

Article

One-Stage Disease Detection Method for Maize Leaf Based on Multi-Scale Feature Fusion

Ying Li ¹, Shiyu Sun ², Changshe Zhang ³, Guangsong Yang ^{3,*}  and Qiubo Ye ^{3,*} ¹ Chengyi University College, Jimei University, Xiamen 361021, China² School of Computer Engineering, Jimei University, Xiamen 361021, China³ School of Ocean Information Engineering, Jimei University, Xiamen 361021, China

* Correspondence: gsyang@jmu.edu.cn (G.Y.); qbye@jmu.edu.cn (Q.Y.)

Abstract: Plant diseases such as drought stress and pest diseases significantly impact crops' growth and yield levels. By detecting the surface characteristics of plant leaves, we can judge the growth state of plants and whether diseases occur. Traditional manual detection methods are limited by the professional knowledge and practical experience of operators. In recent years, a detection method based on deep learning has been applied to improve detection accuracy and reduce detection time. In this paper, we propose a disease detection method using a convolutional neural network (CNN) with multi-scale feature fusion for maize leaf disease detection. Based on the one-stage plant disease network YoLov5s, the coordinate attention (CA) attention module is added, along with a key feature weight to enhance the effective information of the feature map, and the spatial pyramid pooling (SSP) module is modified by data augmentation to reduce the loss of feature information. Three experiments are conducted under complex conditions such as overlapping occlusion, sparse distribution of detection targets, and similar textures and backgrounds of disease areas. The experimental results show that the average accuracy of the MFF-CNN is higher than that of currently used methods such as YoLov5s, Faster RCNN, CenterNet, and DETR, and the detection time is also reduced. The proposed method provides a feasible solution not only for the diagnosis of maize leaf diseases, but also for the detection of other plant diseases.

Keywords: deep learning; plant diseases; leaf detection; multi-scale feature fusion



Citation: Li, Y.; Sun, S.; Zhang, C.; Yang, G.; Ye, Q. One-Stage Disease Detection Method for Maize Leaf Based on Multi-Scale Feature Fusion. *Appl. Sci.* **2022**, *12*, 7960. <https://doi.org/10.3390/app12167960>

Academic Editor: Rocco Furferi

Received: 6 July 2022

Accepted: 5 August 2022

Published: 9 August 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Plant diseases are one of the main factors that affect plant growth, and the detection and identification of plant diseases are the keys to early diagnosis and the precise control of pests and diseases. Maize is the world's top-producing food crop. When encountering diseases, maize plants infected with viruses and fungi produce physiological lesions. The infected parts of the leaves show characteristics such as deformation, discoloration, curling, rotting, and discoloration. There is a need for the quick, easy, and accurate detection of plant disease areas and identification of the disease species. Traditional leaf detection methods by manual observation and judgment of maize leaves require extensive practical experience and professional knowledge, which are time-consuming and have a high cost and high false detection rate.

In recent years, deep learning has been widely used in various fields such as face recognition, intelligent transportation, and automatic driving. Deep learning applied to plant disease detection and identification can overcome the drawbacks of traditional diagnostic methods and significantly improve the accuracy of disease detection and identification, and it has attracted widespread attention [1,2].

Girshick et al. proposed R-CNN [3], which uses a convolutional neural network (CNN) to extract image features for plant disease detection. The proposed Fast R-CNN is based on R-CNN [4] to solve the problem of large numbers of overlapping boxes in the process of

candidate region selection in R-CNN. Faster R-CNN is widely used in the disease detection of grapes [5] and rice [6] due to its outstanding detection accuracy. However, Faster R-CNN is a two-stage detector, and the computational effort of selecting candidate frames is heavy, which leads to its slow detection speed and makes it difficult to achieve real-time detection.

Redmon et al. proposed the YoLo method [7], which is based on one-stage detection and does not require the generation of a proposal box, but it divides the image into a grid to determine target boundaries and classes, which improves the detection speed and alleviates the real-time detection problem compared with Faster R-CNN. YoLo-based methods are also widely used in plant disease detection, such as YoLov3-based tea disease detection [8] and YoLov4-based citrus disease detection [9], but YoLo is not accurate enough in the detection and localization of small targets.

Zhou et al. [10] proposed CenterNet, a detection algorithm without anchor frames, which removes the operation of generating anchor frames and saves some time-consuming operations by estimating the loss from the heat map, thus improving the detection speed. Albattah et al. [11] improved CenterNet by extracting deep-seated key points based on DenseNet-77 and classifying and recognizing 26 kinds of plant diseases in 14 plants, such as tomatoes, apples, grapes, etc., but the detection effect of small plant diseases was not ideal.

Rashid et al. performed potato disease detection [12] based on YoLov5 using multi-scale pooling and feature pyramid up- and down-sampling to enhance contextual linguistic features to accommodate multi-scale plant diseases and directly improve small plant disease accuracy. It also combined the advantages of the accuracy of the anchor-less frame algorithm and the detection speed of the single-segment algorithm, and the detection effect is remarkable.

Scholars have undertaken a lot of research on the target detection and classification and recognition of plant leaf diseases based on deep learning technology, as shown in Table 1. In practical application scenarios, plant disease detection and recognition still face many challenges. The main reasons are as follows:

- (1) The change of illumination makes it difficult to locate the target area accurately. Due to the change in light intensity and reflection, as well as other reasons, it is difficult to accurately locate the diseased area in some detection images. Even under the same light intensity, the shooting angle and height may cause the color depth of the diseased area to be different, making the disease characteristics not significant, and thus affecting the detection accuracy.
- (2) The complex background makes it difficult to detect the target accurately. The image background of plant leaf disease is complex and may include leaves, trunks, weeds, fallen leaves, shadows, etc. The color and shape of the plant disease may be similar to other objects in the background, resulting in an increased difficulty of target detection.
- (3) Occlusion leads to missing target features and overlapping noise. Occlusion problems include blade occlusion caused by blade attitude changes, branch occlusion, light occlusion caused by external illumination, and mixed occlusion caused by different occlusion types. Due to occlusion, feature deletion and noise overlap lead to false detection or even missed detection.
- (4) The sparse target distribution affects the detection accuracy. Due to the limitation of the convolution receptive field, the connection between target pixels with sparse distribution is not strong, and the context extraction is not sufficient, which leads to the failure of modeling, thus affecting the detection accuracy.

This paper proposes a multi-scale feature fusion convolutional neural network (MFF-CNN) for the disease detection of maize leaves based on the anchor-free frame one-stage plant disease detection method, with the addition of the CA attention module and improvement of the SSP module based on YoLov5s.

The main contributions of this paper are as follows:

- (1) We add a coordinate attention (CA) module to the backbone network and increase the weight of key features to strengthen the effective information of the feature map.

- (2) We improve the spatial pyramid pooling (SPP) module to reduce the loss of feature information caused by traditional pooling.
- (3) We solve the problem of the insufficient dataset through data enhancement, enrich the training data, improve the generalization performance and robustness of the model, and prevent overfitting.

Table 1. Summary of related work on plant disease detection.

Plant Type	Dataset	Strength	Detection Network Framework	References	Year
Paddy crops	1500	Better accuracy	Mask R-CNN	Anandhan, Singh, etc. [13]	2021
Sugarcane	2940	Higher accuracy	Faster-RCNN	Kumar [14]	2021
Papaya	2000	Propose the use of lighter versions of YOLO which are more efficient and have high detection speed	YOLO	Maski, Thondiyath [15]	2021
Citrus	392	Higher accuracy	YoLov4	Li Hao, etc. [16]	2021
Apple	1200	Better accuracy	Mask RCNN	Rehman etc. [17]	2021
Grape	4500	Better accuracy	GLDDN	Dwivedi, etc. [5]	2021
Tomatoes	2385	Efficient and precise	MobileNetv2-YoLov3	Liu and Wang [18]	2020
Grape	4449	Higher accuracy and a satisfactory detection speed	Faster DR-IACNN	Xie, etc. [19]	2020
Cassava	2415	Deploy the model in a mobile application and test its performance on mobile images and video	SSD	Ramcharan, etc. [20]	2019
Tea	4000	Identify an accurate yet efficient detector in terms of speed and memory	YOLOv3	Bhatt, etc. [8]	2019

2. Materials and Methods

2.1. Deep Learning-Based Plant Disease Detection Technology

Plant disease detection technology uses computer vision technology to detect plant disease-infested areas and their exact locations under complex natural conditions, which is a prerequisite for the accurate classification and identification of plant diseases and the assessment of disease damage levels.

