sustainability

MDPI

Article

Research on Key Algorithm for Sichuan Pepper Pruning Based on Improved Mask R-CNN

Chen Zhang [1,2], Yan Zhang [1,2], Sicheng Liang [1,2] and Pingzeng Liu [1,2,3,*]

[1] School of Information Science and Engineering, Shandong Agricultural University, Taian 271018, China; zc1370894848@163.com (C.Z.); zhangyandxy@sdau.edu.cn (Y.Z.); liangsichengsdau@163.com (S.L.)
[2] Key Laboratory of Huang-Huai-Hai Smart Agricultural Technology, Ministry of Agriculture and Rural Affairs, Taian 271018, China
[3] Agricultural Big-Data Research Center, Shandong Agricultural University, Taian 271018, China
* Correspondence: pzliu@sdau.edu.cn

Abstract: This Research proposes an intelligent pruning method based on the improved Mask R-CNN (Mask Region-based Convolutional Neural Network) model to address the shortcomings of intelligent pruning technology for Sichuan pepper trees. Utilizing ResNeXt-50 as the backbone network, the algorithm optimizes the anchor boxes in the RPN (Region Proposal Network) layer to adapt to the complex morphology of pepper tree branches, thereby enhancing target detection and segmentation performance. Further reducing the quantization error of the RoI (Region of Interest) Align layer through bilinear interpolation, the algorithm innovatively introduces edge loss (L_{edge}) into the loss function to address the issue of blurred edge features caused by the overlap between retained and pruned branches. Experimental results demonstrate the outstanding performance of the improved Mask R-CNN model in segmenting and identifying pepper tree branches, achieving recognition accuracies of 92.2%, 96.3%, and 85.6% for Upright branches, Centripetal branches, and Competitive branches, respectively, while elevating the recognition accuracy of retained branches to 94.4%. Compared to the original Mask R-CNN, the enhanced model exhibits a 6.7% increase in the recognition rate of retained branches and a decrease of 0.12 in loss value, significantly enhancing recognition effectiveness. The research findings not only provide an effective tool for the precise pruning of pepper trees but also offer valuable insights for implementing intelligent pruning strategies for other fruit trees.

Keywords: Sichuan pepper tree pruning; Mask R-CNN; deep learning; pruning branch identification

check for updates

Citation: Zhang, C.; Zhang, Y.; Liang, S.; Liu, P. Research on Key Algorithm for Sichuan Pepper Pruning Based on Improved Mask R-CNN. *Sustainability* **2024**, *16*, 3416. https://doi.org/10.3390/su16083416

Academic Editors: Jun (Justin) Li and Spyros Fountas

Received: 23 February 2024
Revised: 10 April 2024
Accepted: 11 April 2024
Published: 19 April 2024

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

1. Introduction

Sichuan pepper, as an essential spice, enjoys wide applications in food seasoning and traditional Chinese medicine. Achieving high-quality Sichuan pepper fruits necessitates meticulous shaping and pruning techniques. Though proper shaping and pruning, can not only help the Sichuan pepper tree acquire a robust framework with distinct layers, but it can also effectively increase the pepper tree's yield and prolong its lifespan [1]. In modern agricultural practices, fruit tree pruning has gradually shifted from traditional reliance on experience towards scientific and intelligent methodologies. Prudent pruning directly impacts fruit yield and quality, thereby enhancing the overall industry profitability. However, fruit tree pruning itself is a highly complex and nonlinear process [2], demanding practitioners to possess not only profound theoretical knowledge but also ample practical experience. Particularly for Sichuan pepper trees with unique growth habits and tree structures, existing shaping and pruning techniques still face numerous challenges and shortcomings. Therefore, continuous exploration and innovation in pruning methods are necessary to adapt to the growth characteristics of Sichuan pepper trees, thus optimizing yield and quality.

Currently, the shaping and pruning techniques for fruit trees are still imperfect [3]. Pruning is one of the most crucial activities in fruit tree production, heavily relying on manual labor. The shortage of skilled labor has led to increased labor costs, posing a significant challenge for the fruit tree industry [4]. In the field of fruit tree pruning, some scholars have begun to explore the use of deep learning techniques to assist in pruning decisions. Huang Biao was among the first to conduct relevant research on loquat branch identification technology, proposing algorithms for loquat branch image recognition and framework extraction. Through experiments, these algorithms were verified to meet various recognition requirements for branch images. However, further research is needed to address issues such as dense foliage and branch obstructions [5]. Following this, Bai et al. utilized two-dimensional laser scanning technology to extract growth parameters of fruit trees and proposed pruning methods for canopy shaping and main branch management [6]. Ge Ruiting et al. tested the pruning points of Chinese wolfberry using Mask RCNN and YOLO (You Only Look Once) V3 network models. They proposed an improved and efficient Mask RCNN algorithm, enhancing the detection speed and accuracy of pruning points for Chinese wolfberry. This laid the theoretical foundation for Chinese wolfberry pruning robots [7]. Li Xinxing et al. proposed a method based on tree structure analysis and artificial intelligence pruning decisions. They introduced a three-dimensional skeleton extraction method for branches based on local point clouds and a pruning decision method based on BP (Back Propagation) neural networks. This approach enabled the digitalization and intelligent pruning of apple trees [8]. P G and colleagues fine-tuned and tested two different deep neural networks for segmenting dormant grape branches to be pruned [9]. Xue Huifang et al. developed an optimized YOLOv3 network model and a Kinect v2 depth camera positioning model. They identified and located pruning points on young goji berry trees, laying the groundwork for spatial localization of pruning points [10]. Liang Xifeng et al. proposed an identification method for tomato lateral branch pruning points based on an improved Mask R-CNN model. This method effectively addresses the issue of tomato leaf pruning robots being unable to accurately identify tomato lateral branch pruning points [11]. Liang Kun et al. conducted research on key algorithms for grape pruning using an improved convolutional neural network. They further optimized the pruning process for grapes [12]. While the aforementioned studies have shown promising results in the localization and identification of pruning points, there hasn't been the development of an algorithmic model capable of accurately guiding pruning decisions for Sichuan pepper trees. This is largely due to the unique features of Sichuan pepper branches, including their spiky branches and slender twigs.

In the process of shaping and pruning Sichuan pepper trees, the presence of interfering branches competes for the tree's nutrients and space, ultimately affecting the pepper's yield. Based on the degree of interference and its impact, interfering branches can be classified into two categories: relative interfering branches and absolute interfering branches. The criteria for judging relative interfering branches are unclear, and their impact on the growth of Sichuan pepper trees is minimal. In contrast, absolute interfering branches possess distinct morphological features and are abundant at various stages of branch growth. These branches severely inhibit the nutrient transport and distribution of Sichuan pepper trees, making them the key targets for pruning. Therefore, this study focuses on the intelligent identification of absolute interfering branches in Sichuan pepper trees.

After comparing various instance segmentation models, Mask R-CNN was chosen as the most suitable algorithm for fine segmentation of Sichuan pepper branches. Considering the uniqueness of Sichuan pepper branches, this study improved the original model to achieve accurate segmentation and identification of Sichuan pepper branches, thereby significantly enhancing overall pruning effectiveness. Experimental results have demonstrated that this method not only accurately segments and identifies absolute interfering branches in Sichuan pepper trees but also assists in pruning decision-making. It provides new technical support and solutions for the intelligent pruning decisions of

Sichuan pepper trees, promoting the continuous development and improvement of the Sichuan pepper industry.

