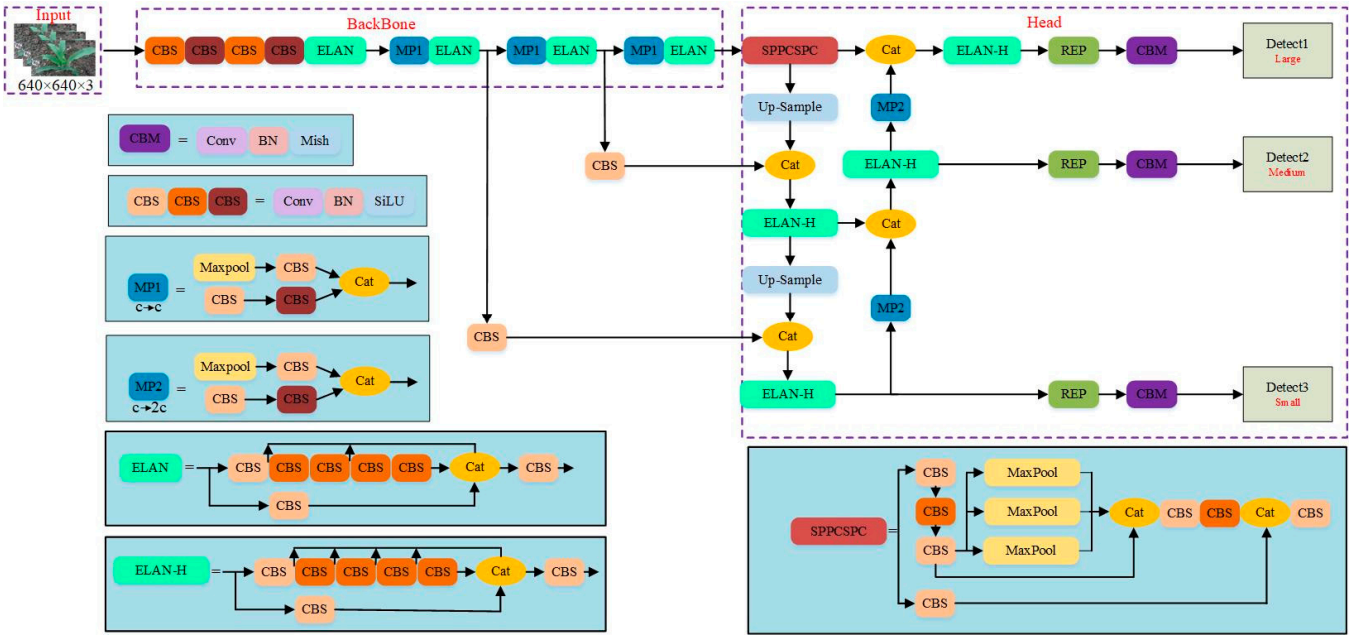


Figure 3. Model structure of YOLOv7.

The Input component of YOLOv7 incorporates two key elements: adaptive scaling and adaptive anchor box. Adaptive scaling is primarily used to adjust the size of the input image. This approach offers several advantages. Firstly, it can save memory when dealing with large-size images. Secondly, it enables the network to adapt to input images of varying sizes, thereby enhancing the model's generalization capability. Finally, adaptive scaling contributes to improved detection accuracy by ensuring that small objects occupy a more significant portion of the image, leading to better object localization. The adaptive anchor box is responsible for automatically selecting the number and size of prior boxes. By adjusting to different object scales and aspect ratios, the adaptive anchor box enhances the accuracy of object detection during testing. Moreover, it offers flexibility in accommodating diverse scenarios and tasks through the utilization of the K-means algorithm. This adaptability allows the model to adjust to different object scales and aspect ratios, improving its overall performance in various detection scenarios.

The backbone component of YOLOv7 is responsible for feature extraction and consists of three modules: CBS (Conv-BN-SiLU), ELAN (Extended Latent Attention Network), and MP (Max-Pooling). The CBS module comprises a sequence of layers, including Convolutional (Conv) layers, Batch Normalization (BN) layers, and SiLU (Sigmoid Linear Unit) layers. It employs three different convolutional kernel sizes and step sizes, allowing it to capture features at various scales. The CBS module plays a crucial role in extracting informative features from the input data. The ELAN module is an efficient network structure designed to control the shortest and longest gradient paths within the network. By doing so, it encourages the network to learn more diverse and discriminative features, resulting in improved robustness. The ELAN module enhances the model's capability to extract meaningful representations from the input data. The MP module consists of two branches for down-sampling. It utilizes max-pooling operations to reduce the spatial dimension of the feature maps, effectively capturing essential information at different scales. Together, the CBS, ELAN, and MP modules within the backbone component of YOLOv7 work harmoniously to extract relevant features from the input data, enabling accurate and efficient object detection.

The head component is responsible for further processing the extracted features and performing object detection. It consists of several modules: Spatial Pyramid Pooling (SPPCSPC), Feature Pyramid Network (FPN), Path Aggregation Network (PANet), and the detection heads. The SPPCSPC module adapts to images of different resolutions by obtaining different receptive fields with maximum pooling. The FPN and PANet enhanced the ability of network to integrate different feature layers. RepVGG is introduced to the head for training and to achieve recognition and classification of images. YOLOv7 has three detection heads, which are used to detect different sizes.

2.3. LW-YOLOv7

Although YOLOv7 performs well in real-time object detection, there are still some issues when we use it to detect maize seedling in images. Firstly, different images have many similarities in features. When the model is training, similar features can be extracted from different images. It will occupy a large amount of memory space for the model to be deployed on mobile devices. Secondly, for detecting small objects, multiple convolutions and up-sampling operations will lead to the loss of location information. Finally, YOLOv7 converges too slowly when calculating loss. To solve the above problems, we propose a lightweight model base on the YOLOv7, which has been improved as follows.

1. To tackle the problem of redundant features during training, we use the GhostNet module to replace the ordinary convolution of the Backbone in the YOLOv7. At the same time, we introduce the CBAM attention mechanism module to improve global attention with channel attention module and spatial attention module.
2. To solve the position information of small objects, we use BiFPN to replace PANet in the Head and enhance the representation ability of features by adding residual links.
3. In terms of loss convergence, the SIoU loss function is used instead of the CIoU loss function to reduce the degree of freedom of the loss function, enhance network robustness, and improve the speed and accuracy of box regression.

2.3.1. Improvement of Backbone

In YOLOv7, a significant computational bottleneck arises from the CBS (Convolution, BN, SiLU) modules. These modules have high computational demand and require substantial memory space during training, posing challenges for deploying the model on resource-limited mobile devices. To address this issue, we integrate the GhostNet [41] model into YOLOv7. The GhostNet model, as shown in Figure 4, employs a two-step process to reduce computational complexity. Firstly, it utilizes ordinary convolutions with a kernel size of 1×1 to obtain intrinsic feature maps. These convolutions help capture essential information and reduce the dimensionality of the features. Secondly, GhostNet employs cheap operations to generate redundant feature maps based on intrinsic features.

These redundant feature maps provide additional information without significantly increasing computational costs. By concatenating the intrinsic and redundant feature maps, GhostNet achieves better performance in object detection while minimizing computational costs. This approach effectively addresses the computational limitations of CBS modules, making it more feasible to deploy the model on resource-limited mobile devices.

