forests

MDPI

Article

Tree Recognition and Crown Width Extraction Based on Novel Faster-RCNN in a Dense Loblolly Pine Environment

Chongyuan Cai [1,2,3], Hao Xu [4], Sheng Chen [5], Laibang Yang [6], Yuhui Weng [7], Siqi Huang [8], Chen Dong [1,2,3,*] and Xiongwei Lou [1,2,3,*]

[1] College of Mathematics and Computer Science, Zhejiang A&F University, Hangzhou 311300, China
[2] Key Laboratory of State Forestry and Grassland Administration on Forestry Sensing Technology and Intelligent Equipment, Zhejiang A&F University, Hangzhou 311300, China
[3] Key Laboratory of Forestry Intelligent Monitoring and Information Technology Research of Zhejiang Province, Zhejiang A&F University, Hangzhou 311300, China
[4] Zhejiang Forestry Bureau, Hangzhou 310000, China
[5] Center for Forest Resource Monitoring of Zhejiang Province, Hangzhou 310000, China
[6] Hangzhou Ganzhi Technology Co., Ltd., Lin'an 311300, China
[7] College of Forestry and Agriculture, Stephen F. Austin State University, Nacogdoches, TX 75962, USA
[8] Longquan Urban Forestry Workstation, Longquan 323700, China
[*] Correspondence: dongchen@zafu.edu.cn (C.D.); lxw@zafu.edu.cn (X.L.);
Tel.: +86-158-6718-0919 (C.D.); +86-135-8822-8755 (X.L.)

Abstract: Tree crown width relates directly to wood quality and tree growth. The traditional method used to measure crown width is labor-intensive and time-consuming. Pairing imagery taken by an unmanned aerial vehicle (UAV) with a deep learning algorithm such as a faster region-based convolutional neural network (Faster-RCNN) has the potential to be an alternative to the traditional method. In this study, Faster-RCNN outperformed single-shot multibox detector (SSD) for crown detection in a young loblolly pine stand but performed poorly in a dense, mature loblolly pine stand. This paper proposes a novel Faster-RCNN algorithm for tree crown identification and crown width extraction in a forest stand environment with high-density loblolly pine forests. The new algorithm uses Residual Network 101 (ResNet101) and a feature pyramid network (FPN) to build an FPN_ResNet101 structure, improving the capability to model shallow location feature extraction. The algorithm was applied to images from a mature loblolly pine plot in eastern Texas, USA. The results show that the accuracy of crown recognition and crown width measurement using the FPN_ResNet101 structure as the backbone network in Faster-RCNN (FPN_Faster-RCNN_ResNet101) was high, being 95.26% and 0.95, respectively, which was 4.90% and 0.27 higher than when using Faster-RCNN with ResNet101 as the backbone network (Faster-RCNN_ResNet101). The results fully confirm the effectiveness of the proposed algorithm.

Keywords: ResNet101; FPN; UAV; deep learning; loblolly pine

check for updates

Citation: Cai, C.; Xu, H.; Chen, S.; Yang, L.; Weng, Y.; Huang, S.; Dong, C.; Lou, X. Tree Recognition and Crown Width Extraction Based on Novel Faster-RCNN in a Dense Loblolly Pine Environment. *Forests* **2023**, *14*, 863. https://doi.org/10.3390/f14050863

Academic Editor: Helmi Zulhaidi Mohd Shafri

Received: 22 February 2023
Revised: 1 April 2023
Accepted: 21 April 2023
Published: 23 April 2023

1. Introduction

A tree crown comprises the part of the tree bearing live branches and foliage. Photosynthesis occurs in leaves, and its resulting products are translocated to other tree parts via branches. Therefore, foresters always use the tree crown's characteristics, particularly the crown width, to describe a tree's growth potential. Previous studies have confirmed strong, positive relationships between crown width, tree growth, and carbon sequestration [1]. Hao et al. studied the relationship between teak growth factor and crown width, and established a crown growth prediction model, providing theoretical support for the management of teak plantations [2]. In a 10-year comparative study, Jones et al. demonstrated relationships between crown damage and survival, diameter growth, and tree height growth in Douglas firs [3]. Putney and Maguire studied nitrogen use efficiency in

Douglas fir plantations in western Oregon, where tree growth was measured by changes in crown shape and vertical leaf distribution [4]. Feng et al. argued that the vertical functional variation in leaf traits might indicate niche partitioning within forests [5]. It has been noted that crown width information is also vital in forest modeling, especially for models that include competition indices [6–8]. Therefore, it is of great interest to foresters to develop methods that can accurately measure crown characteristics such as crown width and height.

Tree crown width is often defined as the average width of a tree crown in the north–south and east–west directions [9]. Despite the wide use of tree crown width data in managing forests, accurately measuring crown width is always challenging. Conventional crown width measurement methods include the vertical sighting method [10] and the projection method [11]. The vertical sighting method is quick but less accurate than the projection method. The projection method takes a long time and has low measurement efficiency [12]. However, trees often grow in rows, with tree crowns of varying shapes overlapping, and there is also incompleteness caused by occlusion, making individual tree crown extraction a challenging problem [13]. The use of new techniques to measure crown width has become a hot topic in recent years. With the popularity of smart mobile devices, some scholars have used smartphones to identify and measure tree crowns. For example, Xinmei et al. proposed a passive method for the measurement of tree height and crown diameter based on a smartphone monocular camera [14]. With the development of artificial intelligence and unmanned aerial vehicle (UAV) technology, interest in using UAVs equipped with laser radar and high-definition cameras to measure the crown width of trees is increasing. For example, Ahmadi et al. proposed segmenting early Ganoderma-infected oil palms based on UAV images and artificial neural networks [15]. Safonova et al. proposed a method for extracting tree crowns from UAV images for species classification and stand assessment [16]. Kolanuvada et al. used a UAV paired with a multispectral camera to obtain photos of multiple frequency bands of a forest, employed a simple deep learning convolutional neural network (CNN) to train the images, and developed a linear clustering algorithm to optimize the crown extraction and obtain the crown measurements [17]. Guerra-Hernández et al. used a UAV equipped with an aerial camera and a laser scanner to obtain the high-density 3D point cloud of a eucalyptus plantation, and then conducted 3D modeling to obtain the 3D canopy structure of the eucalyptus forest, which was then incorporated into a prediction of the volume of eucalyptus plantations [18]. Gurumurthy et al. proposed a method for the semantic segmentation of mango trees in high-resolution aerial images and a new method for single crown detection using the segmentation output [19]. To mitigate the impacts of the great homogeneity of neighboring trees and the interlaced crown, Li et al. proposed a crown width estimation method based on an adaptive neuro-fuzzy inference system to improve the intelligence level of crown width estimation [20]. Ritter and Nothdurft proposed a multi-layer seeded region growing-based approach for automatically assessing crown projection areas (CPAs) based on 3D point clouds derived from terrestrial laser scanning (TLS) [21]. In a study based on larch plantations with different stem densities, a two-stage individual tree crown (ITC) segmentation method using airborne light detection and ranging (LiDAR) point clouds was presented [22]. Quan et al. (2019, 2020) evaluated the ability of a UAV laser scanning (UAVLS) system to extract crown structure information from larch plantations [23,24]. They also compared the accuracy of the UAVLS system and airborne laser scanning (ALS) in extracting crown feature attributes. Currently, most crown extraction methods are based on laser scanning and semantic segmentation techniques. Laser scanning technology and segmentation technology can extract more information about the crown, but laser scanning equipment is expensive, and segmentation technology is complicated to use in terms of dataset establishment and it requires outlining along the crown edge. Therefore, it is necessary to find a low-cost, hardware-intensive crown extraction method. The dataset construction of the object-detection model is highly convenient for rapid crown detection and crown width measurement.

