A Deep Learning-Based Method for Extracting Standing Wood Feature Parameters from Terrestrial Laser Scanning Point Clouds of Artificially Planted Forest

Xingyu Shen , **Qingqing Huang ***, **Xin Wang, Jiang Li and Benye Xi**

School of Technology, Beijing Forestry University, No. 35 Qinghua East Road, Haidian District, Beijing 100083, China
* Correspondence: huangqingqing@bjfu.edu.cn

Abstract: The use of 3D point cloud-based technology for quantifying standing wood and stand parameters can play a key role in forestry ecological benefit assessment and standing tree cultivation and utilization. With the advance of 3D information acquisition techniques, such as light detection and ranging (LiDAR) scanning, the stand information of trees in large areas and complex terrain can be obtained more efficiently. However, due to the diversity of the forest floor, the morphological diversity of the trees, and the fact that forestry is often planted as large-scale plantations, efficiently segmenting the point cloud of artificially planted forests and extracting standing wood feature parameters remains a considerable challenge. An effective method based on energy segmentation and PointCNN is proposed in this work to address this issue. The network is enhanced for learning point cloud features by geometric feature balance model (GFBM), enabling the efficient segmentation of tree point clouds from forestry point cloud data collected by terrestrial laser scanning (TLS) in outdoor environments. The 3D Forest software is then used to obtain single wood point cloud after semantic segmentation, and the extracted single wood point cloud is finally employed to extract standing wood feature parameters using TreeQSM. The point cloud semantic segmentation method is the most important part of our research. According to our findings, this method can segment datasets of two different artificially planted woodland point clouds with an overall accuracy of 0.95 and a tree segmentation accuracy of 0.93. When compared with the manual measurements, the root-mean-square error (RMSE) for tree height in the two datasets are 0.30272 and 0.21015 m, and the RMSEs for the diameter at breast height are 0.01436 and 0.01222 m, respectively. Our method is a robust framework based on deep learning that is applicable to forestry for extracting the feature parameters of artificially planted trees. It solves the problem of segmenting tree point clouds in artificially planted trees and provides a reliable data processing method for tree information extraction, trunk shape analysis, etc.

Keywords: deep learning; semantic segmentation; extraction of tree feature information; TLS LiDAR data; artificially planted forest

1. Introduction

The standing wood characteristics of trees provide important three-dimensional data [1] that can be extracted to obtain detailed information, such as a tree's position, height, wood volume, and diameter at breast height [2]. While the information on standing characteristics is important for forest resource management [3], field inventories [4], and artificial afforestation, it can also assist in the research of tree animal habitats and their habitat structures [5] and in urban gardens for landscape design [6]. Traditional methods to obtain tree information generally require manual field measurements, and there are many tools and methods to measure forestry information directly [7]. However, this process is highly time-consuming and may cause some damage to the trees. The development

of modern remote sensing techniques, particularly light detection and ranging (LiDAR) sensor-based simultaneous localization and mapping (SLAM) [8–11], has made the exploration of imaging is gradually increasing [12,13] and has made it possible for technicians without considerable training to easily collect high-quality 3D information on forestry and reconstruct forestry point cloud maps. The laser scanning systems commonly used to collect forestry information can be divided into the following categories depending on the carrier platform: terrestrial laser scanning (including terrestrial laser, backpack laser, and vehicle-borne laser), satellite lidar scanning, and airborne laser scanning. Among them, terrestrial laser scanning systems are widely used in forest remote sensing because of their high flexibility and portability and good point cloud quality [14–18]. The datasets we collected in this paper are based on terrestrial laser scanning. While the extraction of forestry 3D information has become increasingly rich and high quality, its complexity creates processing challenges.