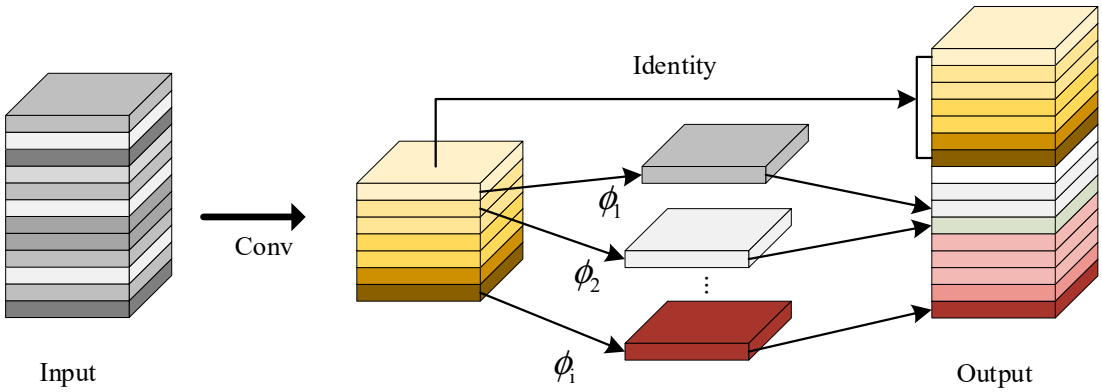


Figure 4. GhostNet. Conv represents an ordinary convolution with a 1×1 kernel size while Φ_i represents a sequence of linear transformations. It operates on each channel, and its computational cost is much lower than ordinary convolution. The convolutional kernel size of cheap operation is 5×5.

While GhostNet helps reduce the number of parameters in the backbone of YOLOv7, it may also lead to the omission of important features. To solve this issue, we have explored the integration of attention modules. Among them, the CBAM (Convolutional Block Attention Module) [42] has demonstrated the most promising results in terms of detection accuracy within our model. The CBAM, depicted in Figure 5, combines channel attention and spatial attention mechanisms to enhance the model's ability to capture informative features.

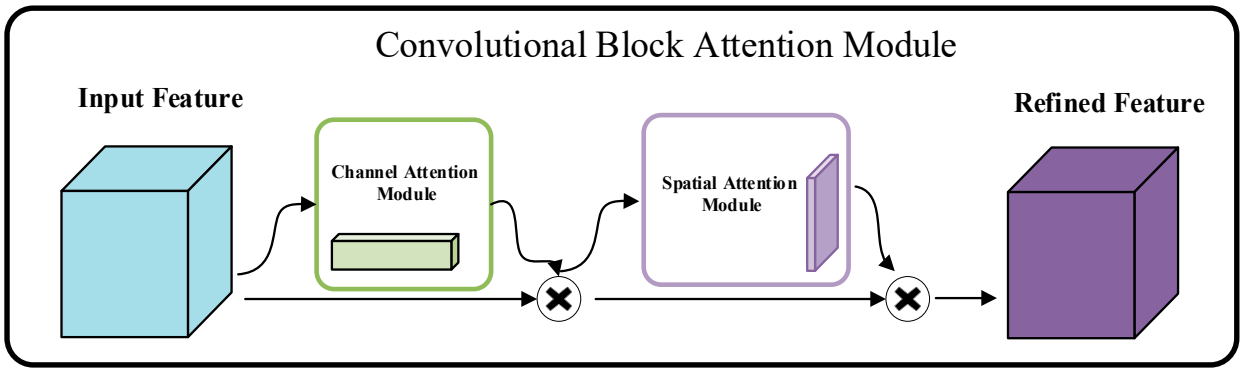


Figure 5. Convolutional Block Attention Module.

CBAM is a lightweight and effective attention module for feed-forward convolutional neural networks. It includes two sub-modules: channel attention module (CAM) and spatial attention module (SAM). The CAM and SAM models can be viewed in Figure 6. Firstly, the input feature map F($F \in R^{c \times h \times w}$) was processed through a one-dimensional convolutional operation of CAM, and the result M_c($M_c \in R^{c \times 1 \times 1}$) was multiplied with the input feature F. Secondly, by using the output result of CAM as channel-refined feature F′, a two-dimensional convolution operation of the SAM was conducted. Finally, the result M_s($M_s \in R^{1 \times h \times w}$) was multiplied with the CAM output to obtain the final result F″($F'' \in R^{c \times h \times w}$).

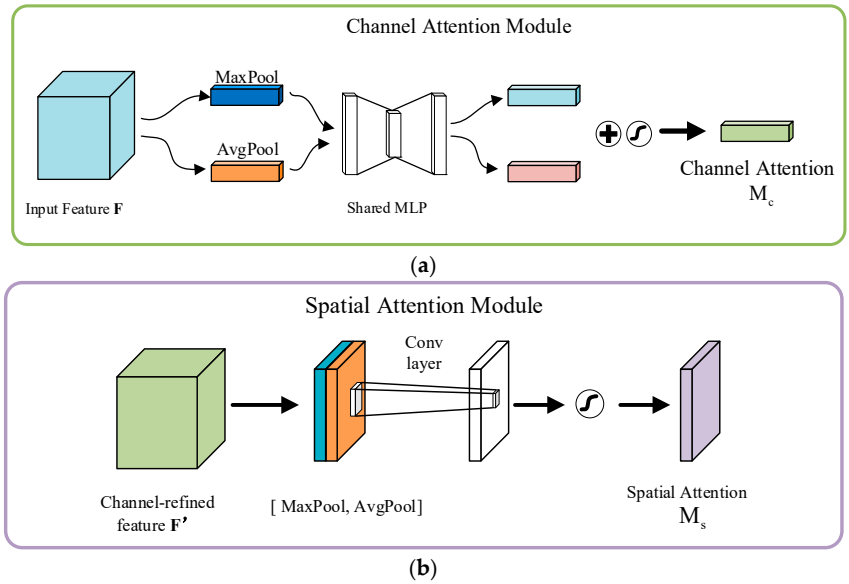


Figure 6. Sub-modules of CBAM. (**a**) Channel Attention Moule; (**b**) Spatial Attention Module.

The attention-generating process of the CBAM are shown in Equations (1) and (2),

$$F' = M_c(F) \otimes F \tag{1}$$

$$F'' = M_c(F') \otimes F' \tag{2}$$

where $\otimes$ denotes element-wise multiplication. M_c represents a one-dimensional convolution operation of CAM, while M_s represents a two-dimensional convolution operation of SAM. F denotes the input feature, and F″ denotes the output feature.

By introducing GhostNet and CBAM modules, we have made changes to the structure of ELAN and MP, as shown in Figure 7. Compare with the original network, the new network decreases the number of parameters and is more suitable for deployment in mobile devices with limited mobile resources. Additionally, it also can improve its feature extraction capability by introducing the CBAM module and solving the problem of easily ignoring small maize seedlings in complex field environments.

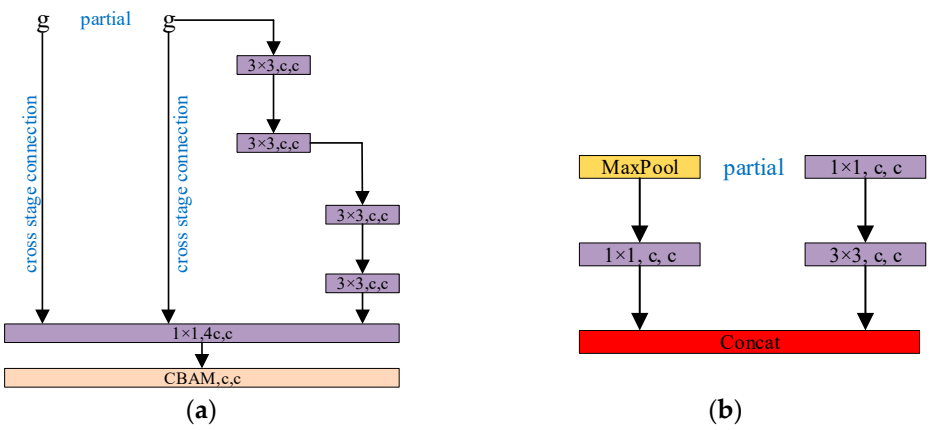


Figure 7. New structures of backbone. (**a**) Denotes GhostELAN; (**b**) Denotes GhsotMP. "1 × 1, 4c, c" denotes the GhostNet operation, which has convolutional kernel size of 1 × 1; its input channel is 4c and output channel is c.