Early plant disease algorithms used a sliding window strategy to select candidate regions, then extract candidate region features, and, finally, using a classifier, classify them to obtain the target regions. This method traverses the image by setting different scales and widths. Although this method does not miss any disease region target, the ensuing redundant candidate windows bring great computational effort and it is very time-consuming to traverse the disease image all over again, resulting in poor real-time detection [21].

With the rapid development of artificial intelligence technology, different techniques are based on artificial vision for digital image processing and the implementation of the image classification model. Reference [22] proposed methodology that consists of five stages, as shown in Figure 1, i.e., image acquisition, preprocessing, segmentation, feature extraction, and classification, to find the damages caused by the cogollero worm in corn fields.

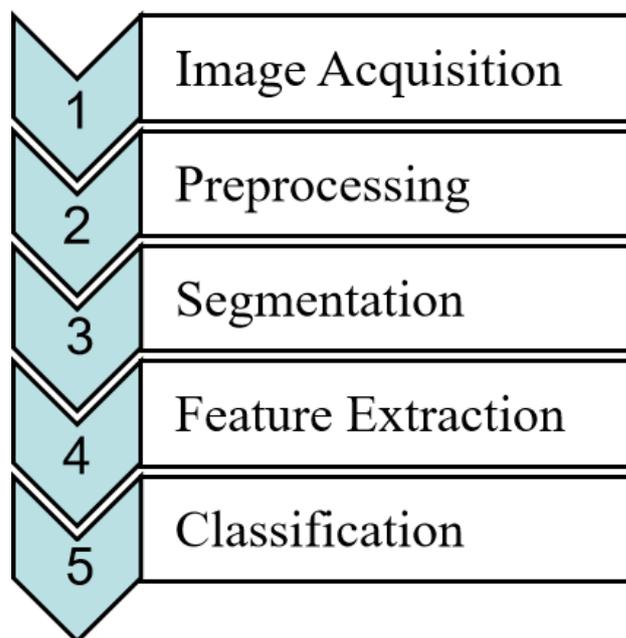


Figure 1. Methodology implemented for the recognition of the damage caused by the cogollero worm [14].

Moreover, many deep learning-based methods can be applied to plant disease diagnosis, as shown in Figure 2.

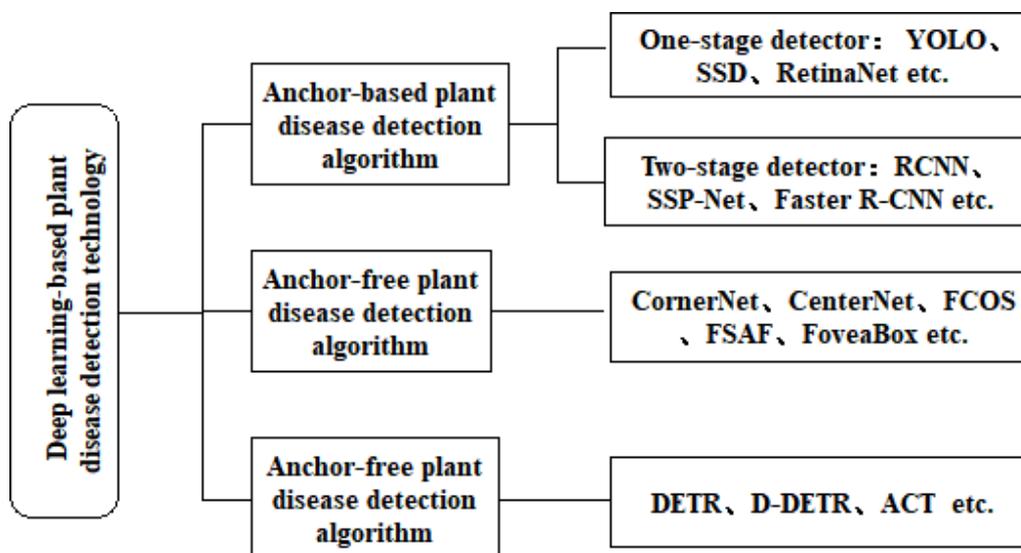


Figure 2. Deep learning-based plant disease detection technology.

2.1.1.1. Anchor-Based Plant Disease Detection Algorithm

The anchor-based plant disease detection algorithm adopts the detection idea of an anchor plus a priori box, setting a priori boxes (anchor-box) with different aspect ratios at each feature point of the plant feature map for screening and adjustment to obtain the final prediction box. Due to the redundant computation of the anchor-box, anchor-based plant disease detection is slow.

Anchor-based plant disease detection frameworks can be divided into two categories: two-stage detectors and one-stage detectors [23], as shown in Figure 3.

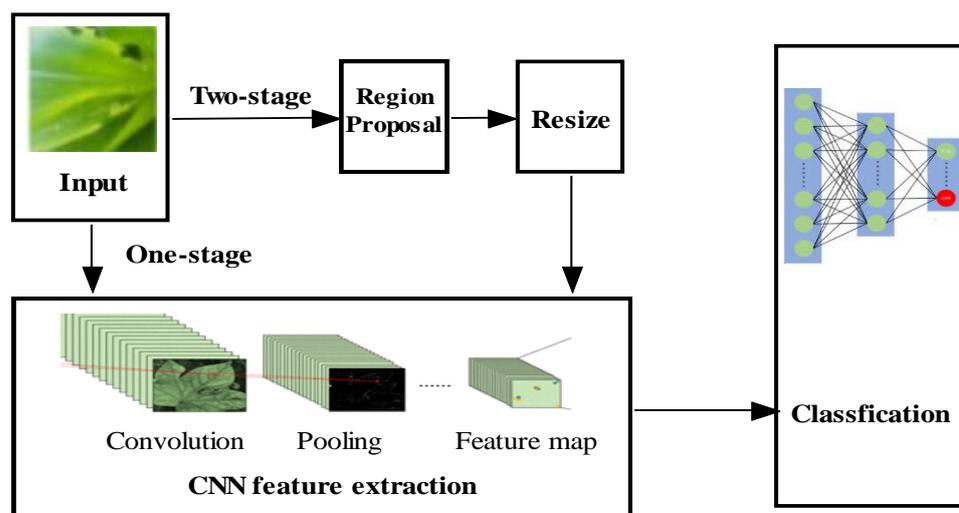


Figure 3. Anchor-based detection method.

Two-Stage Detector

The main plant disease algorithms based on a two-stage detector are RCNN [3], SPP-Net [24], Fast R-CNN [4], Faster R-CNN [25], etc. The two-stage detector utilizes two networks to implement classification and regression, respectively. A feature extractor (backbone) is used to generate a series of proposal boxes that may contain plant disease targets to be detected, and then some filtering rules are applied to filter the proposal boxes and identify the disease targets.

Taking Faster-RCNN as an example, firstly, the images are fed into the feature extractor for feature extraction, and the extracted features are fed into the region proposal network (RPN) to generate candidate frames. Secondly, the final proposal boxes are filtered according to the results of the classification and regression. Next, features are extracted from the suggestion frames in the feature map, and each feature is input to the pooling layer of the region of interest (ROI) and unified into a 7×7 size. Then, it is transformed into a one-dimensional vector by a fully connected layer. Finally, a classification and regression task is carried out to further correct the proposal box and determine the specific class of targets. The two-stage detector is more advantageous in the detection accuracy and classification precision of plant diseases.

One-Stage Detector

One-stage detector-based plant disease detection algorithms mainly comprise YoLo [26], SSD [27], and RetinaNet [28]. A one-stage detector accomplishes the classification and localization of plant disease targets in a network and extracts features directly from the network for plant disease category and location prediction.

Anchor-based plant disease algorithms have been dominating the field of plant disease detection, beginning with the initial RCNN, with the rapid development of algorithms with decreasing numbers of parameters and an increasing speed and accuracy of plant disease detection.

2.2. Anchor-Free Plant Disease Detection Algorithms

The anchor-free plant disease detection algorithms mainly include YoLo, CornerNet [29], FSAF [30], FoveaBox [31], and CenterNet [32]. They abandon the idea of the prior bounding box and adopt the detection idea of key point prediction to obtain the final plant disease prediction box. These algorithms have a small number of network parameters, small calculations, and fast plant disease detection, but their accuracy is not very high.