Loblolly pine is the second most widely distributed tree species in the United States, and it is the most important commercially in the southeastern United States. Therefore, monitoring the growth of these loblolly pine stands is vital to efficiently manage the stands. In 2021, Lou et al. applied object-detection technology to the measurement of loblolly pine crowns [25]. A UAV was used to obtain the orthophoto images of young and mature stands of *Pinus taeda* in eastern Texas, USA, and three advanced object-detection methods were used to identify the crown and extract the crown width. The faster region-based convolutional neural network (Faster-RCNN) method performed significantly better than the single-shot multibox detector (SSD) on sparse young loblolly pine forests, but on the mature loblolly pine stand, the Faster-RCNN model performed poorly in recognizing the crown and measuring the crown width. The poor performance of Faster-RCNN in the mature stand was unexpected since, in theory, Faster-RCNN is a second-order detector, while you-only-look-once (YOLO) and SSD are single-order detectors. Compared with single-order networks, second-order networks are often more accurate with advantages in multi-scale, high-precision, and small-object detection [26]. Faster-RCNN also outperforms the other two methods in handling the spatial constraints of the algorithm. The main Faster-RCNN improvement is to enhance the adaptability of Faster-RCNN for the crown detection and measurement of both sparse young stands and dense mature stands of loblolly pine. In order to enhance the performance of Faster-RCNN in dense loblolly pine forest sample sites, this study proposes two new Faster-RCNN algorithms, which are then applied to a mature stand to evaluate their accuracy in recognizing tree crowns and measuring tree crown widths.

2. Materials and Methods

2.1. Materials

2.1.1. Image Acquisition

Dataset creation is a critical step in object detection using deep learning models. The study area was located in east Texas, which has a subtropical climate, with heavy rain during the summer.

The study area was located in Rusk, Cherokee County (31°45′31.3″ N, 95°02′318″ W). The site was originally an old field on flat terrain. The site was planted with loblolly pine seedlings in 2001, and in September 2019, when the photos were taken, it had become a mature pine stand with a closed canopy and a high density. The trees averaged 22.2 cm in diameter at breast height (DBH), 16.9 m in total height, and the stand had 35.2 m^2 in basal area per hectare. Figure 1 shows a global orthophoto image of the study area.

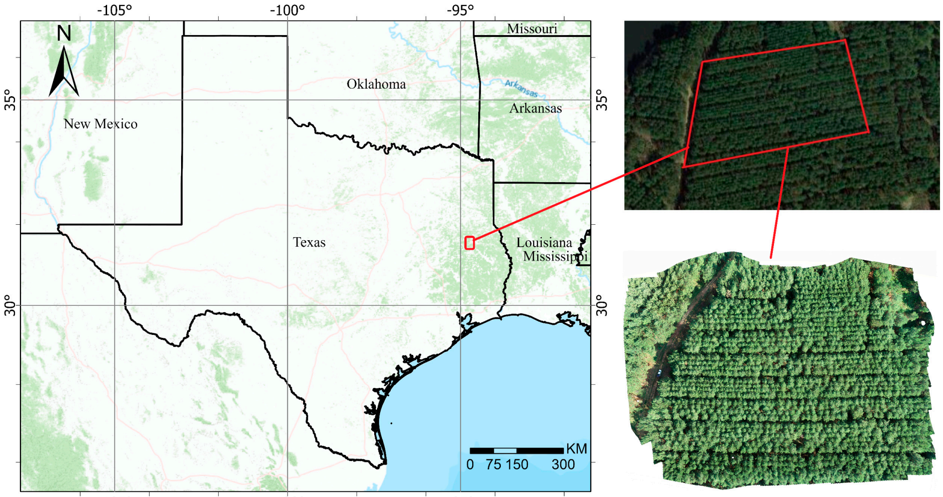
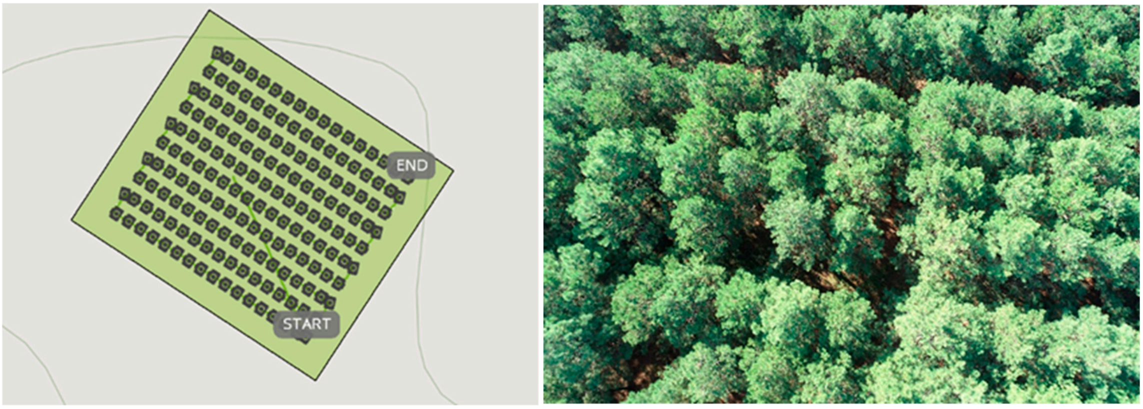


Figure 1. Orthophoto image of loblolly pine plot.

The UAV model used in this study was the DJI Phantom 4 Pro, manufactured by Shenzhen DJI Technology Co., Ltd. This UAV is equipped with a 1-inch 20-million-pixel image sensor with a maximum ascent speed of 6 m/s, a maximum descent speed of 4 m/s, and a maximum horizontal flight speed of 58 km/h in attitude mode. We used Pix4Dcapture (PIX4D) software to control the flight and PhotoScan (v1.2.5) to generate the orthophoto images. Pix4D capture is a mobile flight planning app that allowed us to set flight heights, camera angles, image overlaps, and flight speeds. PhotoScan is an excellent real-world modeling software that automatically generates high-quality 3D models based on images without setting initial values or camera-check calibration. It can process photos according to multi-view 3D reconstruction technology and generate 3D models with real coordinates through control points. In order to maintain sufficient light and to reduce the influence of clouds and ground shadows, the photos were taken during calm periods with stable light intensity. In Pix4Dcapture, we selected the rectangular simple grid route planning mode to instruct the UAV to collect images automatically. The UAV flight parameters were set as follows: an altitude of 46 m, a camera angle of 90° vertically downward, an overlap rate of 90%, and a flight speed of 27 km/h. The original image was in the JPEG format, and the image data included position and orientation system (POS) data, along with precise GPS coordinates. The main orthophoto production steps were as follows: (1) PhotoScan quickly found matching points between all overlapping images, estimated the camera position for each image, and built a sparse point cloud (the processing time depends on the number of photos and the image resolution). (2) A dense point cloud was built. Based on the estimated camera position, the software calculated its depth information and merged it into a dense point cloud model. (3) A grid was generated. After the dense point cloud was reconstructed, a polygon network model was generated based on the dense point cloud data. (4) The DEM model was constructed based on the grid model, and then the high-resolution orthophoto image was generated according to the DEM model. Figure 2 shows the UAV flight routes and the real-time images taken in the study area.



(**a**) (**b**)

Figure 2. (**a**) UAV (unmanned aerial vehicle) flight path planning; (**b**) UAV real-time image.

2.1.2. Image Annotation and Development of the Dataset

In this study, the orthographic images were cut into several 500 × 500-pixel images, each of which contained several loblolly pine tree crowns. LabelImg is a commonly used dataset annotation tool for deep neural network training that is written in Python and uses Qt (a cross-platform C++ graphical-user-interface application-development framework) as its graphical interface. It was used to manually annotate the obtained samples, and the rectangular boxes marked with this tool are shown in Figure 3.

A total of 207 samples were randomly selected from the whole orthoimage as datasets, and the 207 samples were also cut into 500 × 500-pixel datasets. During the training process, the datasets were further divided into training sets and validation sets according to a 9:1

ratio. In the model test section, 185 trees were selected as the test set independent of the training samples. Figure 3 shows the annotated crowns.