2.3.2. Improvements of Head

YOLOv7's Head mainly consists of FPN and PANet. Firstly, FPN structure performs the forward feature extraction of deep convolutional networks. Secondly, based on the P5 layer, the FPN structure performs two times up-sampling from top to bottom. It can obtain feature maps with sizes of 40 × 40 and 80 × 80, respectively. Finally, the feature maps obtained from the previous step are added to the P4 and P3 layers. The structure of FPN is shown in Figure 8a.

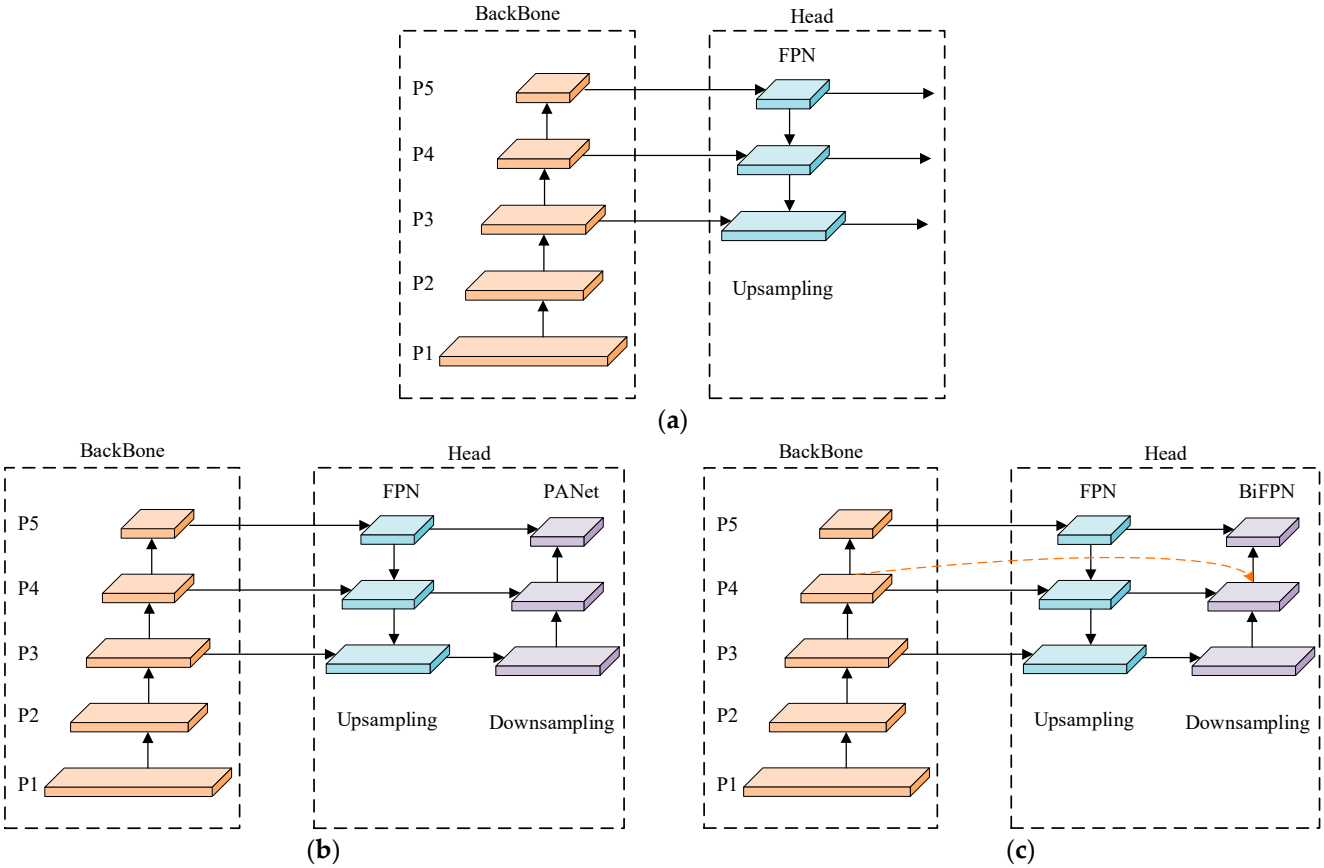


Figure 8. The structure of FPN, PANet, and BiFPN (**a**) is the structure of FPN; (**b**) is the structure of PANet; (**c**) is the structure of BiFPN.

Although FPN can propagate high-level semantic information to lower levels, the nearest neighbor interpolation method employed in up-sampling may not efficiently distribute information. To solve this problem, as an alternative to FPN, PANet is used in YOLOv7. PANet is shown in Figure 8b.

PANet is an instance segmentation framework, and it cannot directly perform object detection. However, it can enhance multi-scale information fusion. In PANet, the fusion of feature maps from different levels is achieved through path aggregation, which ensures the continuity and consistency of features.

Compared to FPN structure, PANet has made significant improvements in object-detection task performance. However, it has some disadvantages, such as high computational cost, model training difficulty, and errors in recognizing small objects.

In order to improve PANet, we referred to the BiFPN [18] structure, which is shown in Figure 8c.

BiFPN is a neural network optimized for object detection based on PANet and offers several advantages. Firstly, BiFPN is highly efficient. It has few parameters that reduce the computational complexity of model and can complete the calculation of the FPN structure in a short time. It is suitable for real-time object detection tasks. Secondly, it

has high precision. BiFPN transfers contextual information through adaptive features. It can maintain the consistency of semantic information at different scales and improve the accuracy of object detection. Finally, it has high robustness. BiFPN can handle objects of different sizes and shapes, and it has better adaptability to image rotation, scaling, and translation.

In this paper, we remove the number of single points of input features and reduce the computational load of the network. Adding a residual link in the output section can enhance the ability to express features. Increasing the weight of each scale feature after fusion can adjust the contribution of each scale.

2.3.3. Improvements to the Loss Function

The accuracy and convergence speed of object detection model depends on the loss function largely. The loss function of YOLOv7 contains three parts: coordinate loss, confidence loss, and classification loss. YOLOv7 adopts the calculation method of CIoU loss in the coordinate loss, and its expression is shown in Equation (3).

$$L_{CIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + av \tag{3}$$

In Equation (3), $\rho^2(b, b^{gt})$ is the Euclidean distance between the center points of the predicted box and ground truth. c represents the diagonal distance of the minimum circumscribed matrix. IoU is defined as the ratio of the intersection area between the predicted box and the ground truth to the union area of the two bounding boxes. a is a weight parameter, and v indicates the similarity between length and width. Equation (4) and Equation (5) display the expressions of a and v, respectively.

$$a = \frac{v}{(1 - IoU) + v} \tag{4}$$

$$v = \frac{4}{\pi}(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w}{h})^2 \tag{5}$$

In Equation (5), w_{gt} and h_{gt} represent the width and height of the ground truth, and w and h represent the width and height of the predicted box.

CIoU is a newer evaluation metric that improves upon the IoU and its variants. The CIoU takes into account not only the extent of overlap between the predicted box and ground truth but also their distance apart and aspect ratio differences. However, it does not take into account the direction between ground truth and predicted box results in slow model convergence, low efficiency, low performance, etc. Therefore, we introduce the SIoU loss function [43] to optimize the coordinate loss. Compared to CIoU, the advantages of SIoU include robustness to partial occlusion, better handling of objects with different scales, and faster convergence speed during the training process. The calculation method of the SIoU loss function mainly includes four parts: angle cost, distance cost, shape cost, and IoU cost, and its expression, shown in Equation (6) is as follows:

$$L_{SIoU} = 1 - IoU + \frac{\Delta + \Omega}{2} \tag{6}$$

In Equation (6), Δ is the distance cost, and Ω is the shape cost. Since the angle cost is added to the SIoU function, it will reduce the probability of a penalty term equal to 0. This not only accelerates the convergence of the loss function, but it also reduces the prediction errors. The parameters and their relationships for the SIoU loss function are shown in Figure 9.

