electronics

MDPI

Article

The Semantic Segmentation of Standing Tree Images Based on the Yolo V7 Deep Learning Algorithm

Lianjun Cao [1,2,3], Xinyu Zheng [1,2,3] and Luming Fang [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, Hangzhou 311300, China
[3] Key Laboratory of Forestry Intelligent Monitoring and Information Technology of Zhejiang Province, Hangzhou 311300, China
[*] Correspondence: fluming@126.com

Abstract: The existence of humans and the preservation of the natural ecological equilibrium depend greatly on trees. The semantic segmentation of trees is very important. It is crucial to learn how to properly and automatically extract a tree's elements from photographic images. Problems with traditional tree image segmentation include low accuracy, a sluggish learning rate, and a large amount of manual intervention. This research suggests the use of a well-known network segmentation technique based on deep learning called Yolo v7 to successfully accomplish the accurate segmentation of tree images. Due to class imbalance in the dataset, we use the weighted loss function and apply various types of weights to each class to enhance the segmentation of the trees. Additionally, we use an attention method to efficiently gather feature data while reducing the production of irrelevant feature data. According to the experimental findings, the revised model algorithm's evaluation index outperforms other widely used semantic segmentation techniques. In addition, the detection speed of the Yolo v7 model is much faster than other algorithms and performs well in tree segmentation in a variety of environments, demonstrating the effectiveness of this method in improving the segmentation performance of the model for trees in complex environments and providing a more effective solution to the tree segmentation issue.

Keywords: tree segmentation; semantic segmentation; fast segmentation; Yolo v7; deep learning

check for updates

Citation: Cao, L.; Zheng, X.; Fang, L. The Semantic Segmentation of Standing Tree Images Based on the Yolo V7 Deep Learning Algorithm. *Electronics* **2023**, *12*, 929. https://doi.org/10.3390/electronics12040929

Academic Editors: Aytaç Altan and Gwanggil Jeon

Received: 13 January 2023
Revised: 7 February 2023
Accepted: 11 February 2023
Published: 13 February 2023

Copyright: © 2023 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

The existence of humans and the preservation of the natural ecological balance depend greatly on trees. They conserve the ecological variety of animals and plants, produce wood and other goods for people, offer habitat and food for wild animals, absorb carbon dioxide, release oxygen, filter the air, maintain water and soil, and stop soil erosion [1].

In order to extract the crown, diameter at breast height (DBH), and other information from standing tree images, semantic segmentation is necessary [2]. This study is crucial to the subject of Digital Forestry. A Region-based Convolutional Neural Network (R-CNN) segmentation approach based on an RGB-Depth Map (RGB-D) Image and Improved Mask R-CNN semantic segmentation of citrus crowns in orchards was suggested by Cong et al. [3]. In order to semantically segment changes in forests in aerial images, Pyo et al. employed the U-Network (U-Net) model of convolution deep learning architecture [4]. For the semantic segmentation of remote sensing images, Marsocci et al. employed a Self-Supervised Multi-Attention Residual U-Network (ResU-Net) [5].

Deep learning has been widely applied in a variety of industries, including facial recognition, automated driving, and intelligent robotics [6]. Yolo series algorithms have many applications in deep learning. Li et al.'s Yolo-Based Traffic Sign Recognition Algorithm application reduced the potential safety hazards caused by human cognitive errors [7]. The Real-Time Human Ear Detection developed by Quoc et al., based on the joining of

Yolo and RetinaFace, could recognize humans with masks and could diagnose ear-related diseases [8]. The Yolo series of algorithms is based on convolutional neural networks, which were developed from early backpropagation (BP) neural networks; they have very good generalization ability and can handle object recognition problems in images well. Early Yolo series algorithms mainly used convolution and pooling as feature extraction and processing methods. A fully activated module was added to Yolo 2.0 to make the network more stable and accurate. The algorithm has undergone some optimizations and added an attention mechanism, and its detection accuracy is higher.

The semantic segmentation of standing tree images based on the Yolo v7 deep learning algorithm in this work is novel [9]. A number of segmentation models have been put forth in the field of image segmentation that successfully address a number of the issues with traditional segmentation, including the semi-manual operation [10], imprecise segmentation [11], and inaccurate targeting of an object [12]. The semantic segmentation model based on Yolo v7 was selected. The Yolo series of algorithms are relatively popular, but research and application for tree segmentation is basically absent; Yolo v7 is a new algorithm recently released by the Yolo series, which represents a Yolo series algorithm with better effect. Yolo v7's detection speed and accuracy are higher than all previous Yolo versions, and it is suitable for the semantic segmentation of trees. In addition to the advantages of the algorithm itself, Yolo v7 was also partially improved, and the attention mechanism was added to the algorithm to improve accuracy.

Semantic segmentation of standing tree images based on the Yolo v7 algorithm is the main research contribution of this paper. Among the many algorithms for deep learning, the latest and appropriate Yolo v7 algorithm was selected to ensure the applicational feasibility of tree segmentation. In order to further improve the accuracy of the algorithm, the Yolo v7 algorithm was partially modified, and the attention mechanism was added. In order to deal with tree semantic segmentation with a complex background, the weighted loss function was introduced. The optimized Yolo v7 algorithm mainly improves the accuracy of its detection. Due to the advantages of the selected algorithm, its detection speed is also far faster than other algorithms. It realizes the fast and accurate semantic segmentation of a tree, which can be applied to fields such as digital intelligent agriculture. Tree samples from different environments in two cities were collected; the specific design is explained in the experimental section. The optimized semantic segmentation effect of Yolo v7 on a tree is detailed in the results section. The algorithm of the Yolo series is mainly composed of four parts, namely the feature extraction layer, the feature enhancement layer, the detection layer, and the postprocessing layer. The first three parts are processed with a traditional convolution layer, and the last two parts are processed with a convolution and pooling layer. In feature extraction, the basis of Yolo series algorithms is to extract effective information or features from images. The Yolo v7 algorithm also uses a convolution neural network, which is composed of three convolution modules: the LSTM module, the Dropout module, and the Softmax output module. The LSTM module is mainly detected through connection and full activation, the Dropout module is mainly added after the first two networks to prevent overfitting, and the output part of the Softmax is processed by using convolution.

The remainder of this article is structured as follows. The core concepts and fundamental tenets of the Yolo v7 model suggested in this article are explained in depth in Section 2. The experiment's setup and procedure are described in Section 3. The outcomes of the experimental comparison are examined in Section 4. Finally, the conclusion of the entire experimental effort is summarized in Section 5.

2. Yolo V7 Model

2.1. Introduction of the Yolo V7 Model

The Yolo series has developed along with deep learning, starting with Yolo v1 in 2015 and progressing to Yolo v2 in 2016, Yolo v3 in 2018, Yolo v4 and Yolo v5 in 2020, and most recently Yolo v6 and Yolo v7 [13–19]. Yolo v7 not only performs target identification, but also has applications in case segmentation and human posture assessment. In the mask branch, Yolo v7's instance segmentation is a SingleStage approach, which contributes to its efficiency in this area [20]. The OrienMaskHead is utilized at the moment, and more approaches may be implemented down the road [21].