2. Materials and Methods

2.1. Experimental Image Acquisition and Pre-Processing

2.1.1. Data Collection

The experimental subject of this study is the Laiwu Sichuan pepper, which has a long history of cultivation. The experimental data was collected from Sichuan pepper trees in the Niuquan Town of Laiwu District, Jinan City, Shandong Province, during the winter dormancy period. The image acquisition device used was a Canon EOS 70D DSLR camera (The equipment was sourced from Canon, located in Tokyo, Japan), with photographs taken at distances ranging from 40 to 70 cm, between September and November 2023. During this period, Sichuan pepper trees shed their leaves, making branches clearly visible and ideal for identifying and segmenting different types of branches.

Winter pruning is a crucial aspect of Sichuan pepper cultivation management. However, during this time, Sichuan pepper branches closely resemble the color of the soil, and due to the small spacing between Sichuan pepper plants and the interlacing of branches, images collected from the natural environment present a complex background, posing challenges for subsequent image processing. To reduce redundant information in the images, this study used a background cloth during image acquisition to isolate the subject for identification, thereby minimizing background interference during image processing. Meanwhile, considering the different growth statuses of pepper tree branches in various directions, this study collected 4342 JPG format images of pepper trees from 300 trees using a multi-view, multi-angle approach. To ensure the quality and quantity of the dataset, this study manually selected 1350 high-quality images with clearly visible branches and no cluttered fine branches as the dataset, as shown in Figure 1. In the figure, this study used all data types except for the first one (no background cloth) to enhance the model's adaptability.

Figure 1. Types of images.

2.1.2. Data Augmentation

Deep neural networks require a large number of training samples to ensure their performance, as insufficient data can lead to overfitting of the network [13]. Therefore, in this experiment, image augmentation techniques were utilized to expand the training set samples to 6904 images by applying mirroring, noise processing, color enhancement, and blurring to the training set samples. This approach addresses the aforementioned issue and effectively reduces the model's dependence on certain attributes, thereby enhancing the model's generalization ability [14]. The effect of data augmentation is illustrated in Figure 2.

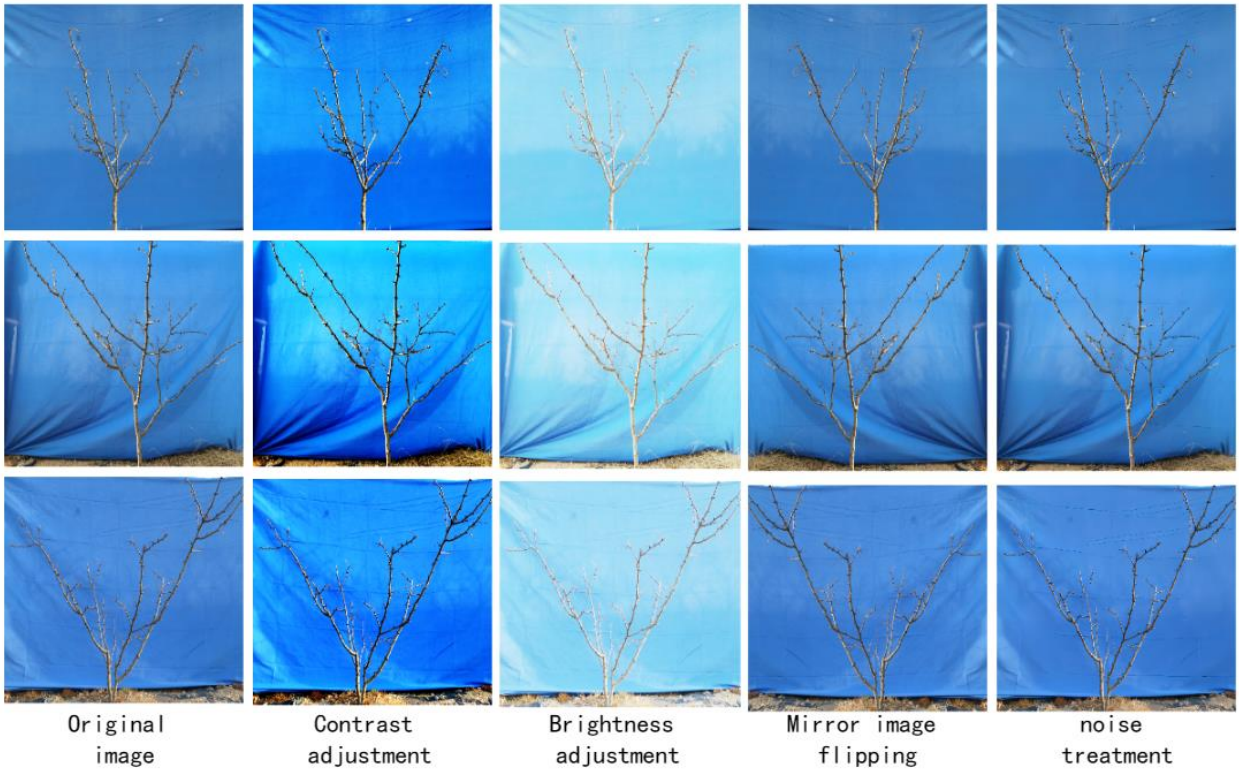

Figure 2. Partial data augmentation effect.

After data augmentation of the Sichuan pepper tree images, To increase the quantity of the dataset, the study selected 5524 images at a ratio of 8:1:1 as the training set. These images were used to train the deep neural network model. Additionally, 690 images were allocated as the testing set to evaluate the performance of the model, while another 690 images were set aside as the validation set to verify the accuracy of the model during training.

2.2. Research Plan Introduction

Based on previous research and practical experience, absolute interfering branches mainly include three types: Upright branches, Centripetal branches, and Competitive branches. Upright branches refer to branches that grow excessively vigorously, are excessively long, and grow upright. These branches consume a large amount of tree nutrients, leading to poor growth of other branches. Centripetal branches refer to branches whose growth direction is towards the center of the tree. These branches occupy the central space of the tree, affecting the ventilation and lighting conditions of the canopy. Branches other than the absolute interference branches are tentatively designated as retained branches in the segmentation task. Competitive branches are similar in morphology to retained branches, but their presence is more conducive to nutrient transport. These branches compete with retained branches for nutrients and space, resulting in poor growth of retained branches. Among them, Upright branches and Centripetal branches can be independently judged based on their respective morphological characteristics without reference to other branches, while Competitive branches, although similar in morphology to retained branches, are more conducive to nutrient transport. Therefore, when judging Competitive branches, it is necessary to compare them with the corresponding retained branches, as shown in Figure 3.

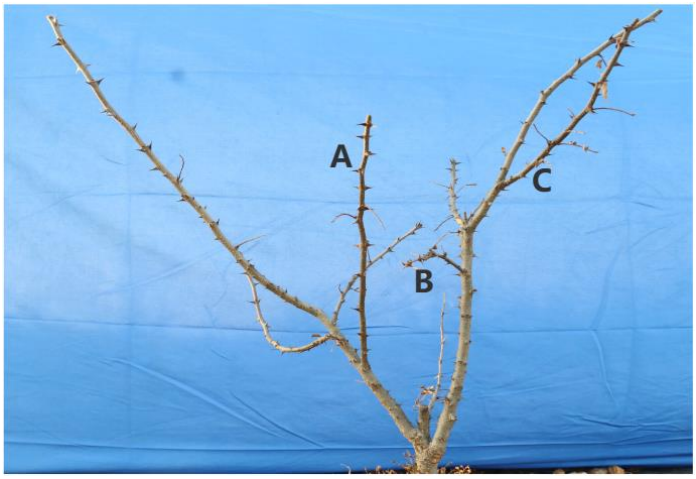

Figure 3. A for Upright branch, B for centripetal branch, C for competitive branches.