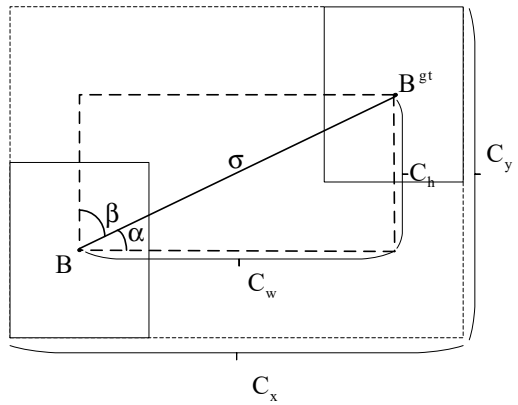


Figure 9. Parameters and their relationships of SIoU loss function.

B represents the predicted box and B_{gt} represents the ground truth. C_x is the width of the minimum bounding matrix. C_y is the height of the minimum bounding matrix. C_w and C_h are the width difference and height difference between the center point of the predicted box and the ground truth, respectively. σ is the shortest linear distance between the predicted box and the ground truth. Sin α is the ratio of C_h to σ, and sin β is the ratio of C_w to σ.

If α is equal to 0 or π/2, the angle cost (Λ) will be 0. If α is less than π/4, we prioritize the minimization of α. Otherwise, we prioritize the minimization of β. The expression for the angle cost is given by Equation (7):

$$\Lambda = 1 - 2\sin^2(\arcsin(x) - \frac{\pi}{4}) \tag{7}$$

in Equation (7),

$$x = \frac{C_h}{\sigma} = \sin(\alpha) \tag{8}$$

$$\sigma = \sqrt{\left(b_{c_x}^{gt} - b_{c_x}\right)^2 + \left(b_{c_y}^{gt} - b_{c_y}\right)^2} \tag{9}$$

$$C_h = \max(b_{c_y}^{gt}, b_{c_y}) - \min(b_{c_y}^{gt}, b_{c_y}) \tag{10}$$

The distance cost in the SIoU loss function is determined by the distance between the center points of the ground truth and the predicted box, which is further affected by the angle cost. The distance cost function is redefined with these considerations, and its expression is given by Equation (11):

$$\Delta = \sum_{t=x,y} (1 - e^{-\gamma\rho_t}) \tag{11}$$

in Equation (11),

$$\rho_x = \left(\left(b_{c_x}^{gt} - b_{c_x}\right)/C_w\right)^2 \tag{12}$$

$$\rho_y = \left(\left(b_{c_y}^{gt} - b_{c_y}\right)/C_h\right)^2 \tag{13}$$

$$\gamma = 2 - \Lambda \tag{14}$$

As α approaches 0, the contribution of distance cost decreases continuously. On the contrary, when α is closer to π/4, the contribution of the distance cost keeps increasing. As the angle increases, γ is assigned a time-preferred distance value.

The shape cost is defined in Equation (15).

$$\Omega = \sum_{t=w,h} \left(1 - e^{-\omega_t}\right)^{\theta} \tag{15}$$

θ controls the attention paid to the shape cost. In this paper, the value of θ is set to 1 to optimize the aspect ratio, thereby restricting the free movement of the shape, where $\omega_{t=w}$ $\omega_w = \left|w - w^{gt}\right| / \max(w, w^{gt})$, $\omega_h = \left|h - h^{gt}\right| / \max(h, h^{gt})$.

2.3.4. LW-YOLOv7 Maize Seedling Detection Model

Based on the original advantages of YOLOv7, we proposed an improved LW-YOLOv7 model to detect maize seedlings in the real environment while ensuring detection accuracy; it reduced the number of parameters and improved the convergence speed of the model. The overall framework of LW-YOLOv7 is shown in Figure 10. In the backbone network, we use ReLU as the activation function, and the total number of layers of the backbone is 54. The specific network architecture of the backbone is shown in Table 2.

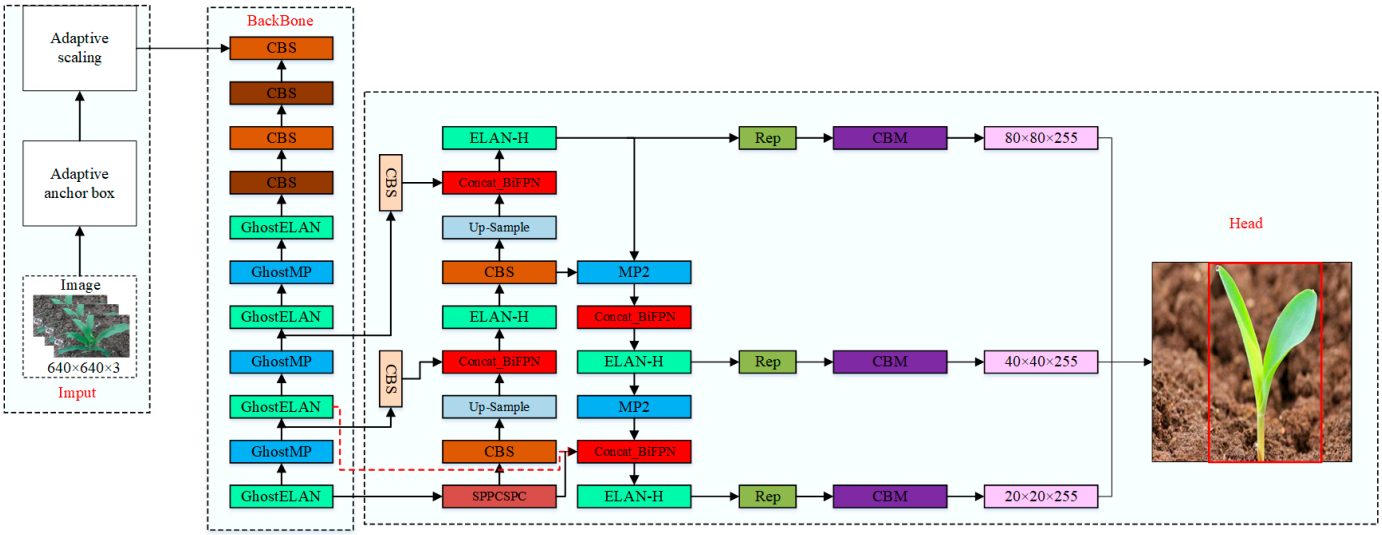


Figure 10. Improved model structure of LW-YOLOv7.

Table 2. Network architecture of the backbone.

Module Name	Input	Output	Filter Size	Layers
CBS	640 × 640 × 3	640 × 640 × 32	1	0
CBS	640 × 640 × 32	320 × 320 × 64	2	1
CBS	320 × 320 × 64	320 × 320 × 64	1	2
CBS	320 × 320 × 64	160 × 160 × 128	2	3
GhostELAN	160 × 160 × 128	160 × 160 × 256	1	4–12
GhostMP	160 × 160 × 256	80 × 80 × 256	2	13–17
GhostELAN	80 × 80 × 256	80 × 80 × 512	1	18–26
GhostMP	80 × 80 × 512	40 × 40 × 512	2	27–31
GhostELAN	40 × 40 × 512	40 × 40 × 1024	1	32–40
GhostMP	40 × 40 × 1024	20 × 20 × 1024	2	41–45
GhostELAN	20 × 20 × 1024	20 × 20 × 1024	1	46–54

3. Results and Analysis

3.1. Experimental Environment and Evaluation Indicators

The training task is implemented on Python 3.9 and Pytorch 1.13. The device information required for training includes Inter(R) Core(TM) i9-10980XE CPU @ 3.00 GHz, NVIDIA GeForce RTX 3090, and 64 GB of memory.