Deep learning is currently one of the most widely researched areas of machine learning, with applications in object part segmentation, natural language processing, target detection, instance segmentation, semantic segmentation, and many other areas. Two-dimensional deep learning algorithms have been effectively used for the automatic classification of images and videos, such as the automatic recognition of whether fruit is corrupt for precision agriculture [19], autonomous driving [20–22], and town survey planning [23]. While more of the representational information of 3D objects is reflected in point clouds, there have been many attempts to use deep learning on large 3D point clouds. For example, SnapNet [24] converted a 3D point cloud into a set of virtual 2D RGBD snapshots, which could then be semantically segmented and projected onto the original point cloud data. SegCloud [25] used 3D convolution on voxels and applied 3D fully convolutional neural networks to generate downsampled voxel labels. However, these methods do not capture the intrinsic structure of the 3D point cloud, and converting the point cloud to a 2D format also causes the loss of original information and spatial features. There are also methods for directly processing point clouds that have shown good performance. Point-Net [26] was a pioneering work that used raw point clouds as deep learning inputs in each voxel, while PointNet++ [27] built on PointNet with enhanced local structural information integration capabilities. These point cloud segmentation methods have many extensions and applications in forestry point cloud segmentation. For example, PointNet is used for the independent segmentation of tree crowns [28], and PointNet++ is employed for the semantic segmentation of forestry environments [29]. This paper focuses on the segmentation of tree point clouds (both stem and foliage) as we believe that good tree point cloud segmentation is a prerequisite for obtaining more accurate stand information.

Although all of the above methods have performed well in forestry point cloud semantic segmentation, the semantic segmentation of tree point clouds in artificial forestry scenarios still faces many challenges, one of which is the mismatch of point cloud geometric features in the scene. In an artificial forest environment where tree trunks are mainly characterized by linearity and verticality, tree crowns are mainly characterized by linearity and scattering, the ground is mainly characterized by planarity, and the number of point clouds for each geometric feature does not match the scene, making the network unable to learn the features of each label better. For example, the different numbers of planarity and vertical feature point clouds of branches affect the network's learning of stem labels. The network does not learn enough about the rest of the geometric features, which affects segmentation when there are too many point clouds of one geometric feature. At the same time, as forestry point clouds are characterized by their large scale and disorder, it is difficult to achieve the same results on forestry point clouds with some convolution methods that work well in indoor environments. However, the energy partitioning proposed by [30,31] can partition largescale forestry point clouds into geometric partitions unsupervised, and then our proposed geometric feature balance model (GFBM) is employed to balance the overall geometric features and finally embed PointCNN [32] for feature learning. PointCNN

can preserve the spatial location information of point clouds due to the introduction of X-Conv, which can solve the problem of disorder in forestry point clouds to a certain extent.

In the context of previous studies, extracting tree parameters directly using commercial software is possible but does not exclude the rest of the point cloud in the environment [17]. The Fully Convolutional Neural Network (FCN) series of networks can also be used to classify foliage and stem point clouds, but the results are mediocre [33]. Our paper presents a method based on deep learning for extracting tree feature parameters from artificially planted forest. It removes distracting points from the environment by semantic segmentation and has good segmentation accuracy. It focuses on the following key points: (1) Energy segmentation partitions the original point cloud into geometric partitions; (2) Geometric feature matching balances the geometric features of the whole scene; (3) The geometrically balanced point clouds are embedded in the PointCNN network for learning; (4) The software 3D Forest [34] and TreeQSM [35–38] are used to build a quantitative structure model (QSM) and then obtain standing tree characteristics, such as tree height and diameter at breast height.

2. Materials

2.1. Methodology Overview

The motivation for this work is to provide a method that can extract standing wood feature parameters in an artificially planted forest environment with TLS point clouds. The semantic segmentation method is an important step in this process and works well for trees in different point cloud datasets. Here, we describe how the dataset was built, the model framework and training methods for deep learning, the methods for building QSM models, and how these models and methods were validated. Figure 1 shows a schematic picture of how the method in this paper dealt with an original point cloud. This schematic shows a tool that extracts standing wood feature parameters from the point clouds of artificially planted forests in different complex scenarios. The tool is suitable for the TLS acquisition of forest point clouds (of high resolution). We classified the point cloud data into four labels: foliage, stem, ground, and other points (including shrubs, grass, and a few human figures). The geometric feature balance model is involved in processing the data during training of the deep learning networks but not when segmenting the test point cloud.