2.2.1. YoLo

The main versions of YoLo are YoLov3 and YoLov5. YoLov3 first compresses the image size to 416×416 and extracts feature maps of the same size through the feature extraction network, then it divides the image into 13×13 grids and locks the grid to predict the certain target once the center coordinate of a target in the ground truth falls in a grid. Each grid corresponds to three anchors, predicts three bounding boxes, and outputs three feature maps at different scales. YoLov3 uses multiple independent logical classifiers for object prediction to calculate the likelihood of belonging to a specific label while using a binary cross-entropy loss for each label when calculating the classification loss, reducing the complexity of the computation.

YoLov5 adds a new focus module (Focus) to YoLov4 [33] to reduce the information loss during the under-sampling operation. In addition, the number of anchors of positive samples is increased to improve the convergence speed.

2.2.2. CenterNet

CenterNet is a detection algorithm based on key points estimation, which enables the detection of disease targets by estimating the center point or corner point. CenterNet improves on CornerNet by detecting an additional key point in addition to a pair of corner points, enhancing the ability to synthesize information about the target as a whole. As a result, CenterNet's detection speed and accuracy are considerably improved compared to the frames with both one-stage and two-stage detectors. CenterNet was proven to be applicable to plant disease detection under natural conditions. Xia et al. [34] performed the detection of apples through CenterNet's detection network combined with MobileNet v3, and the detection speed and accuracy were superior to SSD. However, there is still the problem of inaccurate matching of key points in intensive targets, and the results are less satisfactory for small target diseases of plants.

2.3. Transformer-Based Plant Disease Detection Algorithm

CNN-based target detection algorithms (such as Faster RCNN, YoLo, FCOS [35], etc.) usually rely on a lot of manual design, such as the rules-based label matching mechanism, inspirational reprocessing processing, etc. An end-to-end concise target detection framework based on a transformer [36] was proposed, which has good detection performance.

The transformer is a new neural network structure that mainly uses an attention mechanism to capture global contextual information and achieves long-range information fusion to extract more effective feature extraction. The transformer has had great success in natural language processing.

Carion et al. first proposed the transformer-based detection transformer (DETR) [37]. DETR treats target detection as a simple set prediction problem, removes the NMS and anchor design, has a concise pipeline, and introduces an attention mechanism to enhance feature representation, enabling simple and complete end-to-end target detection. This algorithm has high feature fusion capability and a high accuracy of detection, but the cost of training is significant.

DETR extracts maize leaf image features using CNN networks and compresses the feature dimension into one dimension. The features are encoded at a fixed position before being fed into an encoder–decoder converter. The decoder uses a multi-head attention mechanism to decode N objects in parallel at each decoding layer to produce N outputs. Finally, the target class recognition and bounding box regression are performed by feedforward neural networks (FFNs) to achieve disease target detection.

2.4. The Method Proposed in This Paper

The design framework is divided into the following stages, as shown in the activity diagram in Figure 4. First, obtain the image dataset of maize leaf disease in Kaggle and enhance the data. Secondly, load the relevant model parameters for pre-training. Thirdly,

the MFF-CNN model is obtained through multiple training. Then input the maize leaf data to test the disease area, and, finally, receive the maize leaf disease detection results.

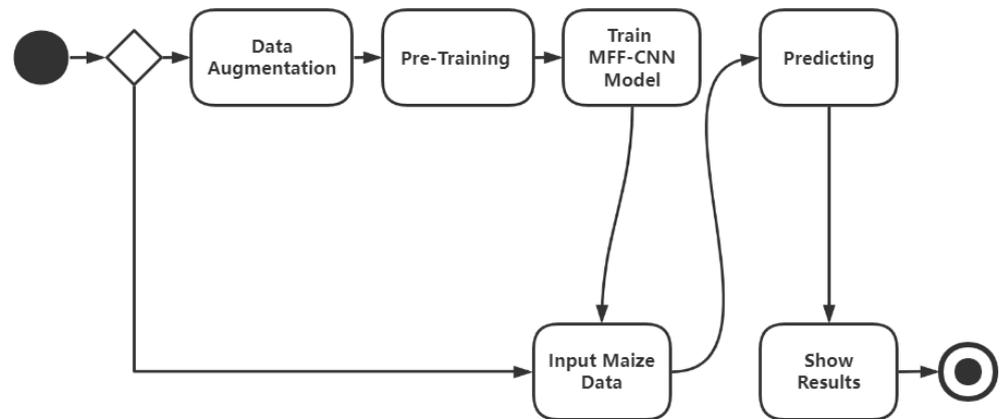


Figure 4. Activity diagram of the proposed network.

2.4.1. Network Structure

A one-stage plant disease detection method based on MFF-CNN is proposed based on YoLov5s with a one-stage detector, as shown in Figure 5. The MFF-CNN consists of three parts, i.e., the backbone, neck, and detection head.

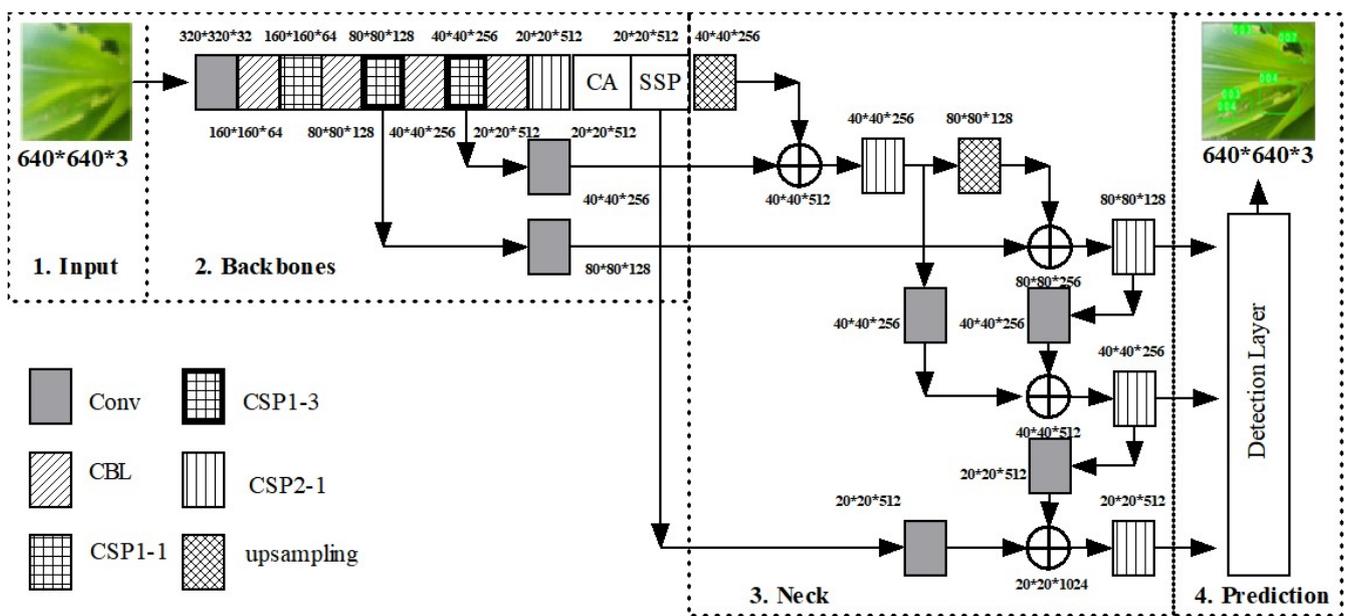


Figure 5. The structure of the one-stage maize leaf detection network based on multi-scale feature fusion.

Backbone

The backbone is based on CSP Darknet53 and mainly uses the Conv module, CBL module, and CSP1_X module to obtain the disease information characteristics of maize leaves.

The Conv module consists of a convolutional layer, a batch normalization operation, and a SiLU activation function. Its kernel size is 3×3 , and the step size is 2. The CBL module is similar to the Conv module in that it also uses convolutional layers with batch normalization operations, except that it uses Leaky ReLU as the activation function.

The MFF-CNN uses two different CSP structures. The CSP1-X with residual components (X bottlenecks) in the backbone is shown in Figure 6, while the neck uses convolutional layers (X CBLs) instead of residual components, shown as CSP2-X in Figure 7. The

cross-layer design of the CSP reduces computation, improves inference speed, reduces memory cost, and guarantees accuracy.