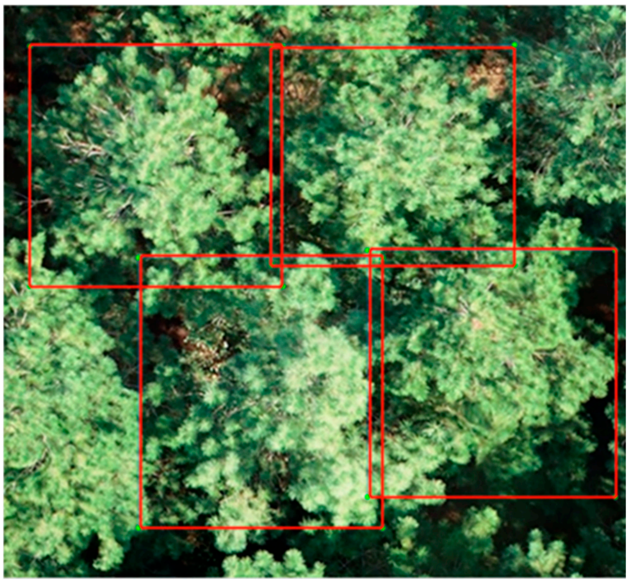


Figure 3. The crowns of each image in the training set were labeled using LabelImg software. The red box is the location of the crown marked by LabelImg.

Each annotated image was saved in PASCAL VOC format as XML files [27]. The file content included the image's path, name, size, and annotated border coordinate.

2.1.3. Image Augmentation

The deep convolution neural network is ideal for many tasks in the field of computer vision. However, using a neural network for object detection generally relies on thousands of pictures for training. Therefore, it is necessary to fine-tune and optimize the model parameters for distinct objects to facilitate the convergence of the model's loss function to its global minimum and enhance its efficacy in detecting diverse objects. However, in the process of real data collection, it is not easy to collect such a huge amount of data; for the model to achieve a better detection result in practical scenarios, and to improve the robustness and generalization ability of the model, data augmentation on the existing dataset is needed [28].

Common augmentation techniques include flipping the image, moving the object position in the image, adding Gaussian noise, improving image contrast, and exposing the image. The crowns in this study exhibited similarities in spectral features. To capitalize on these features, we augmented the dataset by applying operations that manipulate the brightness levels and add Gaussian noise, enhancing the crown's color characteristics. Figure 4 shows a set of data enhancement samples.

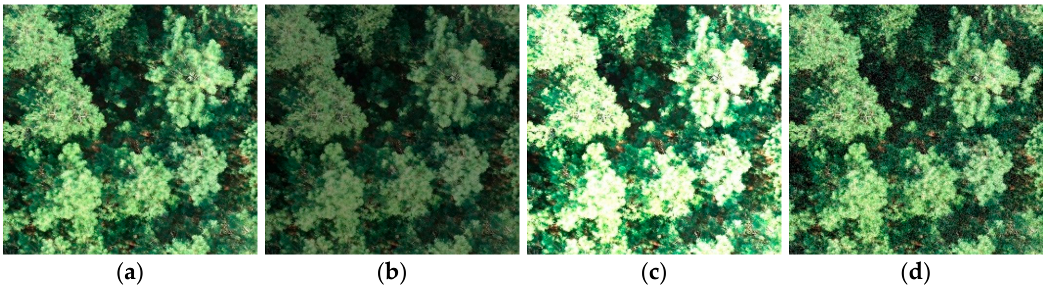


Figure 4. (**a**) Original sample; (**b**) darkened sample; (**c**) variable sample; (**d**) added Gaussian noise sample.

2.2. Methods

To improve the crown recognition and crown width extraction results of different models, the orthophoto map of the whole sample plot was first input into the model for recognition and extraction. However, the orthophoto map was too large so it was cut into several small images using the cropped part of the picture in the crown width extraction program. The size of each small graph was 900 × 900 pixels. To avoid missed detection in the process of object detection, the two connected images maintained a 50% coincidence degree for traversal identification. The prediction box was then scaled and offset. Finally, the crown coordinates were identified. A red detection box was used to mark each identified crown position, and the model could automatically extract the position coordinates of the detection box, such as (990, 1245, 560, 962). Using the position coordinates of the detection box, we computed the number of pixels corresponding to the length and width of the detection box. We then computed the predicted length and width according to the actual length corresponding to a single pixel. Finally, the predicted crown width was calculated by averaging. To measure the actual size of the crown width, we used LabelImg to frame the border of the tree crown, resulting in an XML file with generated position coordinates. The program was then used to extract the coordinates of the framed border to compute the number of pixels, along with the length and width. After extracting the number of pixels for length and width, the real length and width were calculated according to the actual length corresponding to a single pixel. The real crown width was obtained by averaging.

2.2.1. Crown Detection Using Faster-RCNN

As mentioned earlier, object detection in complex environments remains a challenge in machine vision and deep learning. In the field of object detection, RCNN is a classic method. Compared with the traditional method of extracting the target position by traversing images with candidate boxes of different sizes, RCNN introduces the convolutional neural network to extract the depth features, and then maps the extracted features to the classifier, which determines whether the target is contained in the search area and calculates its confidence, obtaining more accurate results.

Ren et al. proposed Faster-RCNN [29], which is based on RCNN and Fast-RCNN [30]. Compared with RCNN and Fast-RCNN, Faster-RCNN has dramatically improved detection accuracy and efficiency. The notable improvement of Faster-RCNN over Fast-RCNN is that it does not use a selective search to create region proposals. However, it introduces a region proposal network (RPN) to extract candidate regions to realize the sharing of convolution features between region proposal and object detection. It can conduct end-to-end training for generating candidate regions, which saves training time.

Faster-RCNN is composed of two parts: Fast-RCNN and RPN. The primary function of RPN is to filter out the high-quality regional proposal boxes in the feature map. Then, the sliding window traverses each point in the feature map and configures k anchor boxes of different sizes on each point. The anchor box is used to extract features, and the softmax is used to determine whether the anchors extract objects that are positive or negative. The bounding box regression is then used to correct them to obtain a more accurate regional proposal. Subsequently, the proposal is input into the region of interest (ROI) pooling layer. This layer mainly transforms the features corresponding to the candidate regions in feature maps and proposals to a fixed size. It inputs the next whole connection layer (classifier) for category judgment and object localization. Figure 5 shows the structure of Faster-RCNN.

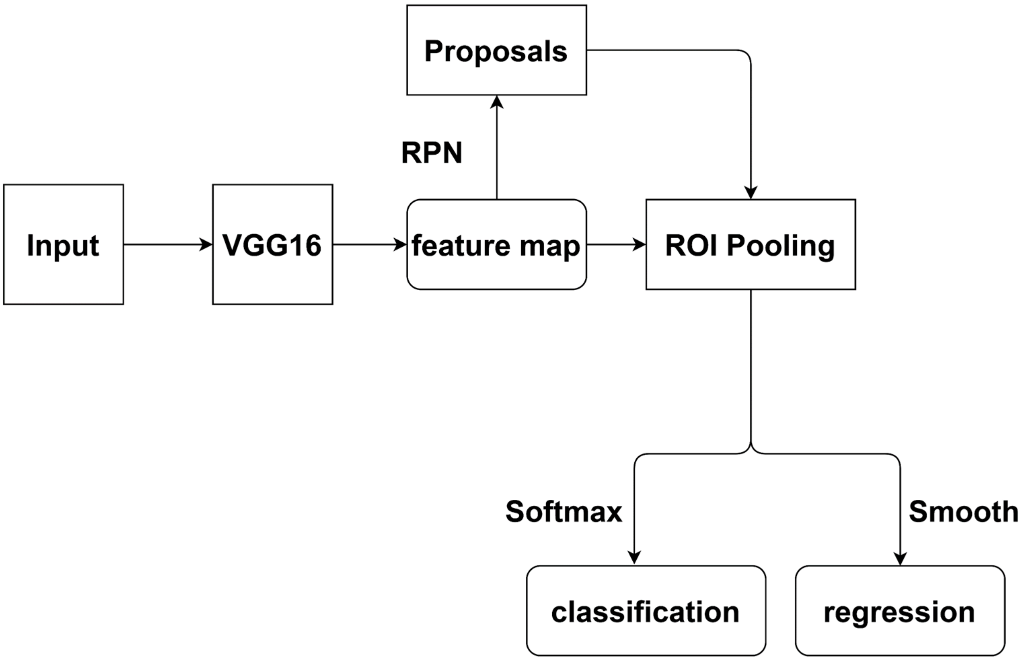


Figure 5. Flow chart of the Faster-RCNN algorithm based on VGG16 backbone network.