During training, we need to calculate the size of the anchor boxes so that the model can better predict the bounding boxes of maize seedlings. K-means is an unsupervised learning algorithm. It treats all the bounding boxes in the training set as samples and takes the width and height of the bounding boxes as two features. During calculation, we set the number of anchor boxes to 9 and use the new distance calculation equation. The new calculation method is shown in Equation (16). Secondly, for each sample in the training set, calculate its distance from all cluster centers and assign it to the nearest cluster. For each cluster, recalculate the center point of the cluster; that is, select the average value of all samples in the cluster as the new cluster center. Finally, K-Means perform 1000 epochs of iteration to obtain the optimal anchor boxes on the above operations. In deep learning, excessive input image resolution may lead to issues such as insufficient memory and long calculation time. Therefore, we set the image resolution to 640 × 640. Additionally, when we were training maize seedlings data, the training results tended to stabilize at 80 epochs. To solve the problem of overfitting, we set the total number of iterations to 100. The learning rate is a very important hyperparameter in deep learning, which controls the speed of updating model parameters during gradient descent optimization. If the learning rate is too large, it may lead to the model cannot converge and large fluctuations in training error. Therefore, we set the initial learning rate to 0.01, the learning rate frequency to 0.1, and the final learning rate to 0.001. In YOLOv7, establishing the batch size has a significant impact on the training effect and speed of the model. Batch size refers to the number of samples processed simultaneously in each round of gradient descent, which can affect multiple aspects such as the direction, amplitude, and speed of model parameter updates. A larger batch-size value can improve the training speed of the model, but it may lead to overfitting, and an inability to converge, among other issues. Therefore, we set the batch size to 16. We also provided other hyperparameters and information during the training process. Table 3 shows the parameters of the training process.

$$\text{Distance} = 1 - \text{IoU}(\text{box}, \text{centroid}) \tag{16}$$

Table 3. Parameters of the training process.

Parameter Name	Parameter Value or Range
Image size	640 × 640
Batch size	16
Learning rate	0.01
Learning rate frequency	0.1
Max training epoch	100
Momentum	0.937
Decay	0.0005
Anchor boxes	[9,8 12,9 9,12] [15,13 19,13 12,17] [28,21 30,34 44,50]

To assess the performance of the improved model, we use five evaluation metrics, including precision, recall, mAP, parameter, and FPS. Precision is a performance metric commonly used in the evaluation of classification models. It measures the proportion of true positives among all instances that the model has classified as positive. Recall is an important metric for evaluating a model's detection ability, reflecting its ability to correctly identify all positive samples. Specifically, recall is defined as the proportion of true positive samples that are correctly detected out of all actual positive samples. Another commonly used metric is mAP, which stands for mean average precision. It is a method to evaluate the overall performance of model-detection results and take into account the performance of the model at different thresholds. The calculation of mAP is as follows: sort all positive samples by confidence from high to low, then calculate the precision at each confidence, and, finally, obtain the average precision across all the confidences. Parameter refers to the

size of the best weight file generated in the deep-learning model. This metric is commonly considered a way to measure the complexity of the model because larger parameters usually indicate higher model complexity. FPS refers to the number of frames per second that a model can process, which is a measure of the model's inference speed. This metric is often hardware-dependent and can be improved by optimizing the computation graph, reducing the model size, etc.

The precision and recall calculation methods are noted by Equations (17) and (18), respectively.

$$\text{Precision} = \text{TP}/(\text{TP} + \text{FP}) \tag{17}$$

$$\text{Recall} = \text{TP}/(\text{TP} + \text{FN}) \tag{18}$$

During the experiment, we set the IOU threshold to 0.5. If an object is correctly predicted and the IOU threshold exceeds our setting, it can be considered a positive sample, while the others are considered negative samples, where precision measures the ratio of the correct predicted samples to the total samples. The recall is the ratio of correctly predicted positive samples to the total number of positive samples. TP is the number of positive samples that are correctly identified. FP is the number of negative samples recognized as positive samples. FN is the number of positive samples recognized as negative samples.

AP represents the area enclosed by precision and recall. Specifically, the method of calculation is shown in Equation (19); P(R)dR is a function that recalls as the abscissa and precision as the ordinate. In Equation (20), mAP refers to the average value of AP across all categories, where n represents the number of categories in the dataset. In this paper, there is only one category to be detected, which makes AP = mAP.

$$AP = \int_0^1 P(R)dR \times 100\% \tag{19}$$

$$mAP = \frac{1}{n}\sum_{i=1}^{n-1} AP_i \times 100\% \tag{20}$$

FPS represents the number of images detected per second. In testing, to meet the requirements of real-time detection, the value of FPS should be greater than 30. FPS contains three parts: the image pre-processing time (Pre), inference time (Inf) of the network, and non-maximum suppression time (NMS). FPS is shown in Equation (21).

$$\text{FPS} = 1000/(\text{Pre} + \text{Inf} + \text{NMS}) \tag{21}$$

3.2. Comparison of Different Backbones Based on YOLOv7

In this experiment, to verify the feature extraction capabilities of different backbones, we select MobileNet-V2, MobileNet-V3, ShuffleNet-V2, and LW-YOLOv7 for comparison with the YOLOv7 algorithm. The comparison results are shown in Table 4. The precision of the YOLOv7+MobileNet-V2 algorithm decreased by 1.3%, the recall decreased by 1.8%, the mAP decreased by 1.8%, and the number of parameters decreased by 25.2 M. The precision of the YOLOv7+MobileNet-V3 algorithm decreased by 3.4%, the recall decreased by 0.4%, the mAP decreased by 2.9%, and the number of parameters decreased by 27.5 M. The precision of the YOLOv7+ShuffleNet-V2 algorithm decreased by 5.6%, the recall increased by 0.7%, the mAP decreased by 3.6%, and the number of parameters decreased by 28.7 M. The precision of the LW-YOLOv7 algorithm decreased by 1.6%, the recall increased by 3%, the mAP increased by 0.5%, and the number of parameters decreased by 15.2 M.

Table 4. Comparison of lightweight backbone network models.

Networks	Precision (%)	Recall (%)	mAP (%)	Parameters (M)	FPS (f/s)
YOLOv7+MobileNet-V2	89.6	80.7	90.9	49.6	93
YOLOv7+MobileNet-V3	87.5	82.1	89.8	47.3	94
YOLOv7+ShuffleNet-V2	85.3	83.2	89.1	46.1	94
YOLOv7	90.9	82.5	92.7	74.8	121
LW-YOLOv7	89.3	85.5	93.2	59.4	90

Based on the comparison results, it can be observed that using lightweight models may result in a decline in precision. Among them, the YOLOv7+ShuffleNet-V2 algorithm experienced the most significant decrease in precision, while the LW-YOLOv7 algorithm exhibited a relatively minor drop compared to other models. This is because lightweight models often adopt simplified network structures, which may reduce their feature representation capability and subsequently impact the models' accurate object detection. However, lightweight models have advantages in reducing parameter quantity and computational resource requirements, making them suitable for resource-constrained situations. Additionally, we can introduce methods such as CBAM and BiFPN to improve the detection accuracy of the model. This is because they can enhance the representation and fusion capabilities of feature maps.

3.3. Comparison between CIoU and SIoU

To assess the efficacy of the SIoU loss function, this experiment conducted a comparative analysis with the CIoU and DIoU loss functions. The results, as presented in Figure 11, demonstrate several important findings.

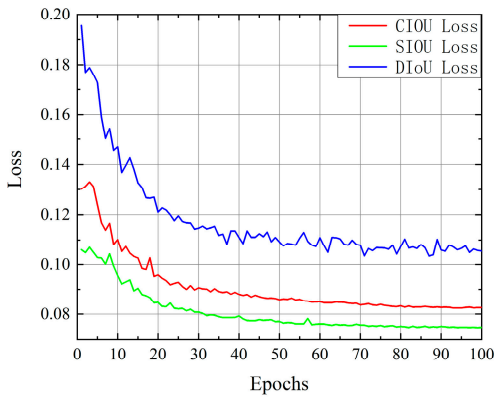


Figure 11. Comparison of loss values.