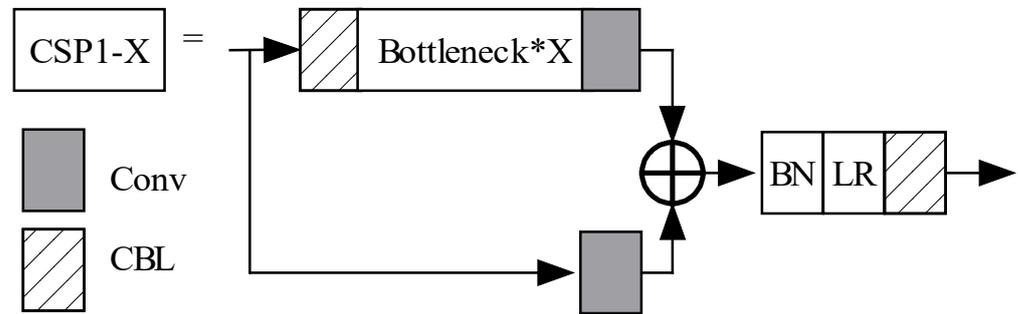


Figure 6. CSP1-X module.

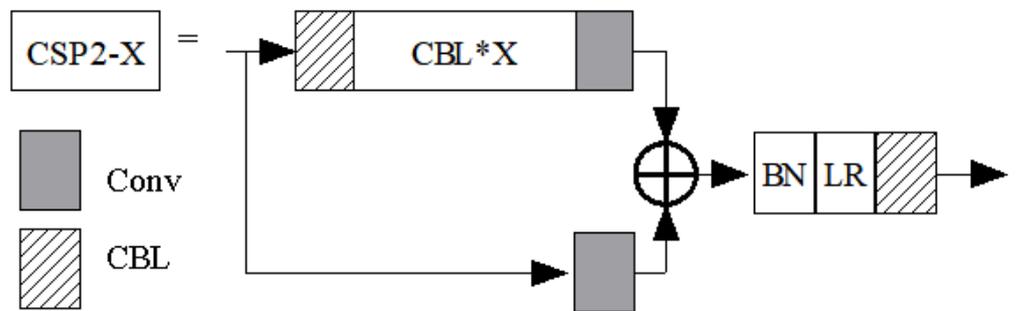


Figure 7. CSP2-X module.

The MFF-CNN model also adds the coordinate attention (CA) attention module (see Section 2.4.2) and the spatial pyramid pooling (SSP) improvement module (see Section 2.4.3) to the backbone.

Neck

The neck adopts the feature pyramid network (FPN) [38] and path aggregation network (PAN) [39]. The MFF-CNN borrows from PAN and adds a bottom-up feature pyramid network after sampling on the FPN for feature fusion. The FPN extracts stronger semantic information from the top-down, while the PAN extracts stronger localization information from the bottom-up, thus fusing the feature maps of the different layers of the CNN and strengthening the feature information extraction ability.

Three feature maps of different sizes with rich semantic information are obtained after three different concatenation operations to meet the needs of plant disease target detection at different scales. Finally, the CSP2-1 operation is added to each of the three feature maps and then sent to the detection end.

Detection Head

The detection network uses three detection heads with GIOU-LOSS as the loss function and outputs three scales of feature maps with 20 × 20, 40 × 40, and 80 × 80 grids for detecting small, medium, and large maize disease targets, respectively. Each grid contains three prediction boxes, each containing information about the confidence of the object and the position of the prediction box. Maize disease detection is accomplished at the detection head by non-maximal suppression (NMS) [40] as post-processing to eliminate duplicate redundant prediction frames and retain the prediction frame with the highest confidence.

Intersection over union (IOU) is a metric for evaluating the accuracy and can be expressed as:

$$IOU = \frac{|A \cap B|}{|A \cup B|} \tag{1}$$

The MFF-CNN for border regression uses complete-IOU (CIOU) instead of IOU [41] for model training. CIOU is expressed as follows:

$$CIOU = IOU - \frac{\rho^2(b, b^{gt})}{c^2} - \alpha v \tag{2}$$

Among them, a denotes ground truth, B denotes the predicted frame, and $\rho^2(b, b^{gt})$ represents the Euclidean distance between the centroids of the prediction box and the ground truth.

The LOSS when CIOU is regressed can be calculated as:

$$LOSS_{CIOU} = 1 - CIOU = 1 - IOU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v \tag{3}$$

where α and v are denoted as follows:

$$\alpha = \frac{v}{1 - IOU + v} \tag{4}$$

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

2.4.2. Coordinate Attention

In this paper, the weight of key features is enhanced by adding coordinate attention (CA) [42] in the backbone to select the feature information extracted from the backbone, as shown in Figure 8 and Algorithm 1.

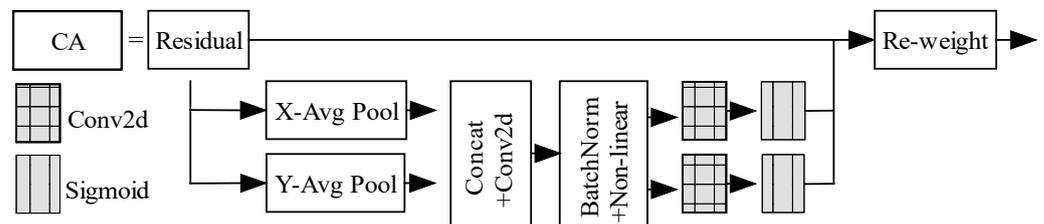


Figure 8. Coordinate attention module.

Algorithm 1: Coordinate Attention

Input: Feature points in the $C \times H \times W$ dimensions of the feature map.

Output: Attentional activation feature map with the three feature dimensions $C \times H \times W$.

1. First, conduct adaptive average pooling along the H direction and W direction, accordingly, to obtain $C \times 1 \times W$ and $C \times H \times 1$ scale feature maps, respectively.
 2. The two feature maps are then concatenated and convolved to obtain the $C/r \times 1 \times (W + H)$ feature map.
 3. Perform BatchNorm and non-linear regression operations
 4. Separately perform Sigmoid activation function operations.
 5. The original input feature map and the output two feature maps are performing matrix multiplication.
 6. Finally, the $C \times H \times W$ feature map is output.
-

In order to enhance the effective features of the feature map, the coordinate attention module embeds the location information into the attention of the channel in the following two steps: coordinate information embedding and coordinate attention generation.

Coordinate Information Embedding

In Figure 8, the coordinate information is embedded into the CA module through the X average pool and the Y average pool. Global pooling is often used for global encoding as it compresses global spatial information into channel descriptors, and it is difficult to

preserve location information. To capture more precise location information, the CA module converts the global pooling into a one-to-one feature encoding operation as established by Equation (6):

$$Z_c = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W x_c(i, j) \tag{6}$$

Specifically, given the input $x_c(i, j)$, each channel is first encoded along the horizontal and vertical coordinates using a pooling kernel of size $(H, 1)$ or $(1, W)$. The output of the c channel of height h can be written as:

$$z_c^h(h) = \frac{1}{W} \sum_{0 \leq i < W} x_c(h, i) \tag{7}$$

Similarly, the output of the c channel of width w can be written as:

$$z_c^w(w) = \frac{1}{H} \sum_{0 \leq i < H} x_c(i, w) \tag{8}$$

The above two transformations, i.e., the X average pool and Y average pool of Figure 7, correspond to two aggregated features of the spatial directions of x and y , thus obtaining a pair of feature maps with perceptual capabilities in different directions. The two transformations enable spatial attention to obtain long-term dependencies along one spatial direction and to preserve precise position information along another spatial direction. This addresses the difficulty of global pooling to preserve location information and helps the network to locate the target of interest more accurately.

Coordinate Attention Generation

To take advantage of the embedded information in Section 3.2.1, the CA module concatenates them and then transforms them using a shared 1×1 convolutional transform function, as follows:

$$f = \delta \left(F_1 \left(\left[z^h, z^w \right] \right) \right) \tag{9}$$

where, $[*, *]$ denotes a concatenation operation along the spatial dimension, δ is a nonlinear activation function, $f \in \mathbb{R}^{C/r \times (H+W)}$ is a feature map encoding spatial information in the horizontal and vertical directions, and r is the reduction ratio.