Figure 1 below displays the Yolo v7 framework network procedure in its entirety. The input image is first resized to 640 × 640 pixels before being fed into the backbone network. Following that, three feature map layers of varying sizes are generated via the head layer network, and Rep and conv are used to output the prediction outcomes. If the dataset belongs to Coco, each output (x, y, w, h, o) is the coordinate location and the backdrop before and after, and there are three anchors. The outputs are divided into 80 categories. The ultimate output of each layer is therefore (80 + 5) × 3 = 255 multiplied by the size of the feature map.

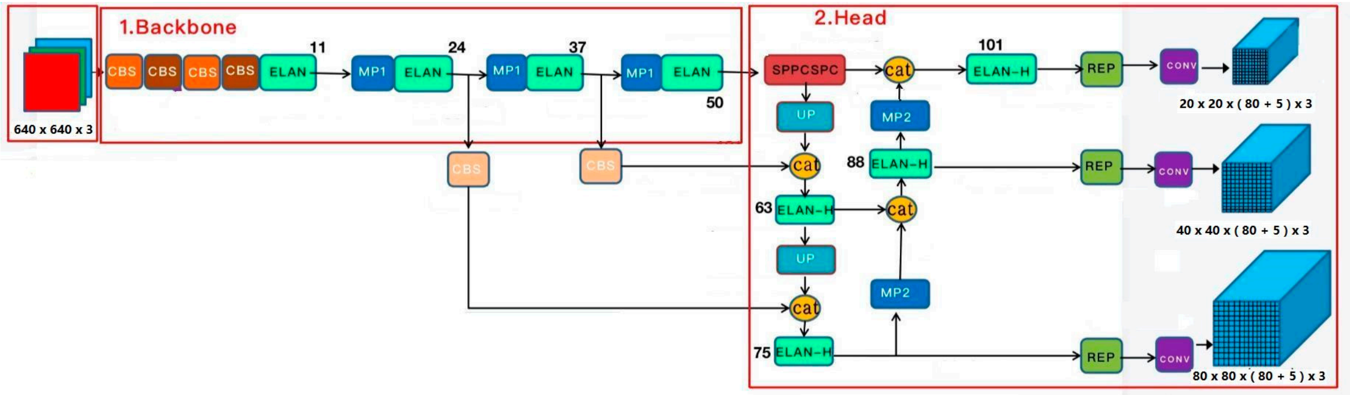

Figure 1. Yolo v7's overall framework network process.

2.2. The Main Pros and Cons of Yolo V7

Yolo v7 is 120% faster (FPS) than Yolo v5, 180% faster (FPS) than Yolo X, 1200% faster (FPS) than Dual-Swin-T, 550% faster (FPS) than ConvNext, and 500% faster (FPS) than SWIN-L [19]. It is also more accurate than Yolo v5 with the same volume. The accuracy and speed of Yolo v7 have both increased compared to previous Yolo algorithms.

Yolo v7, which was evaluated on a GPU V100, outperformed the currently available detectors in the speed and accuracy range of 5 to 160 frames per second. A detection rate of more than 30 FPS may be attained by the model with an accuracy of 56.8% AP (batch = 1). It is also the only detector capable of exceeding 30 FPS while maintaining such high accuracy [19].

In deep learning, some algorithms pursue algorithm accuracy, while others pursue algorithm speed. Yolo pursues algorithm speed while considering algorithm accuracy, though algorithm accuracy is not the main focus. Yolo also has peculiar format requirements for data annotation, which also increases the complexity of application.

2.3. Convolutional Block Attention Module (CBAM)

The channel attention module (CAM) and spatial attention module (SAM) make up the convolutional block attention module (CBAM). While the SAM enables the network to concentrate on the locations that are rich in context information in the whole image, the CAM lets the network focus on the foreground of the image and the significant region. These two modules can be used together. The CBAM begins with the channel and spatial range. To realize the sequential attention structure from channel to space, two analytical dimensions (spatial attention and channel attention) are added [22]. Instead of disregarding the irrelevant region, the neural network may be trained to pay greater attention to the pixel region in the image that defines focus segmentation using a spatial attention module (SAM). The channel attention module (CAM) is used to handle the allocation relationship of the feature mapping channels. At the same time, allocating attention to two dimensions enhances the improvement in the attention mechanism and its effect on model performance. The structure of the CBAM is shown in Figure 2.

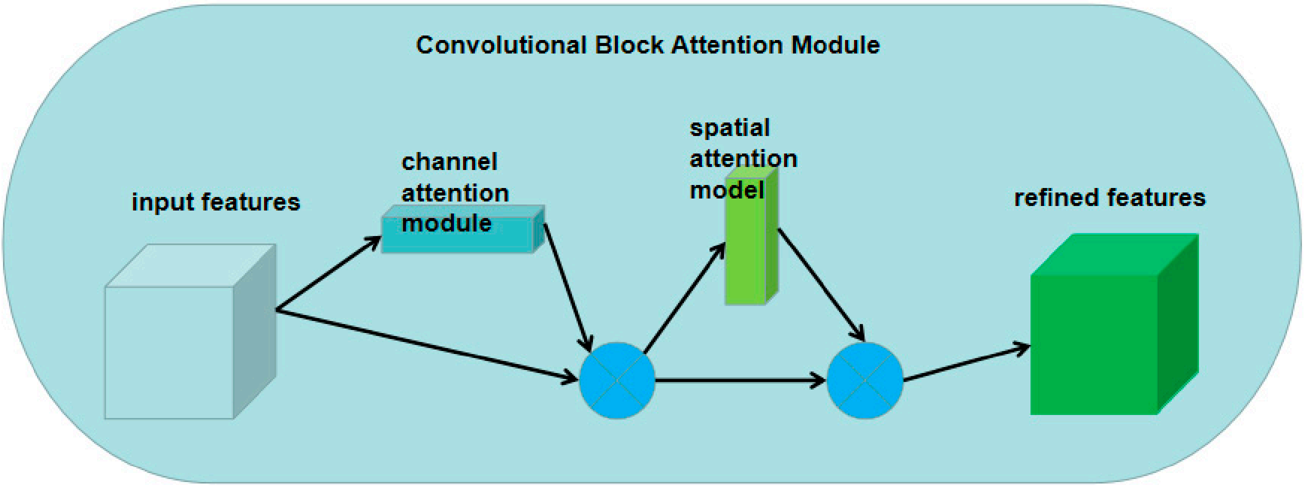

Figure 2. Overview of the convolutional block attention module.

The specific CAM procedure is as follows: We input the feature map F through the global maximum pool and the global average pool in accordance with the width and height to generate two 1 × 1 × C characteristic diagrams and then send them separately. We enter a shared two-layer neural network (MLP). The last channel attention feature, M_c, is created by adding MLP output characteristics in element order and activating them with Sigmoid. The input features needed by the spatial attention module are produced by multiplying M_c and the input feature map F according to the element direction. This can be expressed by the following calculation, Formula (1).

$$M_c(F) = \sigma\left(W_1\left(W_0\left(F^c_{avg}\right)\right) + W_1\left(W_0\left(F^c_{avg}\right)\right)\right). \tag{1}$$

The specific SAM procedure is as follows: We use the channel attention module's output feature map F as this module's input feature map. We create a global maximum pool and global average pool based on the channels, obtain two H × W × 1 characteristic graphs, and then conduct channel-based concatenation using these two characteristic graphs. To reduce the number of channels, we then utilize the 7 × 7 convolution procedure. After sigmoid activation, the spatial attention characteristic M_s is formed. Lastly is M_s. To obtain the final feature, we multiply the feature of S by the module. This can be expressed as the following calculation, Formula (2).

$$M_s(F) = \sigma\left(f^{7\times7}\left(\left[F^c_{avg}; F^s_{max}\right]\right)\right). \tag{2}$$

In the decoding step, the low-level features produced in the shallow network are utilized directly as input data and several background features are added, which has an impact on the segmentation outcomes. The channel focus mechanism module has more weight and becomes more responsive to target objects through addition of the convolutional block attention module (CBAM). The spatial attention mechanism is more attentive to the foreground region and the characteristics of the target region, which contributes to the generation of more efficient feature maps.