2.2.2. Proposed Algorithm: Faster-RCNN with ResNet101

The backbone network of Faster-RCNN is visual geometry group 16 (VGG16) [31], composed of thirteen 3 × 3 convolution layers, three fully connected layers, and several pooling layers. This improves the accuracy of classification results by increasing the number of small convolution kernels and increasing the depth of the network. The network structure is simple and uses the superposition of small convolution kernels instead of large ones, with more nonlinear transformations than a single convolution layer. To further optimize the model recognition effect, this study first adopted the method of deepening the backbone network depth. However, with the deepening of the network, the model may produce gradient disappearance in the training process.

Based on the above premise, this study used ResNet101 [32] to replace VGG16 as the backbone network for feature extraction. Based on the ConvNet model, ResNet introduces numerous identical mappings of $y = x$ across the convolutional layers. Here, x and y represent tensors within the input and output feature maps, respectively. Its main function is to increase the network with depth change without producing the phenomenon of gradient disappearance or weight attenuation. The residual block structure is shown in Figure 6. $F(x)$ and $G(x)$ represent residuals, and $G(x) + x$ is the mapping output; thus, the final network output is $H(x) = G(x) + x$. Since there are three relu functions and three convolution layers in the residual block of the instance, the final framework output results can be expressed as follows:

$$F(x) = relu_1(w_1 \times x) \tag{1}$$

$$G(x) = relu_2(w_2 \times F(x)) \tag{2}$$

$$H(x) = G(x) + x \tag{3}$$

Figure 6 shows the specific structure of the residual block.

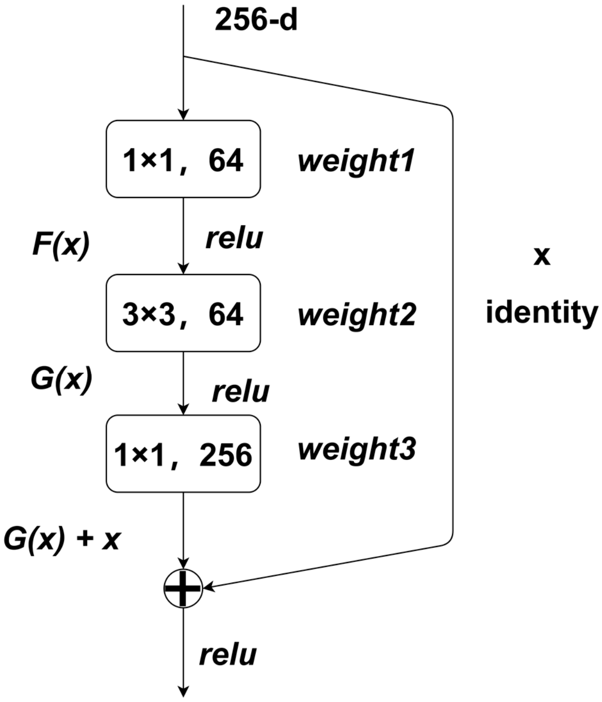


Figure 6. ResNet101 second-layer residual block structure.

The specific network structure of ResNet101 used in this experiment is shown in Figure 7.

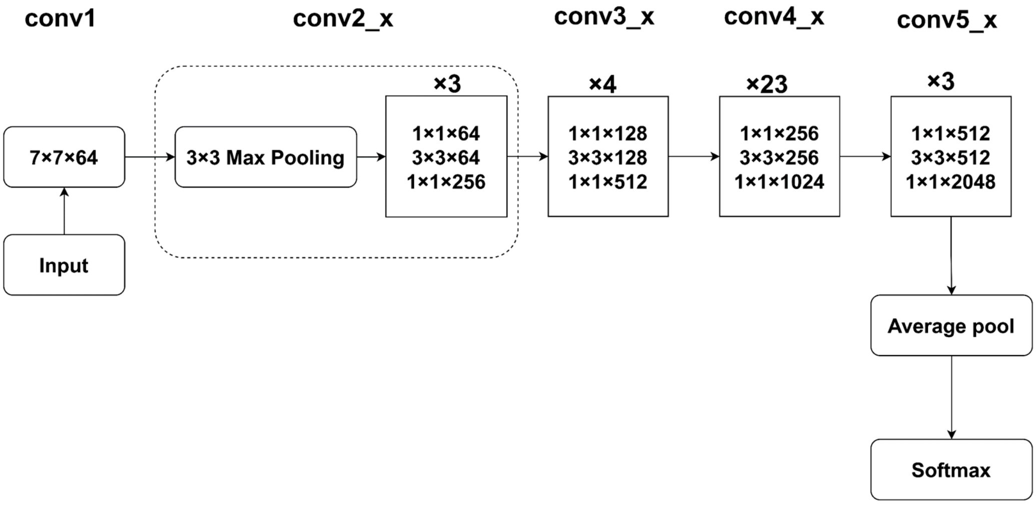


Figure 7. ResNet101 structure chart.

2.2.3. Proposed Algorithm: Faster-RCNN with ResNet101 and FPN

To solve the problem of deep information loss that may occur when ResNet101 replaces VGG16 as the backbone network, this study proposed a combination of ResNet101 and a feature pyramid network to create the FPN_ResNet101 structure. The feature pyramid network (FPN), proposed by Lin et al. [33], is a top-down feature fusion method with horizontal connection. Common object-detection algorithms only use top-level features to predict, while shallow location information is lost. Figure 8 shows the structure of FPN fusing high-level and shallow features for prediction.

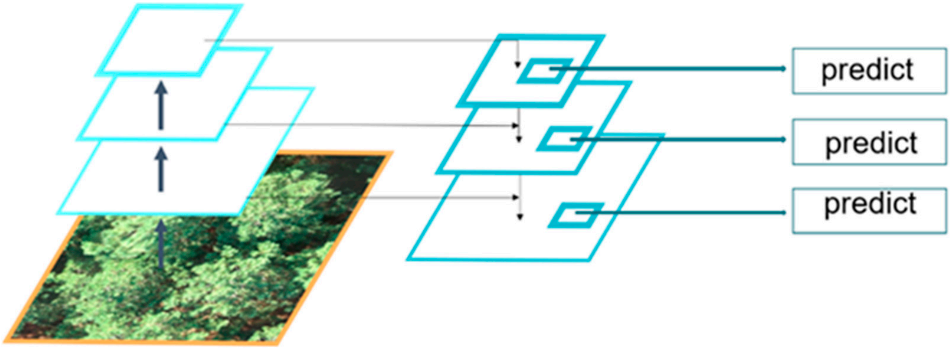


Figure 8. FPN structure.