Subsequently, along the spatial dimension, f are cut into two independent tensors, $f^h \in \mathbb{R}^{C/r \times H}$ and $f^w \in \mathbb{R}^{C/r \times W}$. The feature maps f^w and f^h are then transformed to the same number of channels as input $x_c(i, j)$ using the two convolutional transforms F_h and F_w , respectively, with the following results:

$$g^h = \sigma \left(F_h \left(f^h \right) \right) \tag{10}$$

$$g^w = \sigma \left(F_w \left(f^w \right) \right) \tag{11}$$

Then, expanding g^h and g^w as attention weights, the final output of the CA module is as follows:

$$y_c(i, j) = x_c(i, j) \times g_c^h(i) \times g_c^w(j) \tag{12}$$

2.4.3. SSP Improvement

Both maximum pooling and average pooling will lose some of the feature information of the image during the process of pooling and reduce the performance of the whole network. To solve this problem, we improve the spatial pyramid pooling (SPP) module in the YoLov5s network, as shown in Figure 9. The SSP module has a spatial pyramidal pooling structure, including three SoftPools [43] and a Skip Connection. SoftPool maps all pixels within the receptive field to the next network layer in a Softmax-weighted

summation. The SSP improvement module down-samples the feature map while retaining more fine-grained plant information in the feature map, thus reducing information loss.

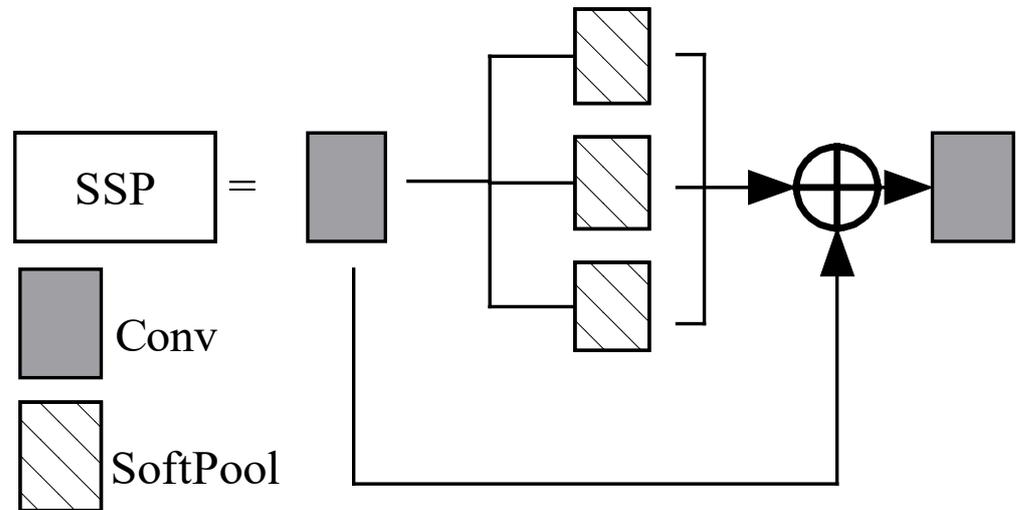


Figure 9. SSP improvement module.

This method achieves the fusion of global and local features, thus enriching the expression capability of the feature map, and it is suitable for the feature extraction network of plant diseases.

Suppose the activation value of each pixel in the SoftPool corresponding to the receptive field is a_i and the other activation values in the kernel region R are a_j . Then, the weight w_i of each pixel in the receptive field is as follows:

$$w_i = \frac{e^{a_i}}{\sum_{j \in R} e^{a_j}} \quad (13)$$

After SoftPool, we can obtain the standard summation output value \tilde{a} of all weighted activations in R , as follows:

$$\tilde{a} = \sum_{i \in R} W_i * a_i \quad (14)$$

3. Experimentation and Performance Evaluation

To examine the effectiveness of the MFF-CNN in maize leaf detection, we present a comparative analysis of the maize leaf detection results of five plant disease algorithms, YoLov5s, DETR, CenterNet, Faster RCNN, and MFF-CNN, based on the maize leaf dataset.

3.1. Dataset and Parameter Settings

There are 2265 maize leaf data in the dataset [44] with image resolutions of 2448×3264 and 3456×4608 . To prevent over-fitting and, at the same time, to improve the robustness of the maize leaf detection network, the data are enhanced with the help of flip transform, random clipping, and scale transform to expand the maize leaf dataset to 6795, and they take an RGB image format. The dataset is in PASCAL VOC [45,46], and it is divided into a training set, a validation set, and a test set according to the ratio of 8:1:1.

To enhance the robustness of the leaf detection network, we use 5437 images of maize leaves with location annotations for training the maize leaf detection model. The detection result is considered correct when the intersection over union (IOU) between the prediction box and the truth box is greater than 0.5. To verify the effectiveness of the MFF-CNN, we selected four mainstream plant disease algorithms for comparison, including Faster R-CNN based on the two-stage detector, YoLov5s based on the one-stage detector, anchor-free CenterNet, and transformer-based DETR. The Faster R-CNN uses a Resnet50 model pre-trained on ImageNet [47] as the backbone with a learning rate of 0.00001, an input

image size of 600×600 , and a training batch of two. CenterNet also uses Resnet50 as the backbone with a learning rate of 0.00001, an input image size of 512×512 , and a training batch of eight. DETR uses Resnet50 as the backbone with a learning rate of 0.00001, input image sizes of 2448×3264 and 3456×4608 , and a training batch of two. CenterNet, Faster-RCNN, and DETR all have an epoch of 300, and YOLO v5 and the MFF-CNN have an epoch of 150.

YoLov5s uses Darknet53 as the backbone with a learning rate of 0.00001, an input image size of 640×640 , and a training batch of 16. The MFF-CNN proposed in this paper uses New CSP-Darknet53 as the backbone with an initial learning rate of 0.00001, an input image size of 640×640 , and a training batch of 16. The graphics card for this experiment is RTX3080ti, with Toolkit CUDA of version 11. and deep neural networks GPU-accelerated library CUDNN of version 8.0.4 developed by NVIDIA Corporation. We installed PyTorch 1.8.2 + cu111 developed by Facebook AI Research, and open source python 3.8.12 on a Linux system.

3.2. Experimental Results and Analysis

The quantitative analysis of all methods in this paper is conducted based on the same indicators and the same dataset.

The mean average precision (mAP) is an essential performance evaluation metric for target detection models. The mAP here is the average of the average precision (AP) of detection calculated for each category when IOU = 0.5. The experimental results are shown in Table 2, and the MFF-CNN has the best performance, in terms of mAP, compared to the others' maize disease detection. It is 10.4% higher than Faster R-CNN and 5.3% higher than CenterNet, and it also outperforms YoLov5s and DETR.

Table 2. Comparison of mAP, detection time, and FLOPs.

Algorithm	Mean Average Precision (mAP)	Detection Time/s	FLOPs
MFF-CNN	0.486	0.039	4.2
YoLov5s	0.47	0.017	16.5
DETR	0.467	2.054	76.5
CenterNet	0.433	1.222	34.97
Faster RCNN	0.382	0.409	256.3

There are two main reasons that the MFF-CNN achieves the best detection performance in maize leaves. One is that the CA channel attention module enhances the feature information of the detected targets. In particular, it enhances the feature information of small targets, overlapping obscured targets and fuzzy targets, which makes the detection better. Another reason is the application of Softpool, which uses down-sampling to reduce the amount of data while retaining as much feature information as possible, thus preventing the loss of maize leaf detection information with blurred edges and corners. These two modules significantly improve the detection accuracy and efficiency of the MFF-CNN model in maize leaf detection.

To verify the superiority in temporal performance of the MFF-CNN, we use detection time, i.e., the time to complete an image, to compare the time performance. To get the detection time, we loop through 1000 corn leaf images for detection and then calculate the average inference time between one image input to the network model and the output model. The detection time of the MFF-CNN is 0.039 s, which is not as good as YoLov5s, but its detection efficiency is faster than DETR, CenterNet, and Faster RCNN by at least 0.37 s. Since the FOCUS module reduces the image size and the computation, it greatly reduces the MFF-CNN detection time. However, as the MFF-CNN also adds a CA module and an SSP module on top of the YoLov5s network, the detection time of the MFF-CNN algorithm is slightly longer than YoLov5s.