2.4. Importing the Attention Mechanism SENet

The Squeeze-and-Excitation Network (SENet) module's construction is depicted in Figure 3 [23]. Its purpose is to allow the network to undertake dynamic channel feature recalibration in order to enhance network characterization capabilities. In other words, it employs learning to automatically determine the relevance of each characteristic, enhancing the qualities that are applicable to the task while suppressing the ones that are not. It mostly consists of the following three sections.

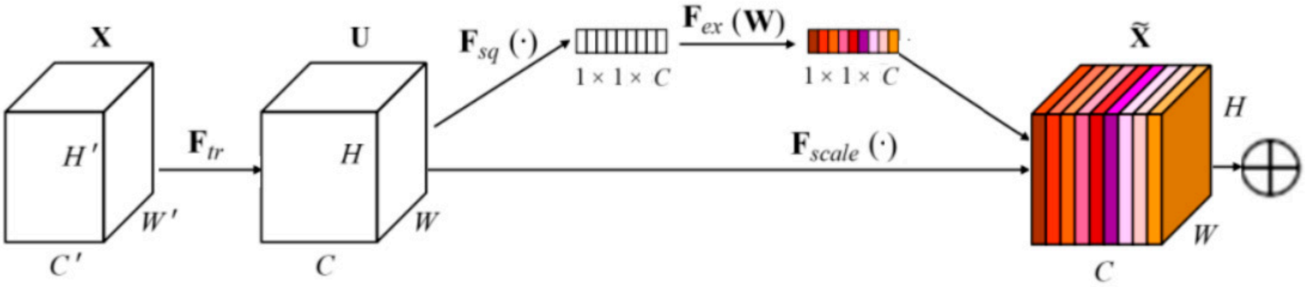

Figure 3. The structure of the SENet.

In the compression operation, after obtaining U (multiple feature maps), each feature map is compressed by the global average pool; thus, the C feature map becomes a 1 × 1 × C real number sequence [23].

In the excitation operation, nonlinear transformation of the extruded results is performed by using a fully connected neural network [23].

In the weighting operation, we use the result of the excitation as the weight and multiply it by the input characteristic [23]. The mapping relationship is shown in Formula (3)–(5).

$$z_c = F_{sq}(u_c) = \frac{1}{H\times W}\sum_{i=1}^{H}\sum_{j=1}^{W} u_c(i,j), \tag{3}$$

$$s = F_{ex}(z,W) = \sigma(W_2\delta(W_1 z)), \tag{4}$$

$$\tilde{x}_c = F_{scale}(u_c, s_c) = s_c u_c, \tag{5}$$

where u_c represents each characteristic channel; W and H represent the width and height of uc, respectively; z_c represents the compressed value of the c-dimension channel; W_1 and W_2 represent the dimension and increment, respectively, which are the weights of the full join operation; δ is a ReLU function, and σ is an s-type function. The function δ represents the first full connection layer, and σ is the second full connection layer. Finally, by multiplying the scalar s_c and the feature u_c, representing the c dimension obtains the final characteristics, $\tilde{x}_c$.

2.5. Importing the Weighted Loss Function

Loss is a function that describes the difference between the output value of the model and the true value of the sample, which is derived from the input dataset [24]. The neural network weights in the deep learning model are taught via loss backpropagation. The training impact of the depth learning model is thus crucially determined by the loss function. Simple and complicated backgrounds are the two categories used in this study to categorize tree labeling. As a result, the loss function is multi-category cross entropy [24]. The network tended to learn features with complex backgrounds during training and was unable to successfully extract features with simple backgrounds, leading to low segmentation accuracy with simple backgrounds. This was due to the large proportion of categories in the dataset that had complex backgrounds. A weighted loss function was provided to address the issue of the imbalanced segmentation's low accuracy.

2.6. Detectron2

Yolo v7's mask branch depends on Facebook's detectron2, and detectron2 needs a version of Torch that is higher than 1.8. In order to recreate detectron, the most popular deep learning framework in 2020, Facebook AI Research (FAIR) developed detectron2 [25].

Additionally, simpledet and mmdetection were also used [25]. With a large number of pretrained models and the integration of sophisticated target recognition and semantic segmentation algorithms, detectron2 N2 was rebuilt from a Caffe-based to a PyTorch-based architecture. Plugging it in and starting it up is fairly easy.

2.7. Overall Process

The training portion and the application portion of the segmentation stages for the standing trees are shown in Figure 4. In order to create a collection of training data for standing tree images, the camera's upright tree image was first utilized, increased by data augmentation, and then labeled. The semantic division network was then fed the learned dataset. The segmentation model was created after training, and the training ended when the loss value approached convergence. In a practical application, the training model was fed an image of the target standing tree to produce the segmentation effect.

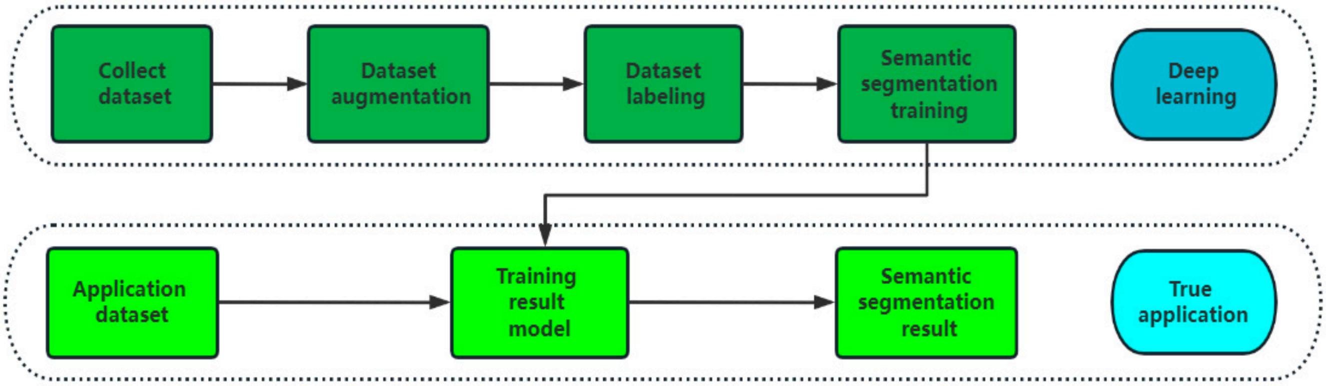

Figure 4. Overall flow chart of the tree semantic segmentation process.

3. Experiments

The outcomes of the suggested Yolo v7 model were tested and validated, as described in this section. The standing tree image collection used for this experiment, the network hyperparameter settings, hardware and software setups, and the experimental methodology are presented.