Firstly, it is observed from the figure that the coordinate loss stabilizes after approximately 80 epochs during the training phase. This indicates that the network has reached a point of convergence where further training may not significantly improve performance. Based on this observation, the training epoch was set to 100. Secondly, the SIoU loss function exhibits a significantly faster convergence speed compared to the CIoU and DIoU loss function. This implies that the SIoU loss function allows for more efficient optimization and training of the model, leading to quicker convergence toward the desired performance. Lastly, the error ratio between the ground truth and the predicted bounding box decreases by 9.49% to 45.78% when using the SIoU loss function. This reduction in the error ratio suggests that the SIoU loss function improves the accuracy of the predicted bounding boxes, leading to better object-detection performance.

To summarize, the comparative analysis reveals that the SIoU loss function outperforms the CIoU and DIoU loss functions in terms of convergence speed and error reduction.

These findings indicate that the SIoU loss function is more effective for assessing the accuracy of object-detection models, particularly in the context of maize detection at the seedling stage.

3.4. Comparative of Different Object Detection Models

To verify the effectiveness of the LW-YOLOv7 model proposed in this paper, we trained and tested it on our dataset and compared it to other popular object-detection models. These models include Faster-RCNN, YOLOv3, YOLOv4, YOLOv5, and YOLOv8. The comparison results are shown in Table 5. Compared to LW-YOLOv7, the precision of the Faster-RCNN model decreased by 26.7%, the recall decreased by 27.4%, the mAP decreased by 37.6%, and the number of parameters increased by 45%, while the FPS is 14.3 f/s and decreased by 529.37%. Therefore, the detection accuracy of Faster-RCNN was significantly lower than YOLOv7 and did not meet the requirements of real-time detection.

Table 5. Comparison results of different detection algorithm models.

Model Name	Precision (%)	Recall (%)	mAP (%)	Parameters (M)	FPS (f/s)
Faster-RCNN	62.6	58.1	55.6	108	14.3
YOLOv3	88.4	82.2	91.3	123.4	82
YOLOv4	89.9	83	92.2	105.4	107
YOLOv5	89.2	79.8	89.3	91.4	114
YOLOv8	91.2	83.3	93.2	87.6	125
LW-YOLOv7	89.3	85.5	93.2	59.4	90

The precision of the YOLOv3 model decreased by 0.9%, the recall decreased by 3.3%, the mAP decreased by 1.9%, the number of parameters increased by 51.86%, and the value of FPS is 82 f/s. YOLOv3 can meet the requirement of real-time detection, but the accuracy is lower than LW-YOLOv7. Additionally, it also takes up more memory and disk space on mobile devices.

The precision of the YOLOv4 model increased by 0.6%, the recall decreased by 2.5%, the mAP decreased by 1%, the number of parameters increased by 43.64%, and the value of FPS is 107 f/s. Therefore, YOLOv4 also requires more memory and disk space.

The precision of the YOLOv5 model decreased by 0.1%, the recall decreased by 5.7%, the mAP decreased by 3.9%, the number of parameters increased by 35.01%, and the value of FPS is 114 f/s. Therefore, YOLOv5 not only has lower detection accuracy than LW-YOLOv7, but it also has a significant computation load.

The precision of the YOLOv8 model increased by 1.9%, the recall decreased by 2.2%, the mAP has not changed, the number of parameters increased by 32.19%, and the value of FPS is 125 f/s. Compared with other popular algorithms, the comparison results show that the LW-YOLOv7 proposed in this paper has the highest mAP value, the model size is smaller than other popular object-detection algorithms, and the detection speed meets the requirements of real-time detection.

3.5. Ablation Test Comparison

In order to provide the impact of different modules on the detection accuracy and speed of the model, the ablation experience results are shown in Table 6.

In Table 6, we used YOLOv7's metrics as a baseline, which is shown in the first row of Table 4. mAP, parameters, and FPS of YOLOv7 are 92.7%, 74.8 M, and 121 f/s, respectively.

Introducing the GhostNet module, the mAP decreased by 1.2%, but the total parameters also decrease by 21.39%. The value of FPS is 94, which meets the real-time requirements.

Table 6. Results of ablation experiments.

Index	GhostNet	CBAM	BiFPN	SIoU Loss	Total Parameters (M)	mAP (%)	FPS (f/s)
YOLOv7	-	-	-	-	74.8	92.7	121
1	√	-	-	-	58.8	91.5	94
2	√	√	-	-	59.4	92.3	92
3	√	√	√	-	59.4	92.9	92
4	√	√	√	√	59.4	93.2	90

Based on GhostNet, introducing the CBAM attention mechanism module, the mAP and total parameters were 92.3% and 59.4, and they decreased by 0.4% and 20.58%, respectively.

Introducing the BiFPN module, the mAP was 92.9%, which increased by 0.2%. The total parameters were 59.4 M, which decreased by 20.58%.

Introducing the SIoU function, the mAP was 93.2%, which increased by 0.5%. The total parameters were 59.4 M, which decreased by 20.58%.

The study results indicate, by introducing the GhostNet model, the total parameters can be reduced and the slightly reduced detection accuracy. Adding the CBAM module, the accuracy of the model will increase. Moreover, through introducing the module of BiFPN and CIoU function, the accuracy also increases, and the total parameters keep unchanged.

3.6. Comparative of Different Class Activation Mapping

Deep-learning networks are often considered black box experiments during training and are not easily interpretable. In order to gain a better understanding of the model's recognition process, it is important to analyze its internal workings and how it processes input data. This can involve examining its architecture, training data, feature extraction methods, and prediction mechanisms, among other factors. Additionally, visualizations such as heatmaps and saliency maps can provide insights into which areas of an input image are most relevant to the model's predictions. Therefore, the experiment introduced Grad-CAM [44]. Grad-CAM is a technique that can be utilized to visualize the attention of the model and identify which parts of the input image are most important for generating its predictions. Specifically, Grad-CAM calculates the gradients of the target class output for the feature map of the final convolutional layer and then applies them to obtain a weighted sum of activation maps. The resulting heatmap highlights regions of the input image that had the greatest influence on the model's predictions. The heat map image of GradCAM is shown in Figure 12. Based on the results displayed in Figure 12, it is evident that the improved models proposed in the paper exhibit greater capability to identify and extract features of maize seedlings compared to their counterparts. Additionally, the models show less susceptibility to being affected by complex environmental factors. As a result, this approach provides a better explanation of the deep-learning process, as it can better prioritize and capture relevant features while filtering out extraneous information.

3.7. Comparison of Object-Detection Visualization Results

In order to verify the effectiveness of the LW-YOLOv7 mode in the field environment, this study uses the model to perform real-time objection on maize seeding images captured by drones and selects other three existing popular object-detection models (Faster RCNN, YOLOv5, YOLOv7) that are compared with the LW-YOLOv7 model. Faster RCNN is a two-stage object detection model.YOLOv5, YOLOv7, and LW-YOLOv7 are all single-stage object detection models. The results are shown in Figure 13.

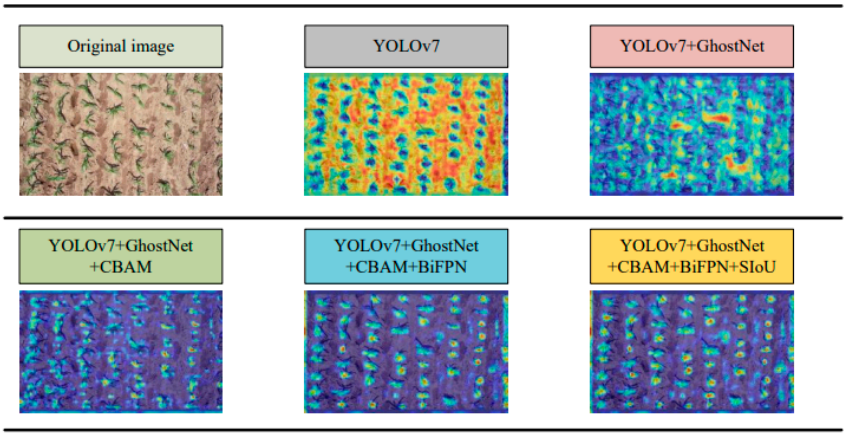


Figure 12. The heat maps of GradCAM.