Finally, the Faster-RCNN model based on FPN and ResNet101 was improved in this study. Since the canopy occupies most of the area of each image in the dataset, while the background area occupies only a tiny portion, to output the canopy color features with greater weight during the training process, the RGB averaging module was added before the base FPN_ResNet1010 structure. An image-averaging operation was performed before inputting each dataset into the model. The resulting values were input to the model as part of the parameters to facilitate more targeted canopy color features trained in the model. The FPN_ResNet101 structure replaced the VGG16, and the Region Proposal Network (RPN) in the Faster-RCNN was scale-separated. The FPN can fuse different scales for detection, and it comprises a three-stage architecture that involves bottom-up feature map generation at multiple scales, top-down feature enhancement, and lateral connections. Given the convolutional outputs at different levels, denoted by Cx, the intermediate feature maps represented by Mx, and the ultimately fused feature map illustrated by Px, the three components are mutually aligned. In the five feature layers of FPN, anchors with different sizes were defined, which were 32 × 32, 128 × 128, 256 × 256, and 512 × 512. There were three ratios of 1:1, 1:2, and 2:1. Therefore, there were 15 anchors. The improved model structure of FPN_Faster-RCNN_ResNet101 is shown in Figure 9.

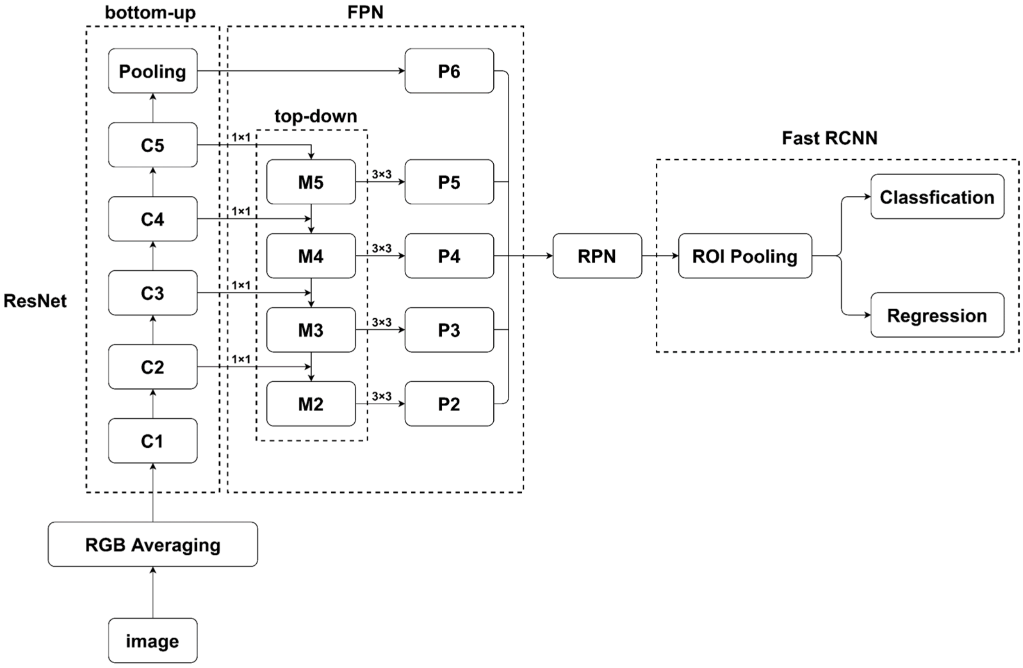


Figure 9. Structure diagram of FPN_Faster-RCNN_ResNet101, where C represents convolutional outputs, M denotes features maps, and P is fused feature maps.

3. Results and Discussion

3.1. Experimental Procedures and Metrics

3.1.1. Experimental Configuration and Dataset

The network training configuration environment was Windows 11, Intel$^{(R)}$ Core^{TM} i7-10750H CPU@2.60 GHz processor, 16 GB memory, and NVIDIA GeForce GTX 1650Ti with 4 GB of video memory as the GPU. Microsoft headquarters in Redmond, Washington, USA. Intel's headquarters and NVIDIA's headquarters are both located in Santa Clara, California, USA. The experimental environment was Python 3.6, TensorFlow-GPU1.12, CUDA9.0, and CUDNN7.3.

Since the Faster-RCNN model requires a large amount of data training to improve its robustness, but the number of existing datasets is limited, migration learning helps to improve this situation. Specifically, it trains on a large dataset and then takes the obtained weight as the training initialization parameter. This study used the initial weights of ResNet101 network model weights from pre-training on the ImageNet dataset. The total number of iterations was 20,000, and the model was saved every 5000 times. The learning rate was set to 0.001, and the batch_size was set to 256. The FPN_Faster-RCNN_ResNet101 model selected the ResNet101 network model, which was pre-trained on the ImageNet dataset for initialization training. The format of the dataset was VOC, and the input image size was set to 512 × 512.

3.1.2. Evaluation Index

For the model evaluation, it is necessary to evaluate the crown recognition and crown width extraction of the model, respectively.

The crown recognition was evaluated by calculating the accuracy, precision, recall, and F1-score:

$$\text{Accuracy } = \frac{TP}{TP + FP + FN} \tag{4}$$

$$\text{Precision } = \frac{TP}{TP + FP} \tag{5}$$

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

$$F1\text{-Score} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{7}$$

where *TP* is the number of correctly divided positive cases (i.e., the number of correctly identified crowns); *FP* represents the number of incorrectly divided positive cases (i.e., the number of incorrectly identified crowns); and *FN* denotes the number of incorrectly divided negative cases (in this paper, the number of unidentified crowns).

Among the four indexes of crown recognition (Equations (4)–(7)), the accuracy is used to reflect the ability of the model to predict the whole sample, the precision is used to reflect the proportion of the real target in the model prediction, the recall rate is used to reflect the proportion of the model prediction positive cases to the number of real positive cases, and the F1-score, also called the balanced F score, is defined as the harmonic average of the precision and the recall rate.

In the crown width extraction part, the following three indicators were calculated to evaluate the accuracy of the crown width model (Equations (8)–(10)). Bias represents the deviation between the estimated value and the actual value. The accuracy of the crown width model is demonstrated by calculating the root mean square error (*RMSE*) and the coefficient of determination (R^2):

$$Bias = \frac{1}{N} \sum_{i=1}^{N} |\hat{y}_i - y_i| \tag{8}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{N}} \tag{9}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \overline{y})^2} \tag{10}$$

where $\hat{y}_i$ represents the estimated value; y_i denotes the actual value; N is the number of samples; and $\overline{y} = \frac{1}{N}\sum_{i=1}^{N} y_i$.

3.2. Results and Discussion

3.2.1. Identify Impressions

In our previous study, the Faster-RCNN, YOLO, and SSD models achieved good results in young forests, for which Faster-RCNN had the highest recognition accuracy. Figure 10 shows the crown-detection effect of the three models on young forests, and Table 1 presents their respective detection results. The data comes from "Measuring loblolly pine crowns with drone imagery through deep learning" [25]. Faster-RCNN outperformed the other two methods in young forest detection.

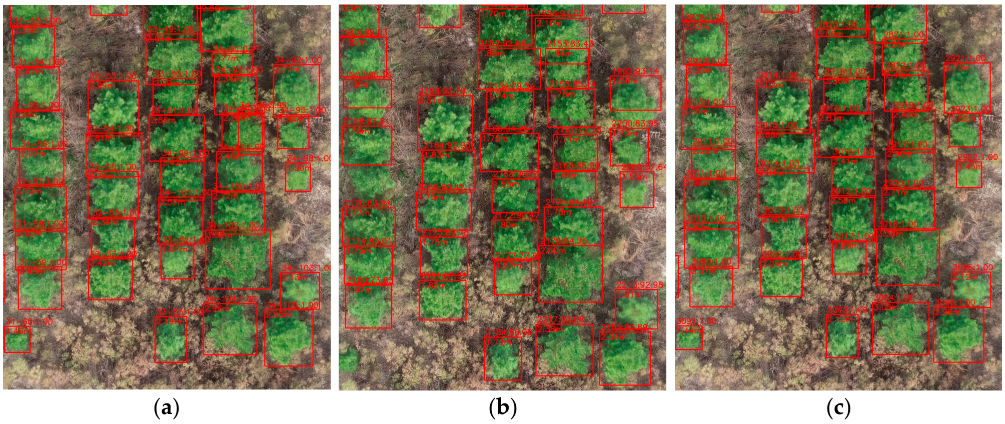


Figure 10. (**a**) Faster-RCNN detection effect image; (**b**) YOLO detection effect image; (**c**) SSD detection effect image. The red square is the crown detected by the model.