We use floating point operations (FLOPs) to test the algorithm's complexity. The value of the FLOPs for the MFF-CNN is 4.2 G, which is much lower than the other algorithms, indicating little model complexity and a minimal computational effort.

In the detection of plant leaf diseases, missing detection is one of the important factors affecting the accuracy of disease detection. This section uses sensitivities to test the missed detection rate of the classifier model [48,49]. Sensitivity describes the ratio of true positive (TP) to actual positive (TP plus FN), where FN is a false negative. The higher the sensitivity, the lower the missed detection rate.

Table 3 compares the performance of the proposed MFF-CNN versus the other algorithms in terms of sensitivities (%) at various false positives (FPs) per image.

Table 3. Sensitivities (%) at various false positives (FPs) per image.

Methods	0.5	1	2	3	4	8	Avg. [0.5, 1, 2, 4, 8]
MFF-CNN	25.59	37.05	47.93	56.13	61.27	71.01	49.83
YoLov5s	25.14	36.49	46.40	53.06	58.11	67.93	47.85
DETR	24.23	34.86	46.85	54.59	59.64	70.09	48.38
CenterNet	21.71	30.54	43.51	50.36	56.85	68.83	45.30
Faster RCNN	18.38	29.01	39.28	45.77	51.44	62.88	41.13

The numbers 0.5, 1, 2, and 4 represent the different values of FPs. As can be seen from the table, when IOU = 0.5 and there are four false positives in each picture, the sensitivity of our algorithm is 3.16% higher than Yolov5s and 9.83% higher than Faster RCNN. The results show that the MFF CNN has the highest sensitivity, i.e., the lowest miss detection rate, and it exhibits the optimal detection performance.

We analyze several specific cases in the following sections.

3.2.1. Detection of Target Area Overlap Occlusion

As shown in Figure 10, with the disease area of the maize leaves overlapped and obscured, we can see that the three algorithms (with the exception of MFF-CNN) missed the overlapping leaves in the upper left corner of the image. Both DETR and CenterNet did not perform very well in the detection of the remaining disease areas in the images.

DETR is a transformer-based target detection framework that transforms target detection into a simple set prediction problem to achieve end-to-end target detection. However, it has a long training time, misses the detection of small target diseases that appear densely in the middle of maize leaves in the experiment, and has insufficient detection accuracy for dense, small targets. CenterNet is an anchor-free method by direct prediction of the centroid coordinates of objects. When there are multiple objects in overlapping centers, they will be misidentified as one object. Further, CenterNet predicts that if the centroids of two objects also overlap in down-sampling during the prediction process, they will also be mistaken for one object. When there are dense targets in the detection area, such as maize leaves close together and overlapping occlusions, CenterNet will mismatch the key points and exhibit poor performance.

In our proposed MFF-CNN, the X average pool and Y average pool aggregate feature along two spatial directions obtains a pair of feature maps with perceptual capabilities in different directions, which helps the network to locate the target of interest more accurately. Thus, the MFF-CNN performed best in the complex case of overlapping shading of diseased areas of maize leaves, achieving a mAP of 0.486.

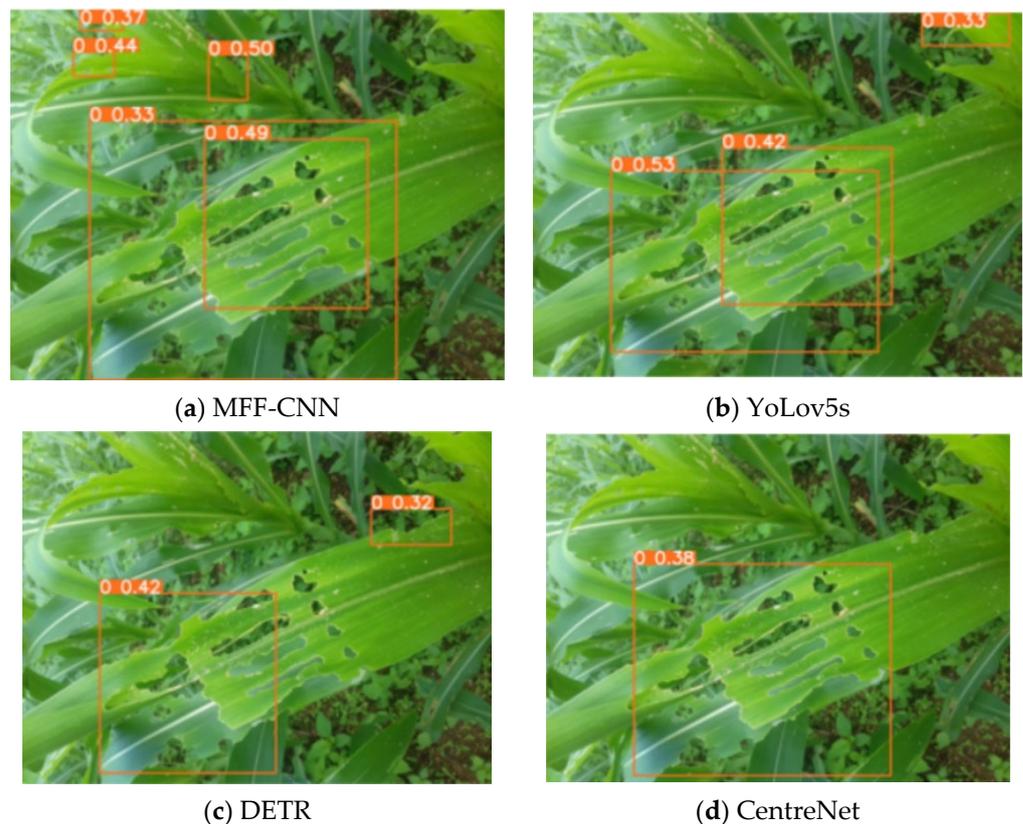


Figure 10. Detection results of the overlapping shading of diseased leaves.

3.2.2. Detection of Sparsely Distributed Targets

As can be seen in Figure 11, the diseased areas of the maize leaves were sparsely distributed, mainly concentrated in the upper right part and the lower left part of the leaves, and there was a certain distance between the two diseased areas. DETR and CenterNet mainly detected disease in the upper right part of the leaf, and there were a large number of missed detections. Although the transformer used in DETR focuses more on local key information, the transformer model often requires a large amount of data and a long training period to make the model converge. While the maize leaf dataset uses data enhancement technology to mitigate the over-fitting, the dataset for this experiment is much less massive than ImageNet, resulting in not fully meeting the data volume requirements of the transformer model. This is the main reason why DETR has a large number of missed detections in the lower left part of the leaf, as well as the leaf edges. Although YoLov5s detects most of the diseases, the detection confidence is lower than that of the MFF-CNN, and the detection effect is not satisfactory in the case of sparse disease distribution.

The MFF-CNN algorithm proposed in this paper uses SoftPool to reduce the loss of information of features in the pooling process, thus maximizing the assurance of the comprehensible extraction of disease feature information in plant leaves. Thus, the MFF-CNN performs best even in the complex case of sparse disease distribution in maize leaves.

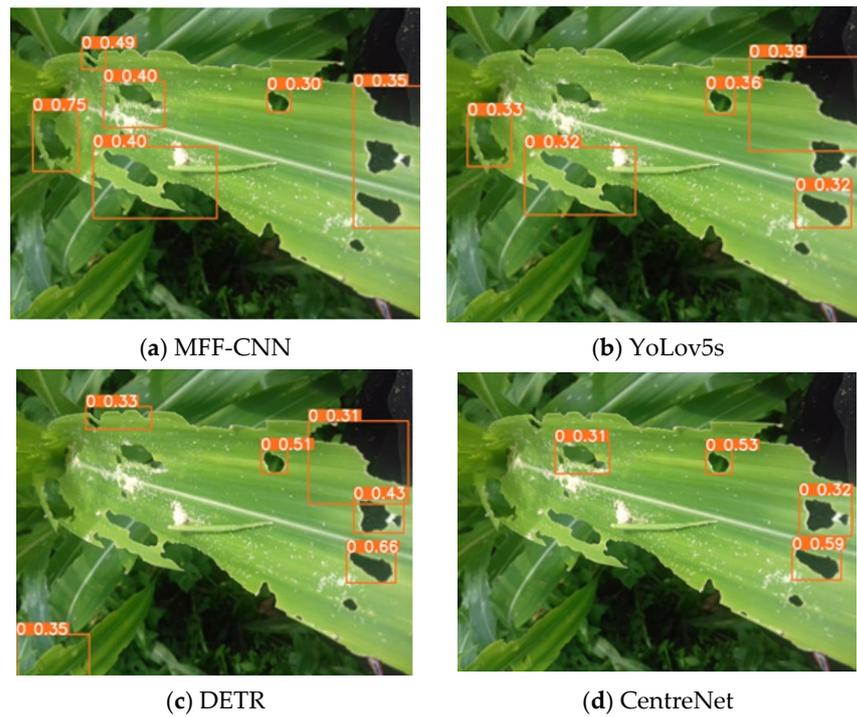


Figure 11. Detection results of the sparse distribution of diseased leaves.