2.3. Dataset Labeling

In the task of pepper tree branch segmentation, data annotation is a crucial step that directly affects the training effectiveness of the model and performance evaluation. Based on the three types of absolute interfering branches in pepper trees (Upright branches, Centripetal branches, and Competitive branches), this study meticulously filtered the collected pepper dataset to ensure the presence of a certain number of interfering branches in each image. The purpose of this step is to select images that meet the requirements of the recognition task to ensure the quality and representativeness of the training set. This study used the Labelme annotation tool and, under the guidance and suggestions of pruning experts, annotated all interfering branches and retained branches in the dataset. As shown in Figure 4, in this figure, red represents reserved branches, while the remaining colors represent different types of interfering branches. This tool provides an intuitive interface and convenient operation, making the annotation process more efficient and accurate.

Figure 4. Labelme labeling diagram.

After annotation the dataset using the Labelme tool, corresponding JSON files are automatically generated in the folder. By using the built-in script file "label-me_json_to_dataset.exe" provided by the Labelme tool, the annotated JSON files can be converted into a folder containing five files. Figure 5 displays the contents of the converted folder, showing detailed annotation information and the segmentation effect. This not only facilitates subsequent data processing and model training but also holds significant importance for subsequent performance evaluation and analysis. With this, the preparation of the dataset for the Mask R-CNN network model is completed. The rigor and accuracy of this step lay a solid foundation for the subsequent model training and result analysis.

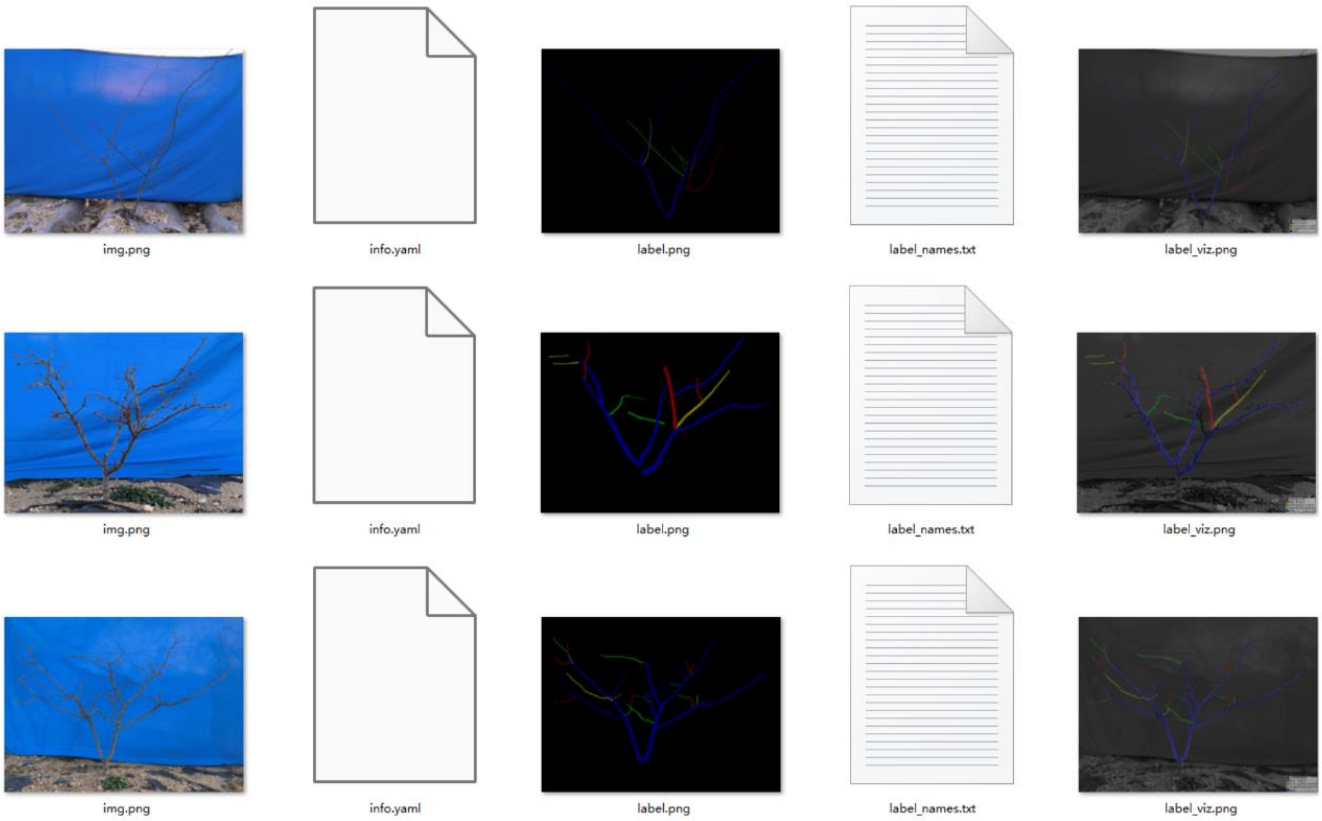

Figure 5. Partial JSON converted files.

2.4. Mask R-CNN Algorithm

The R-CNN (Region-based Convolutional Neural Network) series (R-CNN, Fast R-CNN, Faster R-CNN) is the pioneer in using deep learning for object detection [15]. Fast R-CNN and Faster R-CNN both follow the concept of R-CNN. R-CNN stands for Region with CNN (Convolutional Neural Network) Features, indicating the utilization of CNN to extract features from Region Proposals, followed by SVM (Support Vector Machine) classification and bounding box regression. The innovation of Fast R-CNN lies in proposing the RoI Pooling feature extraction method, effectively addressing the drawback of inputting Region Proposal areas separately into CNN networks in traditional R-CNN. However, its limitation lies in the consistent use of the traditional Selective Search method to determine Region Proposals, consuming a significant amount of time during both training and testing phases. In contrast, Faster R-CNN innovatively utilizes the RPN network to directly extract Region Proposals, thereby improving this limitation. It integrates RPN into the overall network, resulting in significant improvements in comprehensive performance, particularly in terms of detection speed.

The innovation of Mask R-CNN lies in efficiently detecting objects while simultaneously outputting high-quality instance segmentation masks. It is an extension of Faster

R-CNN, augmenting a branch for predicting segmentation masks in parallel with bounding box detection. By combining object detection and semantic segmentation, Mask R-CNN achieves instance segmentation. The algorithm framework of Mask R-CNN is illustrated in Figure 6.

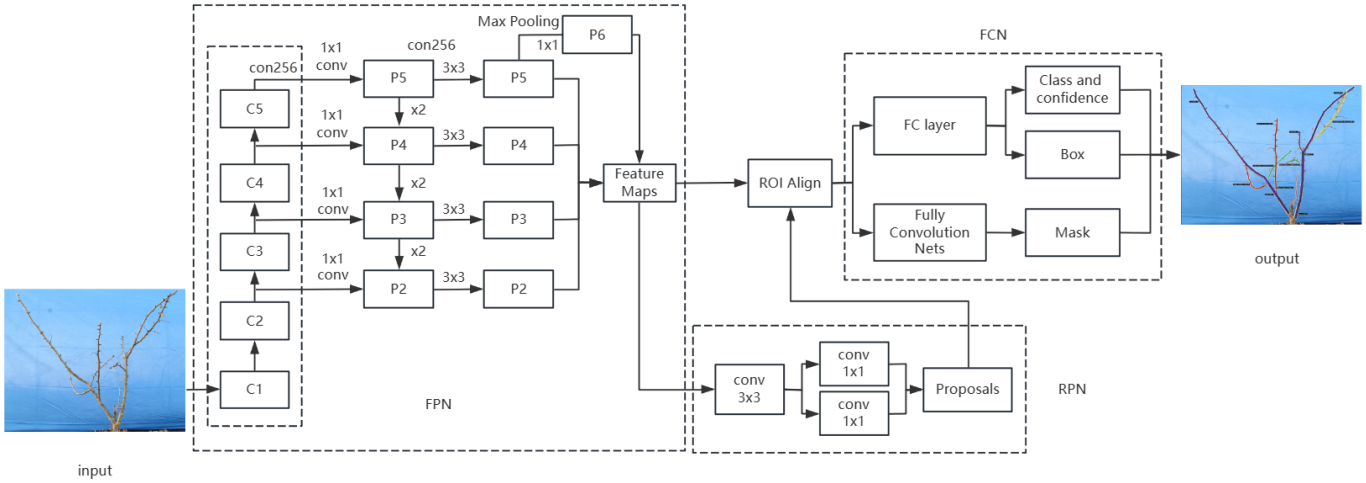

Figure 6. Mask R-CNN algorithm framework.