Table 1. Classification detection results of Faster-RCNN, YOLO, and SSD [25].

Index	Faster-RCNN	YOLO	SSD
TP	128	126	125
FP	1	6	1
FN	0	2	3
Precision (%)	99.22	95.45	99.21
Recall (%)	100.00	98.44	97.66
Accuracy (%)	99.22	94.03	96.90
*F*1-score (%)	99.61	96.92	98.43

However, in the mature forest, the original Faster-RCNN model performed poorly. The objective of this research was to enhance the performance of Faster-RCNN and enhance its versatility when operating in mature forest environments. Figure 11 shows the crown-detection effect of the two models in orthophoto images.

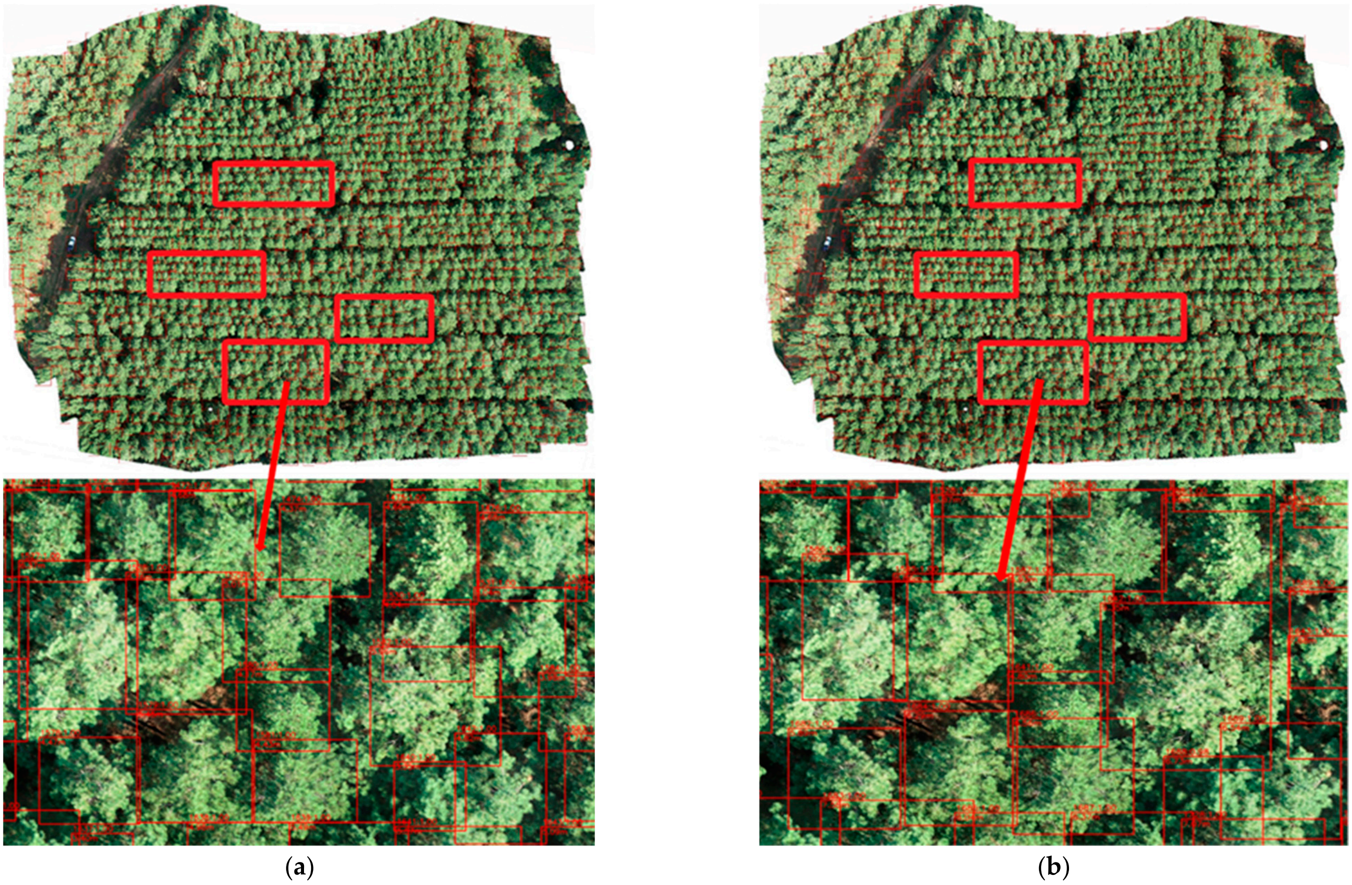

(**a**) (**b**)

Figure 11. (**a**) The Faster-RCNN_ResNet101 model with ResNet101 as the backbone network; (**b**) the FPN_Faster-RCNN_ResNet101 model with FPN_ResNet101 as the backbone network. The red square is the crown detected by the model.

3.2.2. Crown Identification

We use Method 1 to represent YOLO, Method 2 to represent SSD, Method 3 to represent Faster-RCNN_ResNet101, and Method 4 to represent FPN_Faster-RCNN_ResNet101.

Due to the slow growth of trees, in crown identification, the accurate detection of each canopy is more critical than rapid crown detection, so models with higher accuracy are more suitable for this task. In this experiment, we used the computer mentioned in Section 3.1.1 as the experimental equipment, and we selected the single crown recognition time to measure the model detection speed. Two-stage detector recognition speed is slower than a one-stage detector, but the accuracy is higher. It can be seen from Table 2 that Method 1 and Method 2 were faster than Method 3 and Method 4. However, in this task, FPN_Faster-RCNN_ResNet101 (Method 4) was better than SSD in recall, accuracy, and F1-score, but slightly worse than SSD in precision. The two-stage detector, FPN_Faster-RCNN_ResNet101 (Method 4), gave the best overall results, achieving better accuracy than Method 2, even at a similar speed.

Moreover, Method 4 improved the accuracy by 4.9% over Method 3, which also proved the feasibility of the improved method.

Table 2. Classification detection results of YOLO, SSD [25], and two improved Faster-RCNN algorithms.

Index	(1) YOLO	(2) SSD	(3) Faster-RCNN_ResNet101	(4) FPN_Faster-RCNN_ResNet101
Time (ms)	55	57	72	69
TP	175	180	170	181
FP	3	4	13	5
FN	10	5	6	4
Precision (%)	98.31	97.83	93.19	97.31
Recall (%)	94.59	97.30	96.74	97.84
Accuracy (%)	93.09	95.24	90.36	95.26
*F*1-Score (%)	96.42	97.56	94.93	97.58