3.1. Experimental Software and Hardware Configuration

The deep learning framework PyTorch was used to implement the Yolo v7 model and the experiments. The hardware and software configurations are shown in Table 1.

Table 1. Experimental software and hardware configuration.

Project	Detail
CPU	E5 2678v3 × 2
GPU	GTX 3060Ti 8G
RAM	16 GB × 2
Disk	HITACHI A640 3T
OS	Windows 11 Pro
Anaconda	Anaconda3-2022.10-Windows 64
PyTorch	PyTorch 1.8.0
Python	Python 3.8
Cuda	Cuda 11.1
PyCharm	PyCharm Community 2 February 2021
Labelme	Labelme 1.5.1

3.2. Production of Dataset

3.2.1. Image Data Acquisition

The experiment's tree images were derived from two different sources of environmental data. Lin'an City was the first location. The sampling region is situated on the southern boundary of the mid-subtropical monsoon climatic zone in the northwest of Zhejiang Province. There are four distinct seasons and a monsoon climate, which is warm and humid with adequate light and plentiful rainfall. Hills and mountains dominate the landscape, which slopes from the northwest to the southeast and has a pronounced three-dimensional climate [26]. The second location was Lishui City. This sampling region, which is in the heart of the subtropical monsoon climatic zone and has clear subtropical maritime monsoon climate features, is situated at the intersection of the provinces of Zhejiang and Fujian in Zhejiang Province's southwest. With four different seasons, a mild winter and early spring, copious precipitation, synchronous rain and heat, and a variety of vertical climates, the landscape is primarily hilly and mountainous [27]. The backdrop settings of the trees fell into two categories: simple and complicated background environments.

The backdrop surroundings, natural lighting, photography tools, etc., were also thought to have an impact on the images of maple trees. The Redmi K30 Pro's 64 million onboard cameras were used in this experiment, and the image collection period spanned from June to October 2022. These images were randomly captured and gathered during the day. Data collection also included images taken in various weather situations, which could more accurately depict actual observations. In the end, 500 screened images were created, including 100 simple background images of Lin'an, 100 simple background images of Lishui, 150 complex background images of Lin'an, and 150 complex background images of Lishui.

3.2.2. Image Data Preprocessing

Numerous data samples were gathered, most of which were images that were nearly 4K in resolution. Each image was uniformly enlarged to 1280 × 1280 pixels in order to evaluate the performance of Yolo v7 with other comparable segmentation algorithms under the same circumstances and increase the effectiveness of the segmentation trials. Additionally, the pixels in Yolo v7 must be multiples of 32.

3.2.3. Data Enhancement Processing

Because it is easy to overfit the model by directly using the original data for training, the experimental dataset was enhanced and processed. The CVPR Fine Visualization Classification Challenge used a data enhancement operation to enlarge the original data. Using this method, the generalization ability of the model can be enhanced, and the image translation and flipping invariance can be given [28]. The Albumations data enhancement library was used to randomly enhance the brightness, cropping, flipping, and shifting of the marked images, respectively, enlarging them by 2, 4, 2, and 2 times, totaling 24 times. We then processed the images to a resolution of 1280 × 1280 pixels, for a total of 12,000 experimental data images [29].

3.2.4. Data Annotation Processing

An open-source program called LabelMe 1.5.1 was used to annotate the images in the datasets. The Massachusetts Institute of Technology's (MIT) Computer Science and Artificial Intelligence Laboratory (CSAIL) created the image labeling program labelme. This program can be used to perform image tagging or to construct custom labels. The project's source code is available for download [30]. Labelme is a program for graphically marking images that can mark points, line segments, rectangles, polygons, and other shapes. Annotation data are saved in the json format. However, Yolo does not support json files; thus, json code must be used to convert the data of the text. In the figures, the backdrop is black and the segmentation of the standing tree is in red.

3.3. Experimental Design

3.3.1. Classification Settings of Datasets

Standing tree images with simple and complicated backgrounds were randomly allocated according to functions based on the generation of the aforementioned datasets, with 70% serving as the training set, 20% serving as the validation dataset, and 10% serving as the testing dataset. The experimental data needed to be categorized in accordance with the 7:2:1 general division rule. There were four categories: the simple background of Lin'an, the simple background of Lishui, the complicated background of Lin'an, and the complicated background of Lishui.

3.3.2. Experimental Effect Evaluation Index

Image segmentation includes semantic segmentation, instance segmentation, and panoramic segmentation. Their evaluation indexes are basically the same, and these indexes are inseparable from the basic confusion matrix [31]. The commonly used image segmentation indicators are pixel accuracy (PA), class pixel accuracy (CPA), class average pixel accuracy (MPA), intersection over union (IoU), and mean intersection over union (MIoU) [32]. In this study, MPA and MIoU were used as the evaluation indexes. MPA calculates the proportion of the correctly classified pixels of each class separately and calculates the average by accumulation. MIoU divides the intersection of the predicted area and the actual area by the union of the predicted area and the actual area, so that the IoU under a single category can be calculated. We then repeat this algorithm to calculate the IoUs of other classes and calculate their average. Their formulas are shown below.

$$\boldsymbol{MPA} = \frac{1}{k+1}\sum_{i=0}^{k}\frac{p_{ii}}{\sum_{j=0}^{k}p_{ij}} \tag{6}$$

$$\boldsymbol{MIoU} = \frac{1}{k+1}\sum_{i=0}^{k}\frac{p_{ii}}{\sum_{j=0}^{k}p_{ij}+\sum_{j=0}^{k}p_{ji}-p_{ii}} \tag{7}$$

3.3.3. Experimental Scheme Design

For the experimental environment, refer to 3.1. In order to improve the accuracy of the model, the weighted loss function was used as the loss function for the segmentation model, and a new attention module was also introduced to improve the segmentation accuracy of the model with opposite trees. We input the training dataset with labeling information into the improved Yolo v7 mask network for training. For the dataset, 70% of the images were randomly selected for the training set, 20% of the images for the verification set, and 10% of the images for the testing set. Because too low or too high a learning rate will lead to slow or even no convergence of the model, it was necessary to determine the appropriate initial learning rate [33]. In this paper, the accuracy of the five models designed and tested under the initial learning rate in the training was evaluated, and eight classified data were combined. The results are shown in Figures 5 and 6. It can be seen that the model had the highest accuracy when the learning rate was 0.0002, and the epoch was 200. See Table 2 for the training hyperparameters of the Yolo v7 model.

Table 2. Yolo v7 hyperparameters.

Project	Detail
Epoch	300
Batch size	8
Input shape	1280 × 1280

We chose images of standing trees and trained the model using the aforementioned settings. The validation data, which were each 1280 × 1280 pixels in size, were predicted using the model and the loss rate of the stored results from the training phase, which were then superimposed over the original image. The outcomes are displayed in Figure 7.