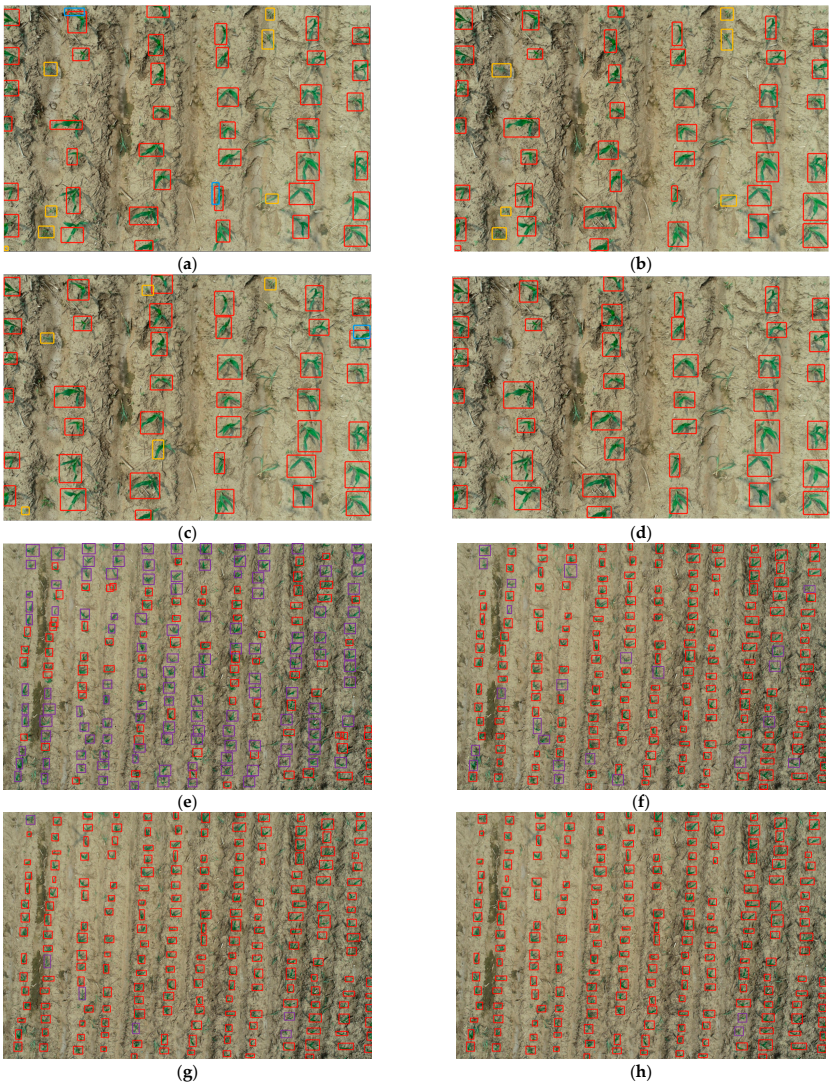


Figure 13. Object-detection results of four models on two different sizes of objects. (**a**) Faster RCNN's detection result on big objects; (**b**) YOLOv5l's detection result on big objects; (**c**) YOLOv7′s detection result on big objects; (**d**) LW-YOLOv7′s detection result on big objects; (**e**) Faster RCNN's detection result on small objects; (**f**) YOLOv5l's detection result on small objects; (**g**) YOLOv7′s detection result on small objects; (**h**) LW-YOLOv7′s detection result on small objects. The red box is the correct prediction, the blue box is the duplicate prediction, the yellow box is the wrong prediction, and the purple box is the not detection.

This experiment selected two kinds (shooting heights) of maize seedlings images at different heights. Where (a)–(d) shooting height is 1.6 m, (e)–(h) shooting height is 5 m.

As shown in Figure 13a–d, using Faster RCNN for detection, there are positioning deviations, detection errors (mistakenly identifying weeds as seedling maize plants), and repeated detection. Using YOLOv5 for detection, although the problem of repeated detection is solved, a large number of weeds and negative samples still be identified as maize seedlings. Using YOLOv7 for prediction, the problem of positioning deviation is solved, but there are false detections. However, with LW-YOLOv7, we solved the problems of repeated detection and recognition errors.

As shown in Figure 13e–h, Faster RCNN has a large number of missed detections. YOLOv5 and YOLOv7 also have some missed detections. However, LW-YOLOv7 performs better in small object detection than the other three models. It can be seen that when using LW-YOLOv7 on maize seedling images, it can effectively avoid the positioning deviation and run-time long of the Faster RCNN model. Additionally, it can solve the single-stage object detection model (YOLOv5 and YOLOv7) and repeated detection problem, and it reduces the probability of misidentification. The comparison of the four models proved the advantages of the LW-YOLOv7 model in real-time object detection tasks of maize seedling images, and it provides a solution basis for the rapid realization of the seedling maize plant detection in the field environment.

4. Conclusions

In this study, we present a new model to solve the difficult tasks of real-time detection in maize seedling images. The new model improves the accuracy of detection and reduces the number of parameters. By using the LW-YOLOv7 detection model, we can obtain the position and quantity of maize seedlings, as well as judge the density and growth uniformity of maize. In addition, we can calculate the emergence rate and replant the missed areas in a timely manner to increase maize yield. It is important for evaluating maize growth and yield. On the training model, we can change the depth and width of the network to reducing the size of weight. This provides a theoretical basis for deploying our improved model on devices with limited mobile resources. The proposed maize seedling detection model can provide benefits for scientific researchers engaged in object detection and for those who use the model for real-time detection in agriculture. Using this model can save a lot of manpower and time. In addition, this model can be integrated into unmanned aerial vehicle (UAV) systems for real-time detection, further enhancing its practicality. In this way, we can quickly and easily monitor large areas of land, identify seedlings that require attention, and take action before problems arise.

While the proposed maize seedling model has many benefits, there are still challenges that need to be addressed to improve its generalizability across different datasets and environments. Firstly, the dataset has certain specificity. The model is trained and tested on a specific dataset, which may limit its adaptability to maize seedlings at different stages. If applied to scenarios with significant differences from the training dataset, the model's performance may decline. Secondly, the model has implicit class limitations. LW-YOLOv7 is a single-stage detector that typically predicts multiple bounding boxes and corresponding classes for each position in the input image. However, this design may result in implicit class limitations, meaning the model can struggle to accurately differentiate objects with overlapping or similar features. Finally, it involves model complexity and computational resource requirements. Despite having a relatively smaller model size compared to other popular object-detection models, LW-YOLOv7 may still require substantial computational resources. On low-performance devices such as mobile or embedded systems, the model may face challenges due to limited computational resources.

Despite these challenges, our findings demonstrate that LW-YOLOv7 object-detection models offer great promise in addressing real-world problems. In the future, we plan to improve the detection accuracy and speed in maize seedling detection. The weight of the

model further reduced to facilitate better deployment of identification tasks on the edge computing platform.

Author Contributions: Conceptualization, L.Z.; methodology, Y.Z.; writing—original draft preparation, K.Z.; project administration, H.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Supported by National Natural Science Foundation of China (Grant No. 31901399); Sub project of the Fourteenth-Five Year National Key R&D Plan (Grant No. 2021YFD1500204); Sub project of National Key R&D Plan (Grant No. 2022YFD2002303-01); Liaoning Province Innovation Capability Enhancement Joint Fund Project (Grant No. 2021-NLTS-11-03).