Since its introduction, Mask R-CNN has achieved excellent results in both object detection and instance segmentation [16]. As shown in the above figure. Its backbone network adopts ResNet (Residual Network) and utilizes the FPN (Feature Pyramid Network) structure, including both bottom-up and top-down pathways as well as lateral connections. In particular, 1×1 convolutional kernels reduce the number of feature maps without altering their spatial dimensions. The bottom-up process follows the standard forward propagation of a neural network, while the top-down process involves upsampling high-level feature maps and merging them with low-level feature maps. The two layers of lateral connections have the same spatial dimensions, allowing for the utilization of detailed localization information from the bottom layers. This iterative process continues until the final resolution map is generated. The design philosophy behind FPN is to fuse feature maps from different hierarchical levels to provide richer contextual information. Such a design is highly beneficial for deep learning applications, as it enhances model performance and robustness. The combination of ResNet and FPN serves as the feature extraction network [17]. The fusion of high-level features and low-level features generates feature maps with multidimensional characteristics, which are then shared with the RPN and the RoI Align layer. The RPN takes the feature maps as input and produces anchor boxes for Sichuan pepper tree branches along with their corresponding scores. NMS (Non-Maximum Suppression) is utilized to remove anchor boxes with lower scores [18].

2.5. Mask R-CNN Model Optimization for Sichuan Pepper Branch Segmentation Application

2.5.1. ResNet Network Improvements

In the field of deep learning, optimizing the structure of neural networks has always been a research hotspot. For the task of Sichuan pepper tree branch segmentation, considering the complexity of the planting environment and the mutual interference between branches, it is necessary to improve the existing Mask R-CNN model to enhance its accuracy in identifying and segmenting Sichuan pepper tree branches. The original Mask R-CNN model uses ResNet as the backbone network for feature extraction. ResNet was proposed by He et al. in 2015, and it effectively addresses the problem of degradation in training deep neural networks by introducing residual learning. The core idea of residual learning is to shift the network's learning target from fitting the original layer functions to fitting residual functions, where the network learns the difference between input and

output. This simplifies the learning process, enhances training stability, and improves the network's performance. The structure is illustrated in Figure 7. The input, with 256 channels of features, undergoes compression by a 1 × 1 convolution to reduce to 64 channels. Subsequently, a 3 × 3 convolutional kernel processes the features, and after expanding the channel number with a 1 × 1 convolution, the residual is connected to the original features to produce the output.

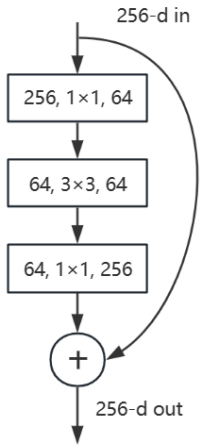

Figure 7. Schematic diagram of ResNet module.

However, in the task of Sichuan pepper branch segmentation, the small spacing between Sichuan pepper plants and the intertwining of branches increase the difficulty of recognition and segmentation. To enhance the model's performance, this study considers replacing the original ResNet structure with the ResNeXt-50 structure as the backbone network. ResNeXt-50 combines the advantages of the Inception structure and ResNet structure by using grouped convolution to increase the width of the network while maintaining its depth. This structure is not only easy to train but also allows for the extraction of features from multiple perspectives. Compared to ResNet, ResNeXt-50 can improve accuracy without increasing parameter complexity and reduce the number of hyperparameters, thereby reducing the model's complexity. Additionally, the ResNeXt-50 structure adopts the idea of shortcut connections, introducing cross-layer connections in the network to facilitate information propagation throughout the network. This design alleviates the problem of gradient vanishing and enhances the model's ability to perceive small Sichuan pepper branches and deep features [19]. Its structure is illustrated in Figure 8. The input features with 256 channels are divided into 32 groups, each compressed to 4 channels after a 64-fold reduction. After the 32 groups are added together and concatenated with the original residual features, the output is obtained. The use of "groups" instead of many smaller pathways is because these groups contain multiple parallel pathways, each used to learn different features. This enables the network to more effectively learn a greater variety of features, increasing its representational capacity. This grouped convolution approach effectively reduces computational complexity. In summary, the ResNeXt-50 network is essentially a ResNet structure with aggregated residual and local connection structures, while also incorporating data augmentation and regularization techniques such as Random Erasing and Mixup.

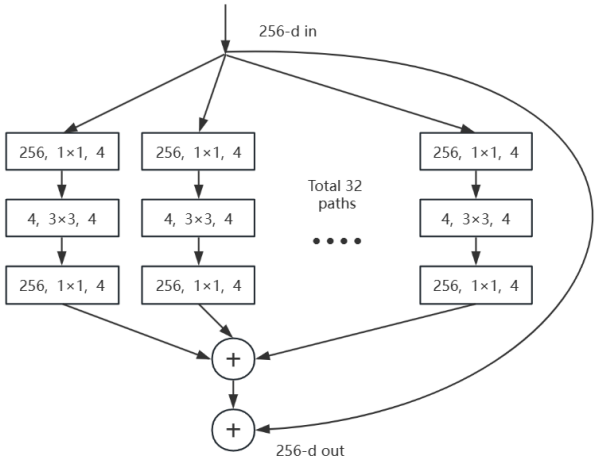

Figure 8. Schematic diagram of ResNeXt50 structure.

2.5.2. RPN Network Improvements

The RPN is a crucial component in target detection algorithms like Mask R-CNN. It's responsible for generating a series of RoI candidates, which are likely to contain target objects. The operation of the RPN is primarily based on anchor boxes, predefined rectangular boxes used to capture targets of different scales and aspect ratios in the image. In the original design of the RPN, the sizes and aspect ratios of the anchor boxes are typically determined based on cluster analysis results from specific datasets (such as human bodies, vehicles, etc.). However, in the application scenario of pepper tree branch segmentation, the shape and size distribution of the targets may differ significantly from these datasets. In consideration of the shape characteristics of pepper tree branches, this chapter conducts a feature similarity analysis and optimizes the anchor boxes in the RPN layer based on the analysis results. Specifically, this study adjusts the aspect ratios and sizes of the anchor boxes to better match the morphology of pepper tree branches. In the original design of the RPN, three sizes {128, 256, 512} were obtained through cluster analysis of the dataset, and they were allocated in proportions of 1:1, 1:2, and 2:1, resulting in a total of $3 \times 3 = 9$ candidate boxes [20]. However, these settings are not entirely suitable for the morphology of pepper tree branches in the application scenario of this paper. Therefore, based on the feature similarity of pepper tree branches, cluster analysis was conducted, and new anchor box configurations were selected. In the new configuration, aspect ratios of (1:1, 1:4, 4:1) and sizes of (128×128, 128×512, 512×128) were chosen. This configuration is closer to the actual morphology of pepper tree branches and is expected to improve the detection and segmentation effectiveness of targets while reducing computational complexity. The structure is illustrated in Figure 9.