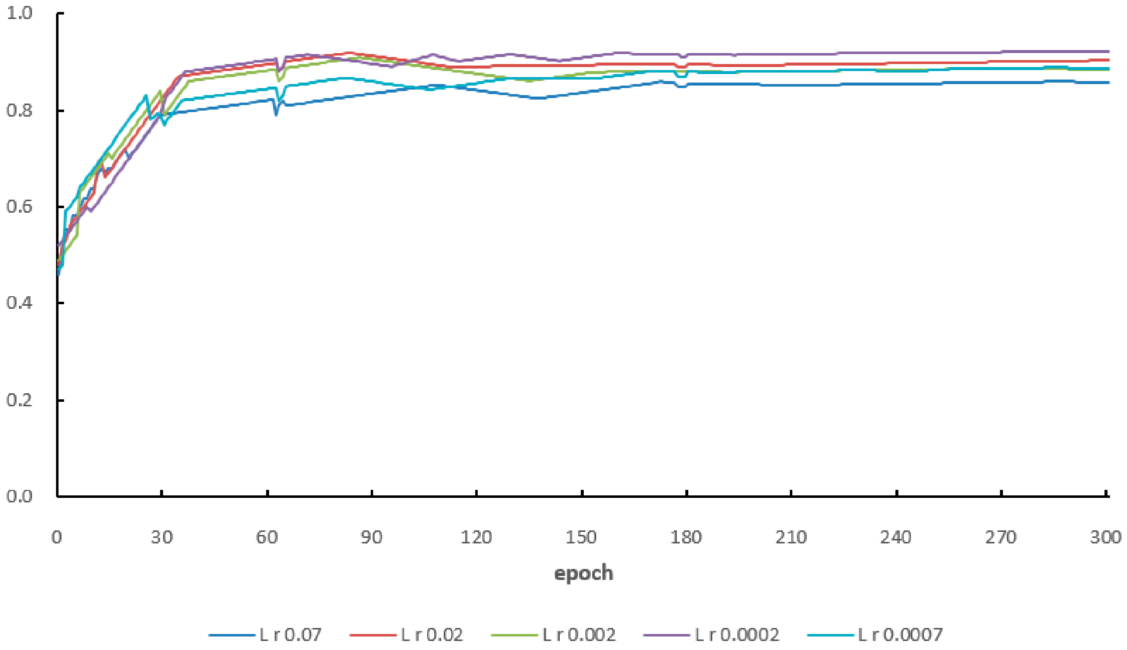

Figure 5. The relationship between model accuracy, learning rate, and iteration times.

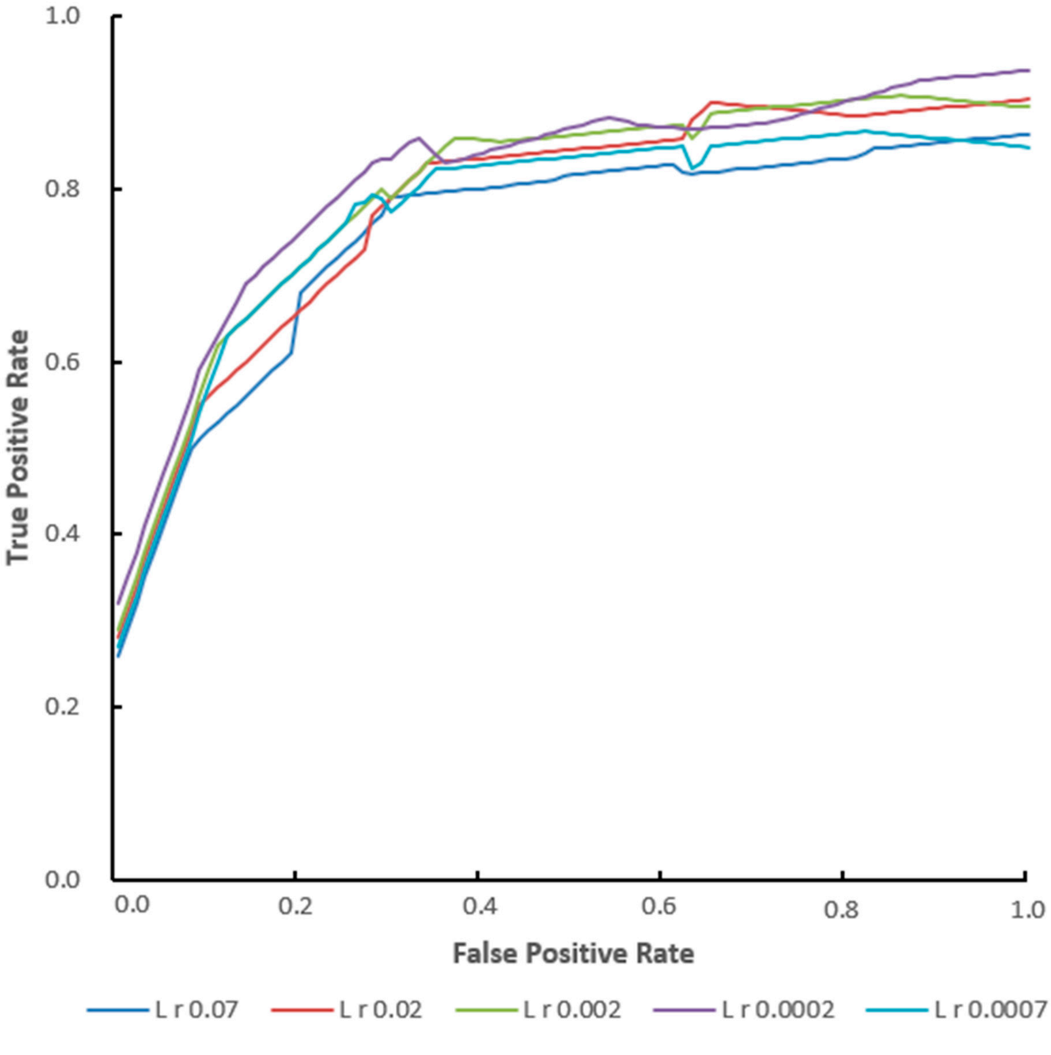

Figure 6. The Receiver Operating Characteristic (ROC).