3.2.3. Detection of Target and Background Texture Similarity

In Figure 12, the texture and color of the maize leaves are very similar to the background, and it is hard to detect disease. YoLov5s does not have the same detection confidence as our proposed MFF-CNN on the maize leaf dataset. The main reason is the relatively large image resolution of the maize leaf dataset used in this experiment, which makes YoLov5s ineffective at detecting small targets with inadequate multi-scale information.

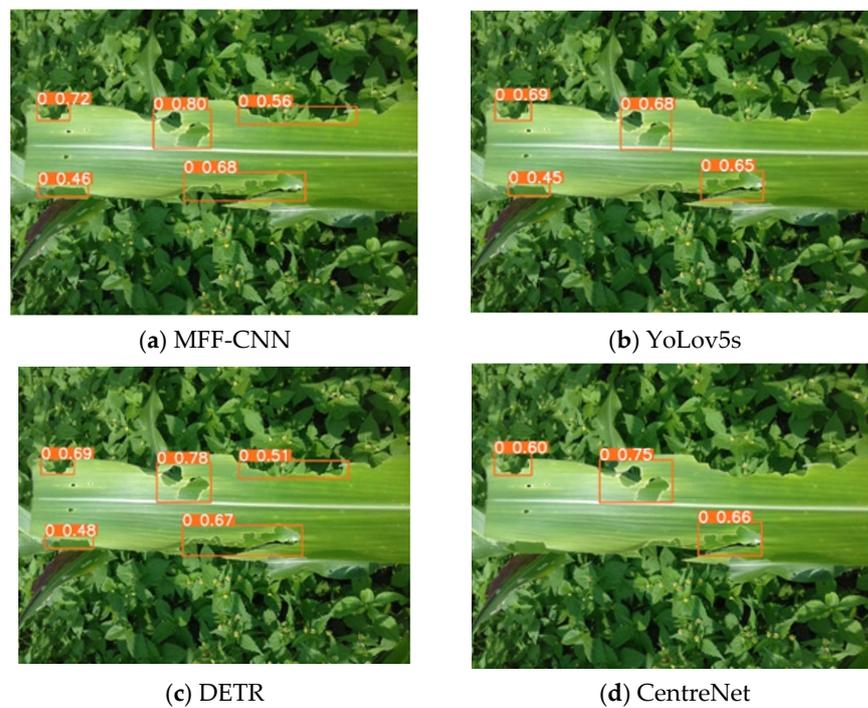


Figure 12. Detection results of similar textures and backgrounds of maize leaves.

Our proposed MFF-CNN enhances the multi-scale features of the CA module and prevents information loss as much as possible with the help of Softpool, which provides a good foundation for the detection of multi-scale targets afterward. It is experimentally demonstrated that with the joint efforts of the CA module and Softpool pooling, the MFF-CNN obtains the optimal detection accuracy, with a 1.6% improvement compared to YoLov5s and a 1.9% improvement compared to DETR.

4. Discussion

The proposed method of multi-scale feature fusion realizes the extraction of context information and the detection of maize leaf diseases. However, in the detection of edge targets and dense small targets, the prefetching of small target detection cannot be divided, and even the problem of missed detection occurs. This is because the limitations of convolution operation cannot model the image globally, resulting in insufficient global context information extraction. To solve the above problems, in the next stage of work, we use the idea of a transformer to achieve the optimization of the baseline and use a DropBlock [50] convolutional regularization method to improve the detection accuracy.

In our model, we treat the images in a simple way such that the data are enhanced with the help of flip transform, random clipping, and scale transform. In some techniques of segmentation and classification using color [51], they are capable of processing trivial features such as shadows, noise, pixel saturation, low light, different crop varieties, and intrinsic camera parameters to improve model quality. We will attempt to use these methods in our model to improve detection performance in the future.

The quality of the dataset plays an important role in the detection effect of the algorithm. In a future study, we will sample maize leaves of different varieties, fertility stages, and shooting angles under field conditions, and strictly label them to establish a large dataset of maize leaves.

Moreover, through learning and analysis of big data, we will develop and design plant disease models applicable to maize leaves in general, explore deep learning networks with better performance, improve the accuracy of maize leaf disease detection, and apply them to early plant diagnosis and intelligent monitoring.

Meanwhile, to improve the practical application capability, the MFF-CNN model was developed into a corresponding access point installed on mobile devices such as UAVs and smartphones to provide timely, accurate, and wide-range, real-time monitoring information for maize leaf disease identification. In addition, we will attempt to use the MFF-CNN for research on the detection of biotic and abiotic stresses in agriculture, such as saline stress and drought stress.

5. Conclusions

In order to realize accurate and real-time maize leaf disease detection, based on deep learning technology, this paper proposes an MFF-CNN based on multi-scale feature fusion for maize leaf disease detection.

We conducted experiments under the complex conditions of combined overlapping occlusion, sparse target distribution, and similar textures of the diseased areas and backgrounds. The results show that the proposed method obtains the best detection performance compared with the maize disease algorithms of Faster R-CNN, CenterNet, YoLov5s, and DETR.

Author Contributions: Data curation, S.S. and C.Z.; Funding acquisition, Y.L. and G.Y.; Investigation, C.Z.; Methodology, Y.L.; Supervision, Q.Y.; Validation, S.S. and Q.Y.; Writing—original draft, Y.L.; Writing—review & editing, G.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Natural Science Foundation of Fujian Province under Grant 2021J01865, the Open Project Fund of Fujian Shipping Research Institute (No. HHXY2020014), and The Education and Scientific Research Project for Middle-Aged and Young Teachers of Fujian Province (No. JAT210678 and No. JT180877).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Singh, A.K.; Ganapathysubramanian, B.; Sarkar, S.; Singh, A. Deep learning for plant stress phenotyping: Trends and future perspectives. *Trends Plant Sci.* **2018**, *23*, 883–898. [[CrossRef](#)] [[PubMed](#)]
2. Bargshady, G.; Zhou, X.; Deo, R.C.; Soar, J.; Whittaker, F.; Wang, H. Enhanced deep learning algorithm development to detect pain intensity from facial expression images. *Expert Syst. Appl.* **2020**, *149*, 113305. [[CrossRef](#)]
3. 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.
4. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; IEEE Press: New York, NY, USA, 2015.
5. Liu, T.; Feng, Q. Detecting grape leaves based on convolutional neural network. *J. Northwest Univ.* **2017**, *47*, 505–512.
6. Bari, B.S.; Islam, N.; Rashid, M.; Razman, A.; Majeed, A. A real-time approach of diagnosing rice leaf disease using deep learning-based faster R-CNN framework. *Peer J. Comput. Sci.* **2021**, *7*, e432. [[CrossRef](#)]
7. 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, Honolulu, HI, USA, 21–26 July 2016.
8. Bhatt, P.; Sarangi, S.; Pappula, S. Detection of diseases and pests on images captured in uncontrolled conditions from tea plantations. In Proceedings of the Autonomous Air and Ground Sensing Systems for Agricultural Optimization and Phenotyping IV, Baltimore, MD, USA, 15–16 April 2019.
9. Wang, C.; Luo, Q.; Chen, X.; Yi, B.; Wang, H. Citrus recognition based on YoLo4 neural network. *J. Phys. Conf. Ser.* **2021**, *1820*, 012163. [[CrossRef](#)]
10. Zhou, X.; Wang, D.; Krhenbühl, P. Objects as Points. *arXiv* **2019**, arXiv:1904.07850.
11. Albattah, W.; Nawaz, M.; Javed, A.; Masood, M.; Albahli, S. A novel deep learning method for detection and classification of plant diseases. *Complex Intell. Syst.* **2022**, *8*, 507–524. [[CrossRef](#)]
12. Rashid, J.; Khan, I.; Ali, G. Multi-level deep learning model for potato leaf disease recognition. *Electronics* **2021**, *10*, 2064. [[CrossRef](#)]
13. Anandhan, K.; Singh, A.S. Detection of Paddy Crops Diseases and Early Diagnosis Using Faster Regional Convolutional Neural Networks. In Proceedings of the 2021 International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), Greater Noida, India, 4–5 March 2021.
14. Kumar, P. Research Paper on Sugarcane Disease Detection Model. *Turk. J. Comput. Math. Educ.* **2021**, *12*, 5167–5174.
15. Maski, P.; Thondiyath, A. Plant Disease Detection Using Advanced Deep Learning Algorithms: A Case Study of Papaya Ring Spot Disease. In Proceedings of the 2021 6th International Conference on Image, Vision and Computing (ICIVC), Qingdao, China, 23–25 July 2021.
16. Li, H.; Liu, H.L.; Liu, S.L. Development of dynamic recognition system of citrus pests and diseases based on deep Learning. *J. Chin. Agric. Mech.* **2021**, *42*, 195–201, 208. [[CrossRef](#)]
17. Rehman, Z.U.; Kha Thondiyathn, M.A.; Ahmed, F.; Damaeviius, R.; Javed, K. Recognizing apple leaf diseases using a novel parallel real-time processing framework based on MASK RCNN and transfer learning: An application for smart agriculture. *IET Image Process.* **2021**, *15*, 2157–2168. [[CrossRef](#)]
18. Liu, J.; Wang, X. Early recognition of tomato gray leaf spot disease based on MobileNetv2-YOLOv3 model. *Plant Methods* **2020**, *16*, 83. [[CrossRef](#)] [[PubMed](#)]
19. Xie, X.; Ma, Y.; Liu, B.; He, J.; Wang, H. A Deep-Learning-Based Real-Time Detector for Grape Leaf Diseases Using Improved Convolutional Neural Networks. *Front. Plant Sci.* **2020**, *11*, 751. [[CrossRef](#)]
20. Ramcharan, A.; McCloskey, P.; Baranowski, K.; Mbilinyi, N.; Mrisho, L.; Ndalawha, M.; Legg, J.; Hughes, D.P. A mobile-based deep learning model for cassava disease diagnosis. *Front. Plant Sci.* **2019**, *10*, 272. [[CrossRef](#)] [[PubMed](#)]