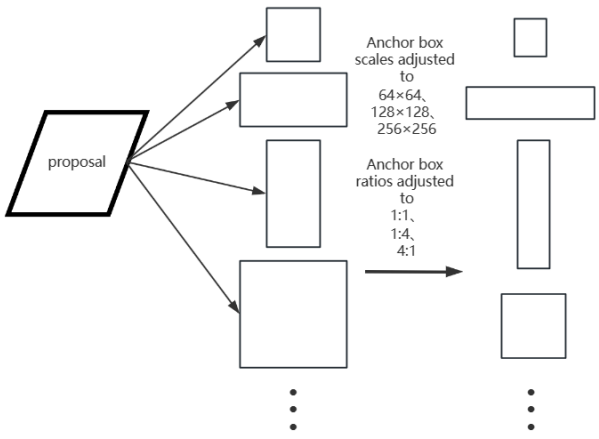

Figure 9. The optimized RPN network structure.

2.5.3. RoI Align Layer Optimization

RoI Align in Mask R-CNN is a method of region feature aggregation that effectively addresses the quantization errors caused by the two quantization operations in RoI Pooling of Faster R-CNN, thereby improving the accuracy of the detection model [21]. As shown in Equation (1), the original RoI Pooling method divides the RoI region into a fixed number of bins and performs a max pooling operation on each bin. However, this process involves two quantization operations: the first is quantizing the boundaries of RoI to integers for alignment with the pixels of the feature map, and the second is quantization during the selection of the maximum value within each bin. These quantization operations introduce errors, especially in high-resolution or precise localization scenarios. The RoI Align layer reduces these errors by avoiding these quantization operations. It uses bilinear interpolation to accurately compute the average of features within each bin, as shown in Figure 10, rather than simply selecting the maximum value. This approach provides more accurate feature aggregation, thereby improving the accuracy of object detection.

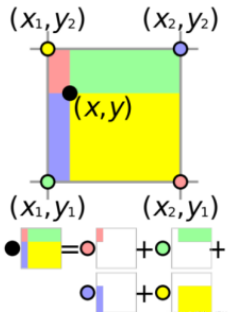

Figure 10. RoI Align bilinear interpolation.

In the task of pepper tree branch segmentation, precise localization is particularly important due to the intersections between retained branches and pruned branches. Therefore, this experiment adopts the double integral method to further reduce the quantization errors that may exist in the pooling layer. As shown in Equation (2), the function $f(x, y)$ is double-integrated within the specific RoI region and normalized, ensuring that the result of the pooling operation is the average based on the area of the region. This optimization is expected to improve the performance of Mask R-CNN in pepper tree branch segmentation tasks, especially in handling complex scenarios such as intersecting branches.

$$\partial \text{Pool}(bin) = \sum_{i=1}^{N} f(a_i - b_i)/N \tag{1}$$

$$\partial Pool(bin, F) = \frac{\int_{y_1}^{y_2} \int_{x_1}^{x_2} f(x, y) dx dy}{(x_2 - x_1)(y_2 - y_1)} \tag{2}$$

In Equation (2), *bin* represents the feature block to be pooled; N is the number of blocks into which these feature blocks are divided; F is the total number of feature blocks; a_i and b_i are the horizontal and vertical coordinates of the blocks obtained by linear interpolation after dividing the RoI region into blocks. x_1, x_2, and y_1, y_2 respectively denote the coordinates of the top-left and bottom-right corners of the RoI region.

2.5.4. Loss Function Optimization

In the Mask R-CNN model, the design of the loss function is crucial for the training effectiveness and final performance of the model. The loss function of this model typically consists of a weighted sum of multiple components, including classification loss (L_{cls}), bounding box loss (L_{box}), and average binary cross-entropy loss (L_{mask}). These loss terms correspond to different tasks of the model: classification, bounding box regression, and instance segmentation.

However, in the specific scenario of pepper tree branch segmentation, the overlap between retained branches and pruned branches can lead to blurry edge features, increasing the difficulty of segmentation. This blurred partial edge feature can result in slower training times and the final segmentation mask being insensitive to edges. To enhance the model's sensitivity to such edges, this paper proposes adding an edge loss L_{edge} to the loss function, which helps reduce the blurriness of edge features caused by overlapping intersections, thus accurately locating the boundaries of pepper tree branches. Specifically, in the implementation, this study first performs convolution operations on the annotated branch masks, using the second-order differential Laplace operator to obtain the actual edge information. Then, these edge information is utilized as additional training data. Subsequently, in the loss function, the Laplace operator is applied to extract the predicted edges from the predicted masks, which are then compared with the actual edges to calculate the edge loss. To prevent the influence of other mask edges on the detected mask edge loss, the average binary cross-entropy is chosen as the edge loss function. This loss function effectively measures the difference between the predicted edges and the actual edges and provides meaningful gradient information to guide the model's training. The edge loss L_{edge} function is represented as shown in Equation (3).

$$L_{edge} = -\sum_{i=1}^{n}[q_i^* log(q_i) + (1 - q_i^*)\log(1 - q_i)] \quad (3)$$

In Equation (3), q_i represents the predicted probability of a pixel, while q_i^* is a binary indicator used to identify whether the pixel is an edge pixel. If the pixel is an edge pixel, q_i^* is set to 1; otherwise, it is set to 0. This paper measures the performance of the model by calculating the difference between the predicted probability q_i and the ground truth label q_i^*. If the model's predictions for edge pixels are very accurate, this loss will be small; conversely, if the model's predictions deviate significantly from the ground truth, then the loss function will be large.

Therefore, the total loss function of the Mask R-CNN network with the addition of the edge loss L_{edge} is represented as Equation (4).

$$L = \alpha L_{cls} + \beta L_{box} + \lambda L_{mask} + L_{edge} \quad (4)$$

where α, β and λ are the weight parameters of L_{cls}, L_{box} and L_{mask}. L_{cls} denotes the loss of categorization; L_{box} denotes the border regression loss; L_{mask} denotes the mask loss.

3. Experiment and Result Analysis

3.1. Model Training

This study utilized the TensorFlow framework to train the Mask R-CNN model and accelerated the training process using a GPU. The computer system used for training comprised an Intel(R) Core (TM) i7-11320H@3.20GHz processor, and an NVIDIA GeForce RTX 1650 GPU with 16GB of memory. The model was developed and written in the Python 3.8.3 environment using the PyCharm 2023 integrated development tool.

In the training of deep learning models, the selection of hyperparameters has a crucial impact on the performance of the model. Table 1 provides detailed key hyperparameter settings used in training the Mask R-CNN model in this study. These hyperparameters include, but are not limited to, learning rate, batch size, number of iterations, and optimizer type, etc. They collectively determine the convergence speed, stability, and final performance of the model during training. Through careful adjustment of these hyperparameters, it is possible to obtain more accurate and robust models [22], better suited to the requirements of pepper branch segmentation tasks.

Table 1. Model hyper parameter settings.

Parameters	Value
WEIGHT_DECAY	0.0001
BATCH_SIZE	2
POOL_SIZE	7
STEPS_PER_EPOCH	100
POST_NMS_RoIS_INFERENCE	1000
POST_NMS_RoIS_TRAINING	2000
MASK_POOL_SIZE	14
LEARNING_MOMENTUM	0.9
LEARNING_RATE	0.001
MAX_GT_INSTANCES	100
DETECTION_MAX_INSTANCES	100
DETECTION_MIN_CONFIDENCE	0.9
DETECTION_NMS_THRESHOLD	0.3

To assess the performance of the models before and after improvement, this study conducted experiments on a specifically constructed pepper dataset. Fine adjustments and reconfigurations were made to the parameters of both the original Mask R-CNN model and the improved version. Specifically, the regularization coefficient (WEIGHT_DECAY) was initially set to 0.0001. Regularization is a technique used to prevent model overfitting by adding a penalty term related to weights to the model's loss function. A smaller regularization coefficient implies a smaller penalty on the weights, which helps retain important features of the model while reducing unnecessary complexity to prevent overfitting. Secondly, the BATCH_SIZE was set to 2, meaning the network processes two training samples before updating weights. Although this increases training time, it allows the model to adjust weights more finely as gradient information from each sample is fully utilized [23]. Moreover, smaller batch sizes can reduce memory usage, making the training process more feasible for memory-limited scenarios.