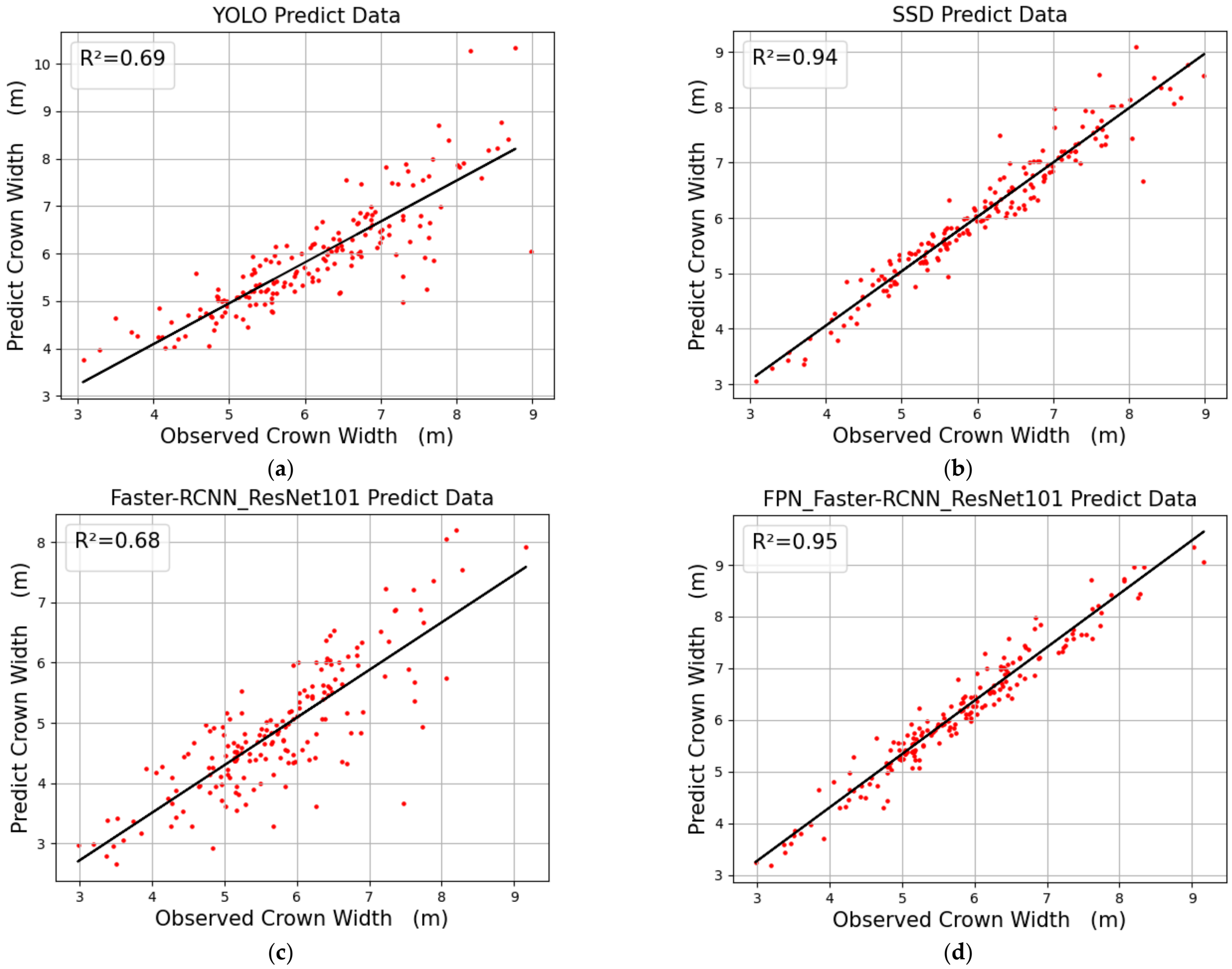
In the actual training process, with the deepening of the network depth, the gradient is backward propagation. After increasing the network depth, the forward gradient will be minimal, while the model also has problems such as learning stagnation and gradient disappearance. Table 1 presents the models' performance in crown recognition based on the independent test dataset. After replacing VGG16 with ResNet101, Method 3 improved the efficiency of crown recognition, with the precision, recall, accuracy, and F1-score reaching 93.19%, 96.74%, 90.36%, and 94.93%, respectively. The accuracy and precision were comparable, although slightly weaker, than those for Method 1 and Method 2. After fusing FPN and ResNet101, VGG16 was replaced by the FPN_ResNet101 structure. In crown recognition, the four indexes of Method 4 were improved to varying degrees, of which accuracy was the most improved, reaching 95.26%. Compared with Method 3, the four indexes increased by 4.12%, 1.10%, 4.90%, and 2.65%, respectively. Using the FPN to help detect objects at different scales can theoretically improve the small-target-detection effect of the model. The experimental results in Table 1 also prove this. It was verified that the improved method helps to enhance the canopy detection performance of Faster-RCNN in dense loblolly pine forests, and the feasibility of the improved means was well illustrated.

3.2.3. Extraction of Crown Width

Table 3 and Figure 12 present the results of the models estimating crown width using the independent test dataset. Overall, the application of Method 3 did not achieve the same accuracy and precision as Method 1 and Method 2. Through the study of the residual block structure, it was found that the ResNet101 network has a deep information loss problem. In ResNet101, identity mapping must be used when the size of the building block does not match the size of the next building block. According to Figure 7, in the four mapping stages of ResNet101, there are only four continuous 1 × 1 convolutions, but there is no linear relationship between the two, which limits its learning ability and eventually leads to the loss of deep information.

Table 3. The mature loblolly pine crown-width-measurement effect index of YOLO, SSD [25], and two improved Faster-RCNN algorithms.

Index	(1) YOLO	(2) SSD	(3) Faster-RCNN_ResNet101	(4) FPN_Faster-RCNN_ResNet101
Bias (m)	0.92	0.99	1.15	0.98
RMSE (m)	0.66	0.31	1.06	0.45
R^2	0.69	0.94	0.68	0.95



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

Figure 12. (**a**) Linear regression graphs of (1) the YOLO model; (**b**) linear regression graphs of (2) the SSD model; (**c**) linear regression graphs of (3) the Faster-RCNN_ResNet101 model; (**d**) linear regression graphs of (4) the FPN_Faster-RCNN_ResNet101 model.

The *FPN* is a way to fuse low-level and high-level features. The shallow feature map has a small receptive field and less semantic information, but the spatial location information is accurate. After the fusion of ResNet101 using the *FPN*, we created a new network structure, named FPN_ResNet101, and applied this structure to Faster-RCNN. Method 4 measured crown width very accurately and precisely, resulting in a bias of 0.98, an *RMSE* of 0.45, and an R^2 of 0.95. These estimates were comparable to those of Method 2, but more improved than those of Method 1. Compared with Method 3, the *RMSE* decreased by 0.61 and the R^2 increased by 0.27.

FPN_Faster-RCNN_ResNet101 offers a huge improvement in crown width measurement, with a higher R^2 than all the other methods. The feasibility of using *FPN* and ResNet101 to improve the original model is illustrated.

4. Conclusions

In this study, high-resolution orthophotos, obtained by UAVs shooting a mature loblolly pine forest in eastern Texas, were used as the data source. ResNet101 and FPN_ResNet101 replaced the backbone network VGG16 of the original Faster-RCNN model. Using FPN_ResNet101, the crown recognition accuracy rate of Method 4 reached 95.26%, and the crown width extraction R^2 reached 0.95. Compared with Method 3, the two indexes had increased by 4.90% and 0.27, respectively, which proves the feasibility of improving the original model using FPN_ResNet101 network architecture and the superiority of the improved model in this research field. At the same time, with regard to recognition speed, the improved Method 4 (FPN_Faster-RCNN_ResNet101) was also enhanced to a certain extent in comparison with Method 3 (Faster-RCNN_ResNet101). The speed of the two-stage detector was improved to a level similar to that of the single-stage detector, and some progress was made in comparison with Method 2 (SSD) in terms of crown detection and crown width extraction. However, due to the similarity of trees, the accurate identification and classification of tree crowns in mixed forests remains a significant challenge. In terrain such as hills, accurate canopy width measurement is impossible due to the change in relative distance between the UAV and the ground, which is a crucial direction for future research. Nonetheless, the excellent accuracy of Faster-RCNN suggests the model's applicability in dense loblolly pine forests, providing an alternative for forestry practitioners in tree mensuration.

Author Contributions: Conceptualization, X.L. and C.D.; formal analysis, X.L., C.D., C.C. and L.Y.; funding acquisition, Y.W. and C.D.; methodology, H.X., S.C. and S.H.; resources, X.L. and Y.W.; writing—original draft, C.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Zhejiang Natural Science Foundation Project, grant number LQ21C160018, the ETPPRP, and the McIntire Stennis program.