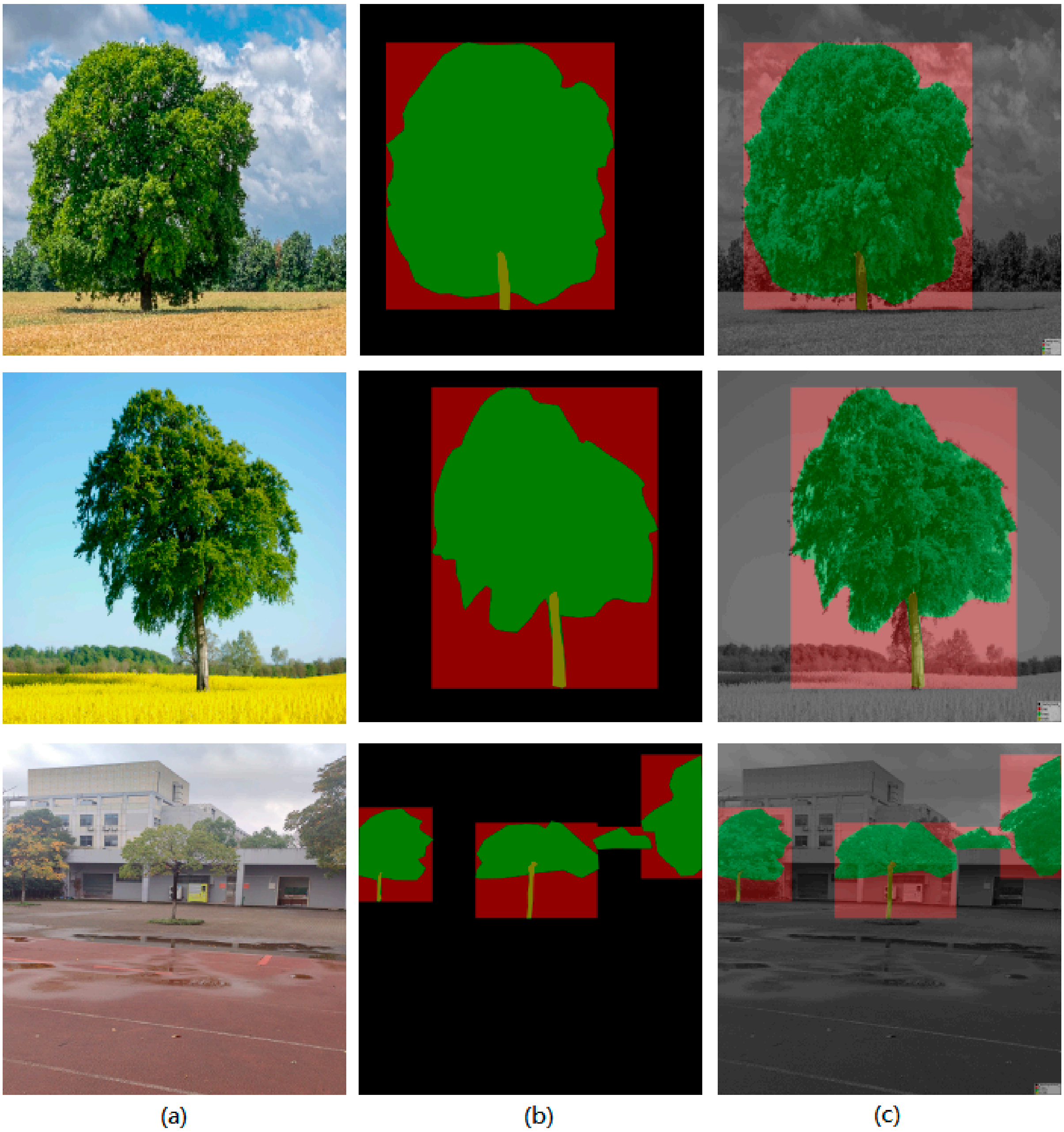

Figure 7. Overlay of the segmentation results. (**a**) Original image. (**b**) Segmentation process. (**c**) Mask overlay.

4. Analysis and Discussion

4.1. Visual Analysis

4.1.1. The Effect of Tree Segmentation with a Simple Background

When compared to other network models for segmenting standing trees on a plain background, FCN segmentation was comparatively subpar [34] and could only approximately segment the canopy's form. The relay network was not correctly divided by the SegNet network [35]. U-segmentation Net's findings were more accurate than SegNet and FCN; however, faulty trunk segmentation results still existed [35]. In every way, DEEBV3+ was a better segmentation algorithm than the prior model, although trunk gap segmentation was incorrect [36]. This issue was well solved by the Yolo v7 model. Figure 8 displays the outcomes of utilizing several techniques to separate standing trees with a simple background.

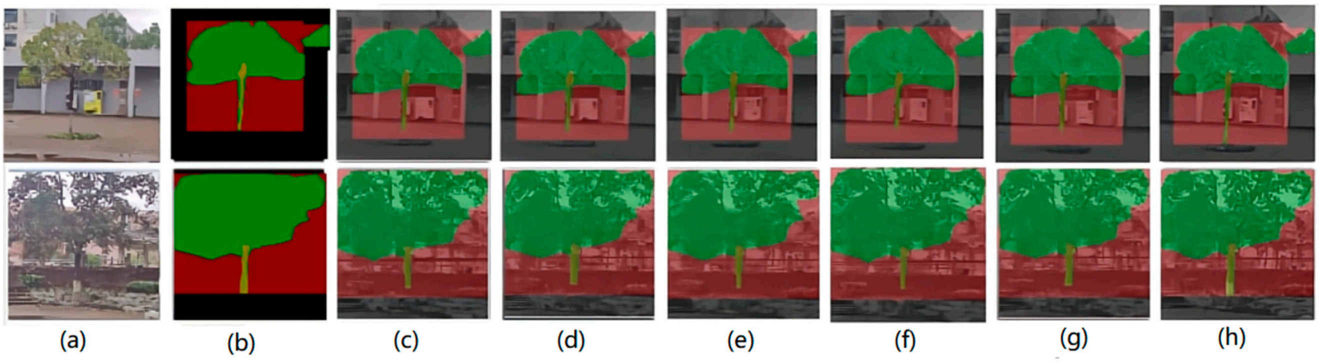

Figure 8. Comparison of standing tree segmentation with simple backgrounds. (**a**) Original image. (**b**) Ground truth. (**c**) FCN. (**d**) SegNet. (**e**) U-Net. (**f**) PSPNet. (**g**) DeepLabV3. (**h**) Yolo v7.

4.1.2. The Effect of Tree Segmentation with a Complex Background

The standing tree image segmentation experiment had a complicated background with a variety of buildings, greenery, and other noise or human activity traces that made segmentation more challenging.

Yolo v7 enhanced the detail processing of the trunk edges, minimized the acquisition of erroneous features, and boosted the accuracy of semantic segmentation when compared to the other five segmentation techniques. It also incorporated the weighted loss function. The entire standing tree's segmentation impact was definitely improved, which helped subsequent factor computation, minimized manual intervention, and resolved the issues of inefficient operation and erroneous segmentation.

4.2. Comparison of the Segmentation Indexes of Different Models

4.2.1. Comparison of the Different Models with a Simple Background

On the standing tree dataset, the Yolo v7 model was compared to the other five methods (FCN, SegNet, U-Net, PSPNet, and the DeepLabV3+) according to the assessment indicators provided in Section 3.3.2. The findings are displayed in Table 3.

Table 3. Performance comparison with a simple background.

Model	Category	MPA (%)	MIoU (%)
FCN	Lin'an sample	85.36	75.32
	Lishui sample	86.72	76.27
SegNet	Lin'an sample	86.29	76.39
	Lishui sample	85.96	74.18
U-Net	Lin'an sample	87.82	77.60
	Lishui sample	86.33	77.13
PSPNet	Lin'an sample	83.56	72.69
	Lishui sample	84.73	73.98
DeepLabV3+	Lin'an sample	91.57	84.15
	Lishui sample	91.85	83.62
Yolo v7	Lin'an sample	95.87	92.12
	Lishui sample	94.69	91.17

Yolo v7's MPA was higher than that of other techniques when compared to the typical mainstream semantic segmentation algorithm [37] and the other five algorithms in the presence of a simple background. With a simple background, accuracy of 95.87% and 94.69% was achieved with the samples of Lin'an and Lishui, respectively, while the other evaluation index (MioU) demonstrated 92.12% and 91.17% accuracy, respectively.

4.2.2. Comparison of the Different Models with a Complex Background

Compared to a simple background, the performance of tree segmentation with a complex background was slightly lower, but the Yolo v7 model still achieved better segmentation results with a complex background, as shown in Table 4. Using the improved Yolo v7 model, the samples of Lin'an City and Lishui City demonstrated MPA of 94.27% and 93.46%, respectively, and the other evaluation index (MioU) was 91.28% and 90.23%, respectively.

Table 4. Performance comparison with a complex background.