21. Legendre, C.P.; Deschamps, F.; Zhao, L.; Chen, Q.F. Rayleigh-wave dispersion reveals crust-mantle decoupling beneath eastern Tibet. *Sci. Rep.* **2015**, *5*, 16644. [CrossRef]
22. Bravo-Reyna, J.L.; Montero-Valverde, J.A.; Martínez-Arroyo, M.; Hernández-Hernández, J.L. Recognition of the Damage Caused by the Cogollero Worm to the Corn Plant, Using Artificial Vision. In Proceedings of the International Conference on Technologies and Innovation, Guayaquil, Ecuador, 30 November–3 December 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 111–122.
23. Hinton, G.E.; Osindero, S.; Teh, Y.W. A fast learning algorithm for deep belief nets. *Neural Comput.* **2006**, *18*, 1527–1554. [CrossRef]
24. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *Pattern Anal. Mach. Intell. IEEE Trans.* **2015**, *37*, 1904–1916. [CrossRef]
25. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef]
26. Shill, A.; Rahman, M.A. Plant Disease Detection Based on YOLOv3 and YOLOv4. In Proceedings of the 2021 International Conference on Automation, Control and Mechatronics for Industry 4.0 (ACMI), Rajshahi, Bangladesh, 8–9 July 2021.
27. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In Proceedings of the Computer Vision—ECCV 2016, Amsterdam, The Netherlands, 11–14 October 2016; Springer: Cham, Switzerland, 2016.
28. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal Loss for Dense Object Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *99*, 2999–3007.
29. Law, H.; Deng, J. Cornernet: Detecting objects as paired keypoints. In Proceedings of the European Conference on Computer Vision 2018, Munich, Germany, 8–14 September 2018; pp. 734–750.
30. Zhu, C.; He, Y.; Savvides, M. Feature Selective Anchor-Free Module for Single-Shot Object Detection. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 734–750.
31. Kong, T.; Sun, F.; Liu, H.; Jiang, Y.; Shi, J. FoveaBox: Beyond Anchor-based Object Detector. *IEEE Trans. Image Process.* **2019**, *29*, 7389–7398. [CrossRef]
32. Duan, K.; Bai, S.; Xie, L.; Qi, H.; Huang, Q.; Tian, Q. CenterNet: Keypoint Triplets for Object Detection. In Proceedings of the IEEE International Conference on Computer Vision 2019, Seoul, Korea, 27 October–2 November 2019.
33. Bochkovskiy, A.; Wang, C.Y.; Liao, H. YoLov4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934.
34. Xia, X.; Sun, Q.; Shi, X. Apple detection model based on lightweight anchor-free deep convolutional neural network. *Smart Agric.* **2020**, *2*, 99–110.
35. Tian, Z.; Shen, C.; Chen, H.; He, T. FCOS: Fully Convolutional One-Stage Object Detection. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019.
36. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. *arXiv* **2017**, arXiv:1706.03762.
37. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-End Object Detection with Transformers. In Proceedings of the Computer Vision—ECCV 2020, Glasgow, UK, 23–28 August 2020; Springer: Cham, Switzerland, 2020.
38. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; 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.
39. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path Aggregation Network for Instance Segmentation. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018.
40. Neubeck, A.; Gool, L. Efficient Non-Maximum Suppression. In Proceedings of the International Conference on Pattern Recognition 2006, Hong Kong, China, 20–24 August 2006.
41. Yu, J.; Jiang, Y.; Wang, Z.; Cao, Z.; Huang, T. Unitbox: An Advanced Object Detection Network. In Proceedings of the 24th ACM International Conference on Multimedia, Amsterdam, The Netherlands, 15–19 October 2016.
42. Hou, Q.; Zhou, D.; Feng, J. Coordinate Attention for Efficient Mobile Network Design. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2021, Nashville, TN, USA, 20–25 June 2021.
43. Stergiou, A.; Poppe, R.; Kalliatakis, G. Refining activation downsampling with SoftPool. In Proceedings of the IEEE/CVF International Conference on Computer Vision 2021, Montreal, QC, Canada, 10–17 October 2021.
44. Corn Leaf Infection Dataset. Available online: <https://www.kaggle.com/datasets/qramkrishna/corn-leaf-infection-dataset> (accessed on 6 June 2021).
45. Everingham, M.; Gool, L.V.; Gool, L.V.; Williams, C.K.I.; Williams, C.K.I.; Winn, J.; Winn, J.; Zisserman, A.; Zisserman, A. Visual Object Classes (VOC) Challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–338. [CrossRef]
46. Vicente, S.; Carreira, J.; Agapito, L.; Batista, J. Reconstructing PASCAL VOC. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2014, Columbus, OH, USA, 23–28 June 2014.
47. Krizhevsky, A.; Sutskever, I.; Hinton, G. ImageNet Classification with Deep Convolutional Neural Networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1106–1114. [CrossRef]
48. Li, Z.; Zhang, S.; Zhang, J.; Huang, K.; Peng, C.L. MVP-Net: Multi-view FPN with Position-aware Attention for Deep Universal Lesion Detection. In Proceedings of the MICCAI 2019, Shenzhen, China, 13–17 October 2019.
49. Yang, J.; He, Y.; Kuang, K.; Lin, Z.; Pfister, H.; Ni, B. Asymmetric 3D Context Fusion for Universal Lesion Detection. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention 2021, Strasbourg, France, 27 September–1 October 2021.

50. Ghiasi, G.; Lin, T.Y.; Le, Q.V. Dropblock: A regularization method for convolutional networks. In Proceedings of the 32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, QC, Canada, 3–8 December 2018; p. 31.
51. Hernández-Hernández, J.; García-Mateos, G.; González-Esquiva, J.; Escarabajal-Henarejos, D.; Ruiz-Canales, A.; Molina-Martínez, J. Optimal color space selection method for plant/soil segmentation in agriculture. *Comput. Electron. Agric.* **2016**, *122*, 124–132. [[CrossRef](#)]