Data Availability Statement: Not applicable.

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

References

1. Li, S.; Zhao, J.; Dong, S.; Zhao, M.; Li, C.; Cui, Y.; Liu, Y.; Gao, J.; Xue, J.; Wang, L.; et al. Advances and prospects of maize cultivation in China. *Sci. Agric. Sin.* **2017**, *50*, 1941–1959.
2. Zhang, Y.; Guo, E.; Wang, Y.; Gu, X.; Kang, Y. The effects of extreme precipitation events on maize yield in Jilin Province. *China Rural Water Hydropower* **2023**, *483*, 52–61.
3. Yu, Z.; Cao, Z.; Wu, X.; Bai, X.; Qin, Y.; Zhuo, W.; Xiao, Y.; Zhang, X.; Xue, H. Automatic image-based detection technology for two critical growth stages of maize: Emergence and three-leaf stage. *Agric. For. Meteorol.* **2013**, *174*, 65–84. [CrossRef]
4. Zhao, B.; Zhang, J.; Yang, C.; Zhou, G.; Ding, Y.; Shi, Y.; Zhang, D.; Xie, J.; Liao, Q. Rapeseed seedling stand counting and seeding performance evaluation at two early growth stages based on unmanned aerial vehicle imagery. *Front. Plant Sci.* **2018**, *9*, 1362. [CrossRef] [PubMed]
5. Xia, L.; Zhang, R.; Chen, L.; Huang, Y.; Xu, G.; Wen, Y.; Yi, T. Monitor cotton budding using SVM and UAV images. *Appl. Sci.* **2019**, *9*, 4312. [CrossRef]
6. Uijlings, J.R.; Van De Sande, K.E.; Gevers, T.; Smeulders, A.W. Selective search for object recognition. *Int. J. Comput. Vis.* **2013**, *104*, 154–171. [CrossRef]
7. Li, B.; Yan, J.; Wu, W.; Zhu, Z.; Hu, X. High performance visual tracking with siamese region proposal network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8971–8980.
8. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
9. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
10. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*. [CrossRef]
11. Pan, Y.; Zhu, N.; Ding, L.; Li, X.; Goh, H.-H.; Han, C.; Zhang, M. Identification and Counting of Sugarcane Seedlings in the Field Using Improved Faster R-CNN. *Remote Sens.* **2022**, *14*, 5846. [CrossRef]
12. Yu, Y.; Zhang, K.; Yang, L.; Zhang, D. Fruit detection for strawberry harvesting robot in non-structural environment based on Mask-RCNN. *Comput. Electron. Agric.* **2019**, *163*, 104846. [CrossRef]
13. Li, Z.; Li, Y.; Yang, Y.; Guo, R.; Yang, J.; Yue, J.; Wang, Y. A high-precision detection method of hydroponic lettuce seedlings status based on improved Faster RCNN. *Comput. Electron. Agric.* **2021**, *182*, 106054. [CrossRef]
14. Wang, Q.; Qi, F. Tomato diseases recognition based on faster RCNN. In Proceedings of the 2019 10th International Conference on Information Technology in Medicine and Education (ITME), Qingdao, China, 23–25 August 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 772–776.
15. Jiang, Y.; Li, C.; Paterson, A.H.; Robertson, J.S. DeepSeedling: Deep convolutional network and Kalman filter for plant seedling detection and counting in the field. *Plant Methods* **2019**, *15*, 141. [CrossRef]
16. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; Springer: Cham, Switzerland, 2016; pp. 21–37.
17. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
18. Tan, M.; Pang, R.; Le, Q.V. Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10781–10790.
19. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.

20. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 June 2017; pp. 7263–7271.
21. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
22. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
23. Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W. Yolov6: A single-stage object detection framework for industrial applications. *arXiv* **2022**, arXiv:2209.02976.
24. 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]
25. Sekharamantry, P.K.; Melgani, F.; Malacarne, J. Deep Learning-Based Apple Detection with Attention Module and Improved Loss Function in YOLO. *Remote Sens.* **2023**, *15*, 1516. [CrossRef]
26. Zhou, F.; Deng, H.; Xu, Q.; Lan, X. CNTR-YOLO: Improved YOLOv5 Based on ConvNext and Transformer for Aircraft Detection in Remote Sensing Images. *Electronics* **2023**, *12*, 2671. [CrossRef]
27. Li, W.; Zhang, L.; Wu, C.; Cui, Z.; Niu, C. A new lightweight deep neural network for surface scratch detection. *Int. J. Adv. Manuf. Technol.* **2022**, *123*, 1999–2015. [CrossRef] [PubMed]
28. Gao, J.; Tan, F.; Cui, J.; Ma, B. A Method for Obtaining the Number of Maize Seedlings Based on the Improved YOLOv4 Lightweight Neural Network. *Agriculture* **2022**, *12*, 1679. [CrossRef]
29. Liu, S.; Jin, Y.; Ruan, Z.; Ma, Z.; Gao, R.; Su, Z. Real-Time Detection of Seedling Maize Weeds in Sustainable Agriculture. *Sustainability* **2022**, *14*, 15088. [CrossRef]
30. Kaya, Y.; Gürsoy, E. A novel multi-head CNN design to identify plant diseases using the fusion of RGB images. *Ecol. Inform.* **2023**, *75*, 101998. [CrossRef]
31. Zhao, Y.; Sun, C.; Xu, X.; Chen, J. RIC-Net: A plant disease classification model based on the fusion of Inception and residual structure and embedded attention mechanism. *Comput. Electron. Agric.* **2022**, *193*, 106644. [CrossRef]
32. Song, C.; Zhang, F.; Li, J.; Zhang, J. Precise maize detasseling base on oriented object detection for tassels. *Comput. Electron. Agric.* **2022**, *202*, 107382. [CrossRef]
33. Minns, A.W.; Hall, M.J. Artificial neural networks as rainfall-runoff models. *Hydrol. Sci. J.* **1996**, *41*, 399–417. [CrossRef]
34. DeVries, T.; Taylor, G.W. Improved regularization of convolutional neural networks with cutout. *arXiv* **2017**, arXiv:1708.04552.
35. Zhong, Z.; Zheng, L.; Kang, G.; Li, S.; Yang, Y. Random erasing data augmentation. In Proceedings of the AAAI Conference on Artificial Intelligence, Hilton, NY, USA, 7–12 February 2020; Volume 34, pp. 13001–13008.
36. Zhang, H.; Cisse, M.; Dauphin, Y.N.; Lopez-Paz, D. mixup: Beyond empirical risk minimization. *arXiv* **2017**, arXiv:1710.09412.
37. Smith, J.M. Analyzing the mosaic structure of genes. *J. Mol. Evol.* **1992**, *34*, 126–129. [CrossRef]
38. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, CB, Canada, 18–22 June 2023; pp. 7464–7475.
39. Tan, M.; Le, Q. EfficientNet: Rethinking model scaling for convolutional neural networks. In Proceedings of the International Conference on Machine Learning (ICML), Long Beach, CA, USA, 9–15 June 2019; pp. 6105–6114.
40. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. Scaled-YOLOv4: Scaling cross stage partial network. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 21–25 June 2021; pp. 13029–13038.
41. 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; pp. 1580–1589.
42. 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; pp. 3–19.
43. Gevorgyan, Z. SIoU Loss: More Powerful Learning for Bounding Box Regression. *arXiv* **2022**, arXiv:2205.12740.
44. Castiglioni, I.; Rundo, L.; Codari, M.; Di Leo, G.; Salvatore, C.; Interlenghi, M.; Gallivanone, F.; Cozzi, A.; D'Amico, N.C.; Sardanelli, F. AI applications to medical images: From machine learning to deep learning. *Phys. Med.* **2021**, *83*, 9–24. [CrossRef]