In the settings of the RoI Align layer, adjustments were made to both its pooling size and mask pooling size. These parameters determine the spatial resolution when extracting region features from feature maps. Appropriate pooling sizes can capture sufficient spatial information while reducing computational and memory usage. During training, these parameters were fine-tuned to find the optimal balance and improve segmentation accuracy. To accelerate the training process and enhance model convergence performance, the learning rate and momentum were adjusted to 0.001 and 0.9, respectively. The learning rate determines the magnitude of weight updates in each iteration, while momentum helps accelerate the gradient descent process, particularly in relevant directions. Such adjustments can lead to a more stable convergence of the model during training.

3.2. Evaluation Indicators

This experiment utilized multiple standard evaluation metrics to comprehensively assess the performance of the algorithm in pepper branch recognition and segmentation tasks, including F1 score, accuracy, recall, and precision [24,25]. Precision refers to the ratio of correctly predicted positive samples to all predicted positive samples; recall refers to the ratio of correctly predicted positive samples to all actual positive samples; accuracy refers to the ratio of correctly predicted samples to all predicted samples; and F1 score is the harmonic mean of precision and recall, balancing the model's ability to classify positive and negative samples, as shown in Equation (5). These evaluation metrics are not only used to evaluate the current model's performance but also to guide subsequent model optimization. In the equation, P represents Precision, and R represents Recall. TP (True Positive) represents true positives, indicating the number of positive samples correctly predicted as positive; TN (True Negative) represents true negatives, indicating the number of negative samples correctly predicted as negative; FP (False Positive) represents false positives, indicating the number of negative samples incorrectly predicted as positive;

and FN (False Negative) represents false negatives, indicating the number of positive samples incorrectly predicted as negative [26,27]. The calculation of each evaluation metric is as follows:

$$\mathrm{F1} = \frac{2PR}{P+R} \tag{5}$$

$$Recall = \frac{TP}{TP+FN} \tag{6}$$

$$Precision = \frac{TP}{TP+FP} \tag{7}$$

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \tag{8}$$

3.3. Test Results and Analysis

After completing the training of the Mask R-CNN models before and after improvement, this study conducted a detailed comparison and analysis of the training results [28], as shown in Table 2 and Figure 11. By comparing the changes in loss values and accuracies of the two models during the same iteration process, the effectiveness of the model improvement was thoroughly explored.

Table 2. Comparison of the performance of different models.

Method	Backbone	Feature Extraction Method	Loss Function	Accuracy	Loss
YOLOv8	Darknet53	——	$L_{cls} + L_{box} + L_{cof}$	0.796	0.20
Faster R-CNN	ResNet50	RoI Pooling	$L_{cls} + L_{box}$	0.751	0.24
Mask R-CNN	ResNet101	RoI Align	$L_{cls} + L_{box} + L_{mask}$	0.878	0.12
improved Mask R-CNN	ResNeXt50	RoI Align + double integral	$L_{cls} + L_{box} + L_{mask} + L_{edge}$	0.945	0.05

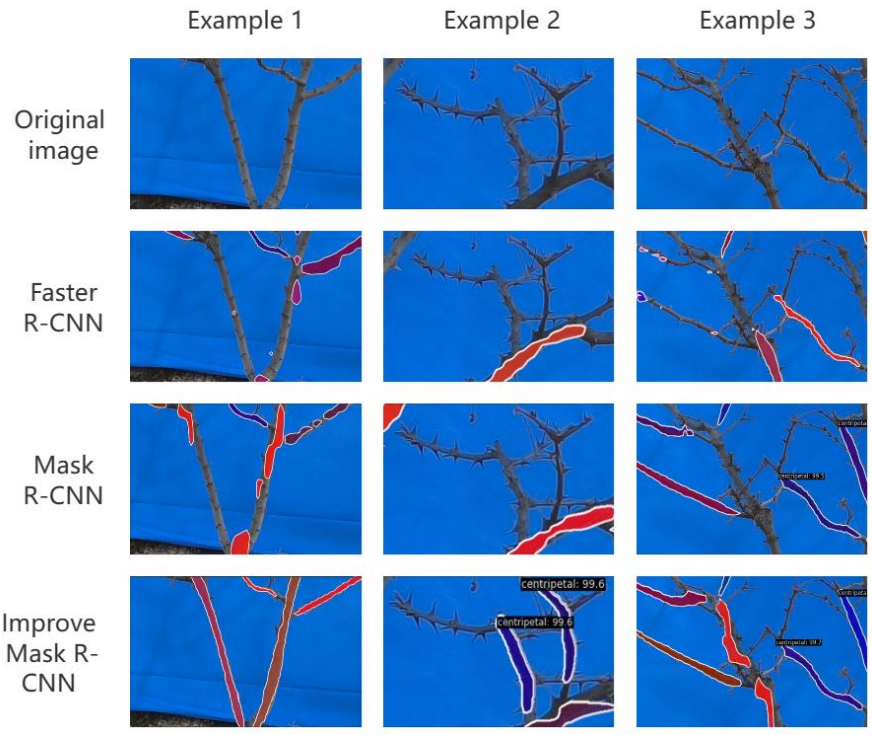

Figure 11. Different models to detect segmentation.

Specifically, by improving the Mask R-CNN model in aspects such as the backbone network, feature extraction method, and loss function, this study successfully increased the model's accuracy and reduced the loss values. To provide a more intuitive display of the model performance, the TensorBoard visualization tool was utilized to track the experimental results, as shown in Figure 12. In the figure, loss represents the loss of the training set; rpn_bbox_loss and rpn_class_loss are the regression loss and classification loss of the rpn network, respectively; mrcnn_bbox_loss and mrcnn_class_loss represent the regression loss and classification loss of the mrcnn network, respectively.