Data Availability Statement: Not applicable.

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

References

1. Gratani, L.; Varone, L. Plant Crown Traits and Carbon Sequestration Capability by *Platanus hybrida* Brot. in Rome. *Landsc. Urban Plan.* **2007**, *81*, 282–286. [CrossRef]
2. Hao, J.; Jia, H.; Yang, B.; Huang, X.; Huang, G. Regession analysis of Teak Crown Growth with Tree Height and DBH. *J. Northwest For. Univ.* **2019**, *34*, 144–148.
3. Jones, D.A.; Harrington, C.A.; Marshall, D. Survival, and Growth Response of Douglas-Fir Trees to Increasing Levels of Bole, Root, and Crown Damage. *For. Sci.* **2019**, *65*, 143–155. [CrossRef]
4. Putney, J.D.; Maguire, D.A. Shifts in Foliage Biomass and Its Vertical Distribution in Response to Operational Nitrogen Fertilization of Douglas-Fir in Western Oregon. *Forests* **2020**, *11*, 511. [CrossRef]
5. Feng, J.; Lian, J.; Mei, Q.; Cao, H.; Ye, W. Vertical Variation in Leaf Traits and Crown Structure Promote the Coexistence of Forest Tree Species. *Forests* **2022**, *13*, 1548. [CrossRef]
6. Bella, I.E. A New Competition Model for Individual Trees. *For. Sci.* **1971**, *17*, 364–372.
7. Röhle, H. Vergleichende Untersuchungen zur Ermittlung der Genauigkeit bei der Ablotung von Kronenradien. *Forstarchiv* **1986**, *57*, 67–71.
8. Thurnher, C.; Klopf, M.; Hasenauer, H. MOSES—A Tree Growth Simulator for Modelling Stand Response in Central Europe. *Ecol. Model.* **2017**, *352*, 58–76. [CrossRef]
9. Fu, L.; Sharma, R.P.; Wang, G.; Tang, S. Modelling a System of Nonlinear Additive Crown Width Models Applying Seemingly Unrelated Regression for Prince Rupprecht Larch in Northern China. *For. Ecol. Manag.* **2017**, *386*, 71–80. [CrossRef]
10. Preuhsler, T. Ertragskundliehe Merkmale oberbayerlscher Bergmischwald-Verjüngungsbestände auf kalkalpinen Standorten im Forstamt Kreuth. *Forstwiss. Cent.* **1981**, *100*, 313–345. [CrossRef]
11. Fleck, S.; Mölder, I.; Jacob, M.; Gebauer, T.; Jungkunst, H.F.; Leuschner, C. Comparison of Conventional Eight-Point Crown Projections with LIDAR-Based Virtual Crown Projections in a Temperate Old-Growth Forest. *Ann. For. Sci.* **2011**, *68*, 1173–1185. [CrossRef]
12. Pretzsch, H.; Biber, P.; Uhl, E.; Dahlhausen, J.; Rötzer, T.; Caldentey, J.; Koike, T.; van Con, T.; Chavanne, A.; Seifert, T.; et al. Crown Size and Growing Space Requirement of Common Tree Species in Urban Centres, Parks, and Forests. *Urban For. Urban Green.* **2015**, *14*, 466–479. [CrossRef]
13. Ning, X.; Ma, Y.; Hou, Y.; Lv, Z.; Jin, H.; Wang, Z.; Wang, Y. Trunk-Constrained and Tree Structure Analysis Method for Individual Tree Extraction from Scanned Outdoor Scenes. *Remote Sens.* **2023**, *15*, 1567. [CrossRef]
14. Wu, X.; Xu, A.; Yang, T. Passive Measurement Method of Tree Height and Crown Diameter Using a Smartphone. *IEEE Access* **2020**, *8*, 11669–11678.
15. Ahmadi, P.; Mansor, S.; Farjad, B.; Ghaderpour, E. Unmanned Aerial Vehicle (UAV)-Based Remote Sensing for Early-Stage Detection of Ganoderma. *Remote Sens.* **2022**, *14*, 1239. [CrossRef]

16. Safonova, A.; Hamad, Y.; Dmitriev, E.; Georgiev, G.; Trenkin, V.; Georgieva, M.; Dimitrov, S.; Iliev, M. Individual Tree Crown Delineation for the Species Classification and Assessment of Vital Status of Forest Stands from UAV Images. *Drones* **2021**, *5*, 77. [CrossRef]
17. Kolanuvada, S.R.; Ilango, K.K. Automatic Extraction of Tree Crown for the Estimation of Biomass from UAV Imagery Using Neural Networks. *J. Indian Soc. Remote Sens.* **2021**, *49*, 651–658. [CrossRef]
18. Guerra-Hernández, J.; Cosenza, D.N.; Cardil, A.; Silva, C.A.; Botequim, B.; Soares, P.; Silva, M.; González-Ferreiro, E.; Díaz-Varela, R.A. Predicting Growing Stock Volume of Eucalyptus Plantations Using 3-D Point Clouds Derived from UAV Imagery and ALS Data. *Forests* **2019**, *10*, 905. [CrossRef]
19. Gurumurthy, V.A.; Kestur, R.; Narasipura, O. Mango Tree Net—A Fully Convolutional Network for Semantic Segmentation and Individual Crown Detection of Mango Trees. *arXiv* **2019**, arXiv:1907.06915.
20. Li, Y.; Zhang, H.; Yang, T.; Ma, Z.; Li, S.; Shen, K. A Method of Estimating Chinese Fir Crown Width Based on Adaptive Neuro-Fuzzy Inference System. *Sci. Silvae Sin.* **2019**, *55*, 45–51.
21. Ritter, T.; Nothdurft, A. Automatic Assessment of Crown Projection Area on Single Trees and Stand-Level, Based on Three-Dimensional Point Clouds Derived from Terrestrial Laser-Scanning. *Forests* **2018**, *9*, 237. [CrossRef]
22. Ma, Z.; Pang, Y.; Wang, D.; Liang, X.; Chen, B.; Lu, H.; Weinacker, H.; Koch, B. Individual Tree Crown Segmentation of a Larch Plantation Using Airborne Laser Scanning Data Based on Region Growing and Canopy Morphology Features. *Remote Sens.* **2020**, *12*, 1078. [CrossRef]
23. Quan, Y.; Li, M.; Zhen, Z.; Yuanshuo, H. Modeling Crown Characteristic Attributes and Profile of Larix Olgensis Using UAV-Borne Lidar. *J. Northeast Univ.* **2019**, *47*, 52–28.
24. Quan, Y.; Li, M.; Zhen, Z.; Hao, Y.; Wang, B. The Feasibility of Modelling the Crown Profile of Larix Olgensis Using Unmanned Aerial Vehicle Laser Scanning Data. *Sensors* **2020**, *20*, 5555. [CrossRef] [PubMed]
25. Lou, X.; Huang, Y.; Fang, L.; Huang, S.; Gao, H.; Yang, L.; Weng, Y.; Hung, I.K. Measuring Loblolly Pine Crowns with Drone Imagery through Deep Learning. *J. For. Res.* **2022**, *33*, 227–238. [CrossRef]
26. Soviany, P.; Ionescu, R.T. Optimizing the Trade-Off between Single-Stage and Two-Stage Deep Object Detectors Using Image Difficulty Prediction. In Proceedings of the 2018 20th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), Timisoara, Romania, 20–23 September 2018; pp. 209–214.
27. Everingham, M.; Van Gool, L.; Williams, C.K.; Winn, J.; Zisserman, A. The Pascal Visual Object Classes (Voc) Challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–338. [CrossRef]
28. Perez, L.; Wang, J. The Effectiveness of Data Augmentation in Image Classification Using Deep Learning. *arXiv* **2017**, arXiv:1712.04621.
29. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Red Hook, NY, USA, 2015; Volume 28.
30. Girshick, R. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
31. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2015**, arXiv:1409.1556.
32. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
33. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.

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.