Model	Category	MPA (%)	MIoU (%)
FCN	Lin'an complex	83.21	72.67
	Lishui complex	84.16	74.38
SegNet	Lin'an complex	84.65	75.23
	Lishui complex	85.63	75.18
U-Net	Lin'an complex	84.25	73.64
	Lishui complex	83.18	71.59
PSPNet	Lin'an complex	82.62	72.66
	Lishui complex	83.67	72.97
DeepLabV3+	Lin'an complex	90.54	83.62
	Lishui complex	91.22	84.58
Yolo v7	Lin'an complex	94.27	91.28
	Lishui complex	93.46	90.23

We chose five random samples with a simple background. See Table 5 for the PSNR (Peak Signal-to-Noise Ratio) values of the samples of standing trees under various models. Yolo v7 had better performance in terms of segmentation results, almost 2 dB higher than the other models on average, which was about 1 dB higher than the best one. This proves that the model has high feasibility in the segmentation of standing trees.

Table 5. PSNR (dB) comparison of different models.

Model	1	2	3	4	5
FCN	15.86	15.21	15.67	16.75	15.84
SegNet	15.92	15.34	16.21	16.32	16.38
U-Net	17.26	18.02	18.33	18.66	17.94
PSPNet	15.93	16.35	16.37	17.53	17.14
DeepLabV3+	19.34	18.63	19.57	20.04	19.32
Yolo v7	20.21	19.83	21.04	21.36	21.16

4.3. Ablation Experiments

An ablation experiment is an important method that can evaluate the quality of a model. In this study, an ablation experiment was carried out using the dataset of standing trees in Lishui City [38]. In order to verify the effectiveness of introducing the attention and weighted loss function, four groups of different cases were established for the experiments, and the experimental results are shown in Table 6. Yolo v7+O. refers to the original Yolo v7 mask branch model without the convolutional block attention module, Yolo v7+L. refers to the branch model of the Yolo v7 mask with the weighted loss function, and Yolo v7+S. refers to the branch model of the Yolo v7 mask with the convolutional block attention module. It can be seen from the table that the improved Yolo v7 model not only improved segmentation accuracy, but also shortened the running time of prediction and optimized the size of the model's results and the spatial complexity of the model.

Table 6. Comparison results of the different model architectures.

Model	MPA (%)	Time (s)	Size (Mb)
Yolo v7 + O.	87.34	0.13	74.9
Yolo v7 + L.	90.17	0.12	73.6
Yolo v7 + S.	91.51	0.13	73.8
Yolo v7	93.75	0.12	73.3

4.4. Computational Complexity

Time complexity and spatial complexity are both components of computational complexity. People often focus more on time complexity and less on space complexity. In this part, we compare the training and reasoning times of the proposed Yolo v7 model and the other five models [39]. Based on the measured values of the hardware infrastructure mentioned in Section 3.1, Table 7 displays the training and reasoning times.

Table 7. Average processing time for each method.

Model	Training Time (h)	Inference Time (s)
FCN	24	0.52
SegNet	22	0.59
U-Net	19	0.34
PSPNet	26	0.51
DeepLabV3+	23	0.47
Yolo v7	18	0.12

Both results heavily depended on the network's depth and the size of the chosen number, as these approaches were trained using the same optimizer and learning rate. For instance, SegNet was quite fast at training and reasoning; however, it performed poorly when measured against the assessment metrics. U-Net also had a faster training time due to its basic network topology and limited number of parameters; however, the outcome was somewhat poorer than Yolo v7. Because DeepLabv3+ was more complex than the previous networks, training and reasoning took longer. Yolo v7's prediction phase was substantially faster than the previous models' due to the use of model scaling and other functions, and the improvement phase added very little to time complexity.

4.5. Robustness Test

Experiments with multiple input image sizes were used to evaluate the performance of the Yolo v7 model with varying input sizes; the same settings for the other parameters were retained in order to assess the robustness of the model. Table 8 presents the outcomes. The table indicates that the Yolo v7 model performed similarly over a range of input sizes, demonstrating its high resilience.

Table 8. Comparison results of the different input shapes.

Input Shape	MIoU (%)	Speed (s)
1280 × 1280	91.17	0.11
640 × 640	91.08	0.10
320 × 320	89.87	0.08

5. Conclusions

To address the semantic segmentation of standing tree images with simple and complicated backdrops, a better semantic segmentation approach based on deep learning and the most recent version of the well-known Yolo algorithm was suggested. To increase segmentation accuracy and robustness, decrease the acquisition of erroneous feature information, and widen the acceptance domain, the Yolo v7 model was coupled with the attention mechanism module SENet. A weighted loss function was developed to address the issue of category imbalance with complicated backgrounds and enhance the segmentation accuracy of the model. The enhanced Yolo v7 mask standing tree segmentation model was evaluated and contrasted with other approaches using the standing tree image dataset. According to the experimental findings, detection speed was substantially faster than that of other methods. This technique also successfully segmented the tree at the same time. The MPA of the Yolo v7 Model was improved to 94.69% and 93.46% with simple and complicated backgrounds, respectively, while the MIoU of the Yolo v7 Model was improved to 91.17% and 90.23%, respectively.

Yolo v7, which was proposed in this paper, performed poorly in environments with high background complexity, and the semantic segmentation effect was even worse when there was insufficient light or more occlusion. This is true even though it did achieve a better segmentation effect on standing trees in complex backgrounds.

Author Contributions: Conceptualization, L.C.; methodology, X.Z.; software, X.Z.; formal analysis, L.C.; investigation, L.C.; resources, L.F.; data curation, L.C.; writing—original draft preparation, L.C.; writing—review and editing, L.F.; project administration, L.C.; funding acquisition, L.F. All authors have read and agreed to the published version of the manuscript.

Funding: This work was financially supported by the Zhejiang Provincial Key Science and Technology Project, Grant No. 2018C02013; the National Natural Science Foundation of China, Grant No. 42001354; and the Natural Science Foundation of Zhejiang Province, Grant No. LQ19D010011.

Data Availability Statement: Unfortunately, the data is not available due to the supervision of the local government and the competent department of State Forestry.

Acknowledgments: We appreciate Xiaoming Lai (Longquan Forestry Bureau of Zhejiang Province, Lishui 323000, China) for contributions to validation, visualization, and supervision.