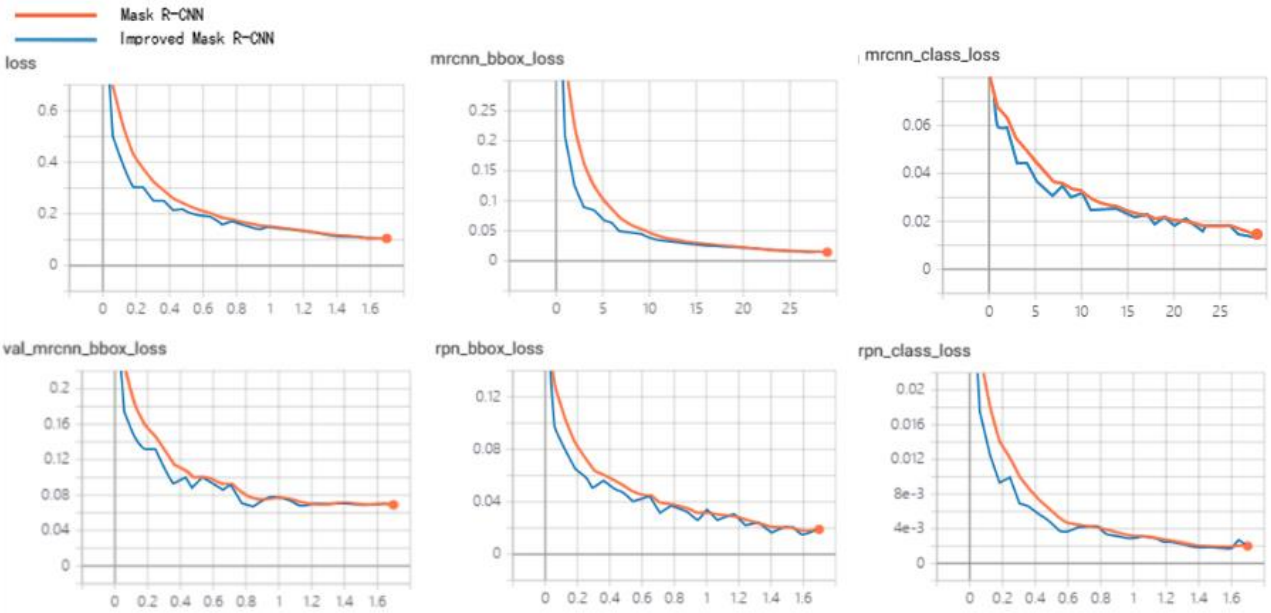

Figure 12. Comparison of original Mask R-CNN and improved Mask R-CNN results.

The results indicate that the network before improvement reached a loss value of 0.12 and an accuracy of 0.878 after 100 iterations. This suggests that the original model had relatively high performance but still had room for improvement in accuracy. On the other hand, the improved network showed a more significant performance improvement, with the loss value decreasing to 0.05 and the accuracy increasing to 0.945 after the same number of iterations. This outcome reflects the effectiveness of optimizing the model's structure and training process in this study. It's noticeable throughout the iterations that the improved model exhibits a faster decrease in loss value, especially in the early stages of iteration. This may be attributed to the optimized network structure and training strategies, allowing the model to learn and adapt to the features of pepper tree branches more quickly.

3.4. Evaluation and Validation of the Capsicum Branch Segmentation Model

3.4.1. Pepper Tree Branch Segmentation Model Evaluation

To further validate the effectiveness of the improved Mask R-CNN model in segmenting absolute interfering branches in pepper plants, this experiment adopted accuracy, precision, recall, and F1 score for comprehensive evaluation of the model [29]. Table 3 provides detailed results of these evaluation metrics. According to the table data, the improved Mask R-CNN model demonstrates excellent performance in identifying upright branches and centripetal branches. The accuracy and F1 score are more important in the table, with values above 0.94 and 0.8 respectively, demonstrating good discriminative ability. However, for competitive branches classification, the accuracy is slightly lower, at 0.856. Overall, the improved Mask R-CNN model exhibits remarkable performance in segmenting and identifying absolute interfering branches in pepper plants, achieving an overall accuracy of approximately 92% for the four types of branches. Moreover, with an average processing

time of only 3.57 s, the detection accuracy and processing speed can meet the requirements for the segmentation of absolute interfering branches in practical applications.

Table 3. Evaluation table of pepper dendritic segmentation models.

Type of Branches	Accuracy	Precision	Recall	F1-Score
Retained branches	0.921	0.808	0.875	0.840
Upright branches	0.944	0.824	0.865	0.844
Centripetal branches	0.963	0.816	0.882	0.848
Competitive branches	0.856	0.865	0.753	0.805

3.4.2. Pepper Branch Segmentation Model Verification

After accurately segmenting the pepper branches, this study randomly selected 20 images for validation. The model was used to segment absolute interference branches in these images, and the number of each type of absolute interference branches in the images was calculated. The validation results were obtained by comparing the model segmentation results with those annotated by experts, as shown in Table 4.

Table 4. Improved Mask R-CNN segmentation model expert comparison results.

Verification Image	Upright Branch		Centripetal Branch		Competitive Branch		Fault Detection Number	Detection Rate (%)	Error Rate (%)
	Actual Number	Detection Number	Actual Number	Detection Number	Actual Number	Detection Number			
1	2	2	1	2	1	1	1	100	20
2	1	2	2	2	2	1	2	100	40
3	2	2	2	1	0	0	1	75	25
4	1	1	1	1	0	0	0	100	0
5	3	3	3	2	2	1	0	75	0
6	2	2	2	1	1	1	2	80	50
7	5	4	3	3	6	7	2	100	14
8	3	4	2	2	2	3	2	100	22
9	0	0	1	1	0	1	1	100	50
10	1	1	1	1	2	1	1	75	25
11	2	2	1	1	3	2	2	83	40
12	1	1	2	2	1	1	0	100	0
13	1	1	2	2	0	1	1	100	25
14	0	0	1	1	0	1	1	100	50
15	2	1	2	2	1	1	1	75	25
16	2	2	3	3	1	1	0	100	0
17	3	3	0	0	1	0	1	75	33
18	1	1	1	1	0	0	0	100	0
19	2	2	0	1	0	0	1	100	33
20	0	0	0	0	1	1	0	100	0
Total	25	24	21	21	19	21	15	91.9	22.6

This study experimentally validated the effectiveness and application potential of the improved Mask R-CNN model in pepper tree branch segmentation tasks. Even in complex pepper images, the model demonstrated high recognition accuracy, successfully identifying absolute interfering branches in pepper images, as shown in Figure 13. Overall, the improved Mask R-CNN model exhibited significant advantages in pepper tree branch recognition tasks. In practical applications, users can upload images of pepper tree branches, and then use this model to discriminate the types of branches. The model accurately identifies three types of absolute interfering branches—Upright branches, Centripetal branches, and Competitive branches—and determines whether pruning is necessary based on the recognition results.

Figure 13. Improved Mask R-CNN algorithm detection result graph.

4. Conclusions

For the segmentation and detection of pepper tree branches, this study replaced the traditional ResNet architecture with the ResNeXt50 structure, significantly enhancing the model's ability to recognize fine branches and deep features of pepper trees. Additionally, through clustering analysis of candidate boxes in the Region Proposal Network (RPN) layer, the anchor box ratios and scales were optimized, effectively reducing computational complexity and improving target detection rates. To address cases of branch adhesion in images, the RoI Align layer was optimized using double integral methods to further improve edge segmentation accuracy. Finally, by introducing edge loss in the loss function and using mean binary cross-entropy as the edge loss function, the influence of other mask edges on model performance was successfully reduced. Ultimately, a pepper tree branch classification model based on improved Mask R-CNN was constructed and validated, enabling the accurate segmentation of Upright branches, Centripetal branches, and Competitive branches of pepper trees. Leveraging deep learning technology, this paper provides a scientific basis for identifying absolute interfering branches in the growth process of pepper trees, thereby promoting the automation of pruning operations.

Although the detection accuracy and processing speed of this model fully meet the real-time detection requirements for pepper tree branches in practical applications, limitations were observed during the instance segmentation [30] of pepper tree interfering branches. This study noticed that the model still has certain limitations in handling some complex phenomena on pepper trees, such as thin branches, excessively short branches, or excessive pepper thorns on branches, where the segmentation accuracy was not fully accurate. This exposes certain deficiencies in the model to some extent. To address these issues, future research will consider increasing the specificity of the dataset to focus on the segmentation of these complex branches, aiming to improve the model's recognition accuracy and adaptability. Overall, while this study has made significant progress in pepper tree branch recognition, there is still room for further exploration and optimization in terms of adaptability and practical application potential.

Author Contributions: Conceptualization, C.Z. and P.L.; Methodology, C.Z. and P.L.; validation, C.Z. and P.L.; investigation, S.L., C.Z. and P.L.; writing—original draft preparation, Y.Z.; writing—review and editing, Y.Z. and P.L.; visualization, C.Z. and P.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Major Agricultural Applied Technology Innovation Project of Shandong Province, grant number SD2019ZZ019; the Key Research Development Program (Major Science and Technology Innovation Projects) of Shandong Province, grant number 2022CXGC010609; and the Major Science and Technology Innovation Project of Shandong Province, grant number 2019JZZY010713.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data can be made available upon request from the authors.