Conflicts of Interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Semeraro, T.; Gatto, E.; Buccolieri, R.; Catanzaro, V.; De Bellis, L.; Cotrozzi, L.; Lorenzini, G.; Vergine, M.; Luvisi, A. How Ecosystem Services Can Strengthen the Regeneration Policies for Monumental Olive Groves Destroyed by Xylella fastidiosa Bacterium in a Peri-Urban Area. *Sustainability* **2021**, *13*, 8778. [CrossRef]
2. Dechesne, C.; Mallet, C.; Le Bris, A.; Gouet-Brunet, V. Semantic segmentation of forest stands of pure species combining airborne lidar data and very high resolution multispectral imagery. *ISPRS J. Photogramm. Remote Sens.* **2017**, *126*, 129–145. [CrossRef]
3. Cong, P.; Zhou, J.; Li, S.; Lv, K.; Feng, H. Citrus Tree Crown Segmentation of Orchard Spraying Robot Based on RGB-D Image and Improved Mask R-CNN. *Appl. Sci.* **2023**, *13*, 164. [CrossRef]
4. Pyo, J.; Han, K.-J.; Cho, Y.; Kim, D.; Jin, D. Generalization of U-Net Semantic Segmentation for Forest Change Detection in South Korea Using Airborne Imagery. *Forests* **2022**, *13*, 2170. [CrossRef]
5. Marsocci, V.; Scardapane, S.; Komodakis, N. MARE: Self-Supervised Multi-Attention REsu-Net for Semantic Segmentation in Remote Sensing. *Remote Sens.* **2021**, *13*, 3275. [CrossRef]
6. Cao, J.; Song, C.; Song, S.; Xiao, F.; Zhang, X.; Liu, Z.; Ang, M.H., Jr. Robust Object Tracking Algorithm for Autonomous Vehicles in Complex Scenes. *Remote Sens.* **2021**, *13*, 3234. [CrossRef]
7. Li, M.; Li, Z.; Li, L.; Song, W. Yolo-Based Traffic Sign Recognition Algorithm. *Comput. Intell. Neurosci.* **2022**, *2022*, 2682921. [CrossRef] [PubMed]
8. Quoc, H.N.; Hoang, V.T. Real-Time Human Ear Detection Based on the Joint of Yolo and RetinaFace. *Complexity* **2021**, *2021*, 7918165.
9. Qi, L.; Gao, J. Small target detection based on improved Yolo v7. *Comput. Eng.* **2023**, *49*, 41–48.
10. Kim, K.; Jung, S.-W. Interactive Image Segmentation Using Semi-transparent Wearable Glasses. *IEEE Trans. Multimed.* **2017**, *20*, 208–223. [CrossRef]
11. Hu, T.; Yang, M.; Yang, W.; Li, A. An end-to-end differential network learning method for semantic segmentation. *Int. J. Mach. Learn. Cybern.* **2019**, *10*, 1909–1924. [CrossRef]
12. Wang, Z.; Gao, X.; Wu, R.; Kang, J.; Zhang, Y. Fully automatic image segmentation based on FCN and graph cuts. *Multimed. Syst.* **2022**, *28*, 1753–1765. [CrossRef]
13. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.

14. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.
15. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
16. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolo v4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
17. Lin, G.; Liu, K.; Xia, X.; Yan, R. An Efficient and Intelligent Detection Method for Fabric Defects Based on Improved YOLO v5. *Sensors* **2023**, *23*, 97. [CrossRef] [PubMed]
18. Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W.; et al. YOLO v6: A single-stage object detection framework for industrial applications. *arXiv* **2022**, arXiv:2209.02976.
19. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLO v7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv* **2022**, arXiv:2207.02696.
20. Chen, W.; Han, G.; Zhu, H.; Liao, L.; Zhao, W. Deep ResNet-Based Ensemble Model for Short-Term Load Forecasting in Protection System of Smart Grid. *Sustainability* **2022**, *14*, 16894. [CrossRef]
21. Du, W.; Xiang, Z.; Chen, S.; Qiao, C.; Chen, Y.; Bai, T. Real-time instance segmentation with discriminative orientation maps. In Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 7314–7323.
22. Wang, Y.; Li, J.; Chen, Z.; Wang, C. Ships' Small Target Detection Based on the CBAM-YOLOX Algorithm. *J. Mar. Sci. Eng.* **2022**, *10*, 2013. [CrossRef]
23. Yang, L.; Yan, J.; Li, H.; Cao, X.; Ge, B.; Qi, Z.; Yan, X. Real-Time Classification of Invasive Plant Seeds Based on Improved YOLOv5 with Attention Mechanism. *Diversity* **2022**, *14*, 254. [CrossRef]
24. Rengasamy, D.; Jafari, M.; Rothwell, B.; Chen, X.; Figueredo, G.P. Deep Learning with Dynamically Weighted Loss Function for Sensor-Based Prognostics and Health Management. *Sensors* **2020**, *20*, 723. [CrossRef] [PubMed]
25. Wu, Y.; Kirillov, A.; Massa, F.; Lo, W.Y.; Girshick, R. Detectron2. 2019. Available online: https://github.com/facebookresearch/detectron2 (accessed on 8 September 2022).
26. Hangzhou Lin'an District People's Government. Lin'an Geogr. 2022. Available online: http://www.linan.gov.cn/art/2022/3/1/art_1366301_11082111.html (accessed on 7 April 2022).
27. Lishui Municipal Party History Research Office, Lishui Local Chronicles Research Office. Physical Geography.2022. Available online: http://lssz.lishui.gov.cn/art/2022/5/16/art_1229634360_7027.html (accessed on 7 April 2022).
28. Liu, Z.; Luo, P.; Wang, X.; Tang, X. Deep learning face attributes in the wild. In Proceedings of the IEEE International Conference on Computer Vision, 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 3730–3738.
29. Tian, Y.; Chen, Y.; Diming, W.; Shaoguang, Y.; Wandeng, M.; Chao, W.; Xu, C.; Long, Y. Augmentation Method for anti-vibration hammer on power transimission line based on CycleGAN. *International Journal of Image and Data Fusion* **2022**, *13*, 362–381. [CrossRef]
30. Nath, V.; Yang, D.; Landman, B.A.; Xu, D.; Roth, H.R. Diminishing Uncertainty Within the Training Pool: Active Learning for Medical Image Segmentation. *IEEE Trans. Med. Imaging* **2020**, *40*, 2534–2547. [CrossRef] [PubMed]
31. Xu, J.; Zhang, Y.; Miao, D. Three-way confusion matrix for classification: A measure driven view. *Inf. Sci.* **2020**, *507*, 772–794. [CrossRef]
32. Unnikrishnan, R.; Pantofaru, C.; Hebert, M. A measure for objective evaluation of image segmentation algorithms. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)-Workshops, San Diego, CA, USA, 21–23 September 2005; p. 34.
33. Zeiler, M.D. Adadelta: An adaptive learning rate method. *arXiv* **2012**, arXiv:1212.5701.
34. Lu, Y.; Chen, Y.; Zhao, D.; Chen, J. Graph-FCN for image semantic segmentation. In *Advances in Neural Networks, Proceedings of the ISNN 2019: 16th International Symposium on Neural Networks, ISNN 2019, Moscow, Russia, 10–12 July 2019*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 97–105.
35. Atika, L.; Nurmaini, S.; Partan, R.U.; Sukandi, E. Image Segmentation for Mitral Regurgitation with Convolutional Neural Network Based on UNet, Resnet, Vnet, FractalNet and SegNet: A Preliminary Study. *Big Data Cogn. Comput.* **2022**, *6*, 141. [CrossRef]
36. De Andrade, R.B.; Mota, G.L.A.; da Costa, G.A.O.P. Deforestation Detection in the Amazon Using DeepLabv3+ Semantic Segmentation Model Variants. *Remote Sens.* **2022**, *14*, 4694. [CrossRef]
37. Zhao, F.; Xie, X. An overview of interactive medical image segmentation. *Ann. BMVA* **2013**, *7*, 1–22.
38. Zhou, B.; Sun, Y.; Bau, D.; Torralba, A. Revisiting the importance of individual units in cnns via ablation. *arXiv* **2018**, arXiv:1806.02891.
39. Goldreich, O. Computational complexity: A conceptual perspective. *ACM Sigact News* **2018**, *39*, 35–39. [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.