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

Abbreviations

The following abbreviations are used in this manuscript:

CNN	Convolutional neural network
R-CNN	Region-based convolutional neural networks
RPN	Region Proposal Network
FPN	Feature pyramid networks
RoI	Region of interest
ResNet	Residual Network
SVM	Support Vector Machine
Bbox	Bounding box
NMS	Non-Maximum suppression

References

1. Zhang, B. The Study on the Impact of Pruning on Pepper Trees. *J. Hebei For. Sci. Technol.* **2012**, *3*, 11–24. [CrossRef]
2. Sun, J.; Xing, K.; Yang, Z.; Duan, J. Simulation and experimental research on fruit branch pruning process based on ANSYS/LS-DYNA. *J. South China Agric. Univ.* **2022**, *43*, 113–124.
3. Zhang, J. Cultivation and management techniques for high quality and high yield of Sichuan Pepper. *Seed Sci. Technol.* **2021**, *39*, 48–49. [CrossRef]
4. Long, H.; James, S. Sensing and Automation in Pruning of Apple Trees: A Review. *Agronomy* **2018**, *8*, 211. [CrossRef]
5. Huang, B. *Research on Key Technologies of Loquat Pruning Robot*; South China University of Technology: Guangzhou, China, 2016.
6. Bai, J.; Xing, H.; Ma, S.; Wang, M. Studies on Parameter Extraction and Pruning of Tall-spindle Apple Trees Based on 2D Laser Scanner. *IFAC Pap.* **2019**, *52*, 349–354. [CrossRef]
7. Ge, R. Research on Key Algorithm of Young Lycium Berry Pruning Based on Improved Mask RCNN Network Model. Master's Thesis, Northern University for Nationalities, Yinchuan, China, December 2020. [CrossRef]
8. Li, X.; Liang, B.; Liu, S.; Li, H. Construction of apple tree pruning decision-making system using local point cloud and BP neural network. *J. Agric. Eng.* **2021**, *37*, 170–176.
9. Guadagna, P.; Fernandes, M.; Chen, F.; Santamaria, A.; Teng, T.; Frioni, T.; Caldwell, D.G.; Poni, S.; Semini, C.; Gatti, M.; et al. Using deep learning for pruning region detection and plant organ segmentation in dormant spur-pruned grapevines. *Precis. Agric.* **2023**, *24*, 1547–1569. [CrossRef] [PubMed]
10. Xue, H. Research on Identification and Localization of *Lycium barbarum* Tree Pruning Point Based on Kinect. Master's Thesis, Northern University for Nationalities, Yinchuan, China, December 2021. [CrossRef]
11. Liang, X.; Zhang, X.; Wang, Y. Recognition Method of Tomato Pruning Point Based on Improved Mask R-CNN. *J. Agric. Eng.* **2022**, *38*, 112–121.
12. Liang, K.; Hu, Y. Research on key algorithm of grape pruning based on improved convolutional network. *Autom. Instrum.* **2023**, *6*, 58–62. [CrossRef]
13. Blok, P.M.; Kootstra, G.; Elghor, H.E.; Diallo, B.; van Evert, F.K.; van Henten, E.J. Active learning with MaskAL reduces annotation effort for training Mask R-CNN on a broccoli dataset with visually similar classes. *Comput. Electron. Agric.* **2022**, *197*, 106917. [CrossRef]
14. Ma, Z.; Zhang, X.; Yang, G. Research on rice stalk impurity segmentation method based on improved Mask R-CNN. *Chin. J. Agric. Mech. Chem.* **2021**, *42*, 145–150. [CrossRef]

15. Samphors, P.; Hanbo, Z.; Yonghui, S. Research on Detection Method of Insulator Image Based on Improved Faster R-CNN. *J. Phys. Conf. Ser.* **2022**, *2213*, 012036.
16. Niu, H.; Bao, T.; Li, Y.; Huang, S. Pixel-level crack detection method for concrete dams based on improved Mask R-CNN. *Adv. Sci. Technol. Water Resour.* **2023**, *43*, 87–92.
17. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 29th Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26–30 June 2016; pp. 770–778. [CrossRef]
18. Wang, Y.; Wu, Y.; Qi, B. Research on unmanned aerial spraying method for intercropping class farmland based on Faster RCNN. *Chin. J. Agric. Mech. Chem.* **2019**, *40*, 76–81. [CrossRef]
19. Tan, H.; Li, Y.; Zhu, M.; Deng, Y.; Tong, M. Overlapping fish tail detection by image enhancement with improved Faster-RCNN network. *J. Agric. Eng.* **2022**, *38*, 167–176.
20. Xiong, F.; Dong, B.; Zhang, X.; Liu, H.; Han, X.; Kuang, L. Remote sensing building detection based on improved Mask RCNN algorithm. *Comput. Eng. Des.* **2023**, *44*, 218–223. [CrossRef]
21. Ma, Y.; Fu, H.; Wu, P.; Chen, X.; Wang, D.; Chen, S.; Cao, C. Deep network adaptive optimization of Mask R-CNN model in casting surface defect detection. *Mod. Manuf. Eng.* **2022**, *4*, 112–118. [CrossRef]
22. Li, X.; Zhang, J. A Segmentation Method for Tool Detection in Security Image Based on Semantic Contour Information. Chinese Patent CN111127499A, 8 May 2020.
23. Wu, J.; Chen, S.; Liu, X. Efficient hyperparameter optimization through model-based reinforcement learning. *Neurocomputing* **2020**, *409*, 381–393. [CrossRef]
24. LI, L.; Chen, G.; Ding, C. An improved Mask R-CNN method for defect detection of cosmetic cotton pads. *J. Donghua Univ. (Nat. Sci. Ed.)* **2023**, *49*, 78–87. [CrossRef]
25. Padilla, R.; Passos, W.L.; Dias, T.L.; Netto, S.L.; Da Silva, E.A. A Comparative Analysis of Object Detection Metrics with a Companion Open-Source Toolkit. *Electronics* **2021**, *10*, 279. [CrossRef]
26. Li, J.; Jiang, J.; Dou, Y.; Xu, J.; Wen, D. Soft-NMS-based candidate frame de-redundant gas pedal design. *Comput. Eng. Sci.* **2021**, *43*, 586–593.
27. Gao, X.; Zan, X.; Yang, S.; Zhang, R.; Chen, S.; Zhang, X.; Liu, Z.; Ma, Y.; Zhao, Y.; Li, S. Maize seedling information extraction from UAV images based on semi-automatic sample generation and Mask R-CNN model. *Eur. J. Agron.* **2023**, *147*, 126845. [CrossRef]
28. Punnarai, S.; Warisa, Y.; Thidarat, B. Recognizing the sweet and sour taste of pineapple fruits using residual networks and green-relative color transformation attached with Mask R-CNN. *Postharvest Biol. Technol.* **2023**, *196*, 112174. [CrossRef]
29. Tong, S.; Zhang, J.; Li, W.; Wang, Y.; Kang, F. An image-based system for locating pruning points in apple trees using instance segmentation and RGB-D images. *Biosyst. Eng.* **2023**, *236*, 277–286. [CrossRef]
30. Cao, Y.; Zhao, Z.; Huang, Y.; Lin, X.; Luo, S.; Xiang, B.; Yang, H. Case instance segmentation of small farmland based on Mask R-CNN of feature pyramid network with double attention mechanism in high resolution satellite images. *Comput. Electron. Agric.* **2023**, *212*, 108073. [CrossRef]

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