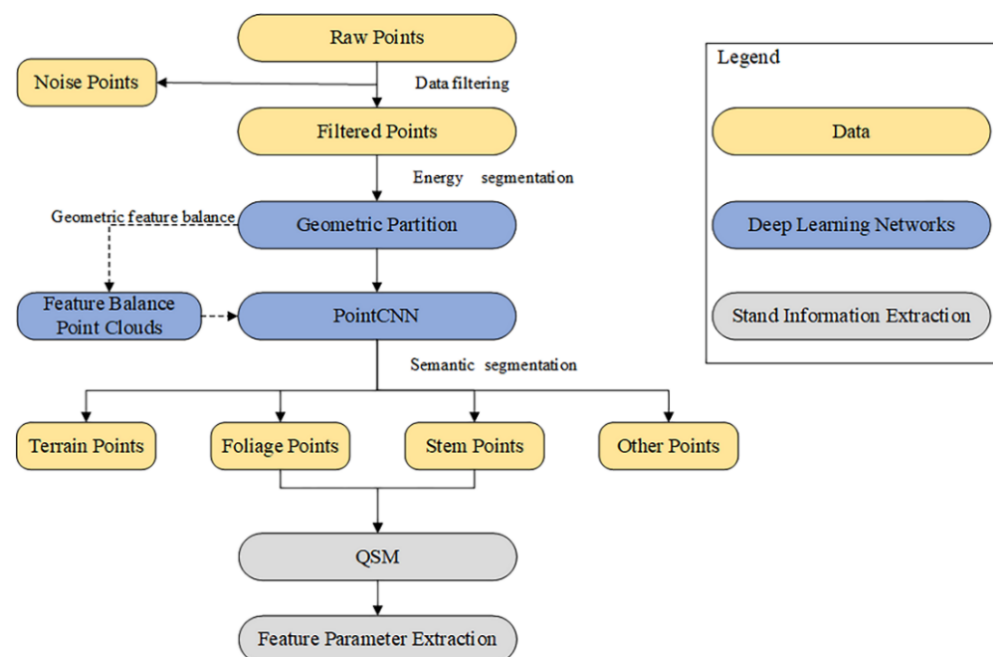



Figure 1. Main steps for the feature parameter extraction of artificial forest based on deep learning.

2.2. Class Selection Approach

The classes for semantic segmentation were chosen based on the visual inspection of the point clouds with color information omitted. Although some frameworks [39–41] can segment point clouds with color and reflection, our model is intended to work on spatial (X, Y, Z) coordinates alone such that it can work on more artificial forest point clouds collected by different equipment. The selected forestry point cloud is subject to certain conditions. Because the purpose of our semantic segmentation was to extract feature information from the collected point clouds of the forestry environment, QSM models needed to be built from the collected point clouds. Thus, the collected point clouds must be relatively complete when stitching together to form a forestry point cloud map, and the point clouds require a certain coverage and accuracy. For example, the DBH and height of trees can at least be measured directly from the tree point cloud, and most LiDAR equipment can collect such point clouds. While some of the stems or branches in the point cloud diagram do not work very effectively when reconstructed, we kept them because they are still important for trunk shape analysis or when analyzing the growth trend of stems. The manual labeling of point clouds requires a high degree of concentration and judgment, and to ensure the consistency of the data in this paper, all datasets in the text were manually segmented by one author. When labeling foliage and stem, if a part of the junction between the two faintly resembled the points of foliage, it was labeled as foliage. When labeling ground points and stems, if a part looked like a point of the ground, then it was classified as ground. Object points above or below the ground except trees were not our primary segmentation targets and were grouped into the other points.

2.3. Study Area

Our data were collected autonomously using terrestrial laser scanning. The extracted terrestrial laser scanning data were obtained using a RIEGL VZ-2000i scanner with high laser pulse repetition rate of up to 1.2 MHz, a field of view of $-40°$ to $+60°$ for vertical scanning and $360°$ for horizontal scanning, a maximum scanning range of 2500 m, and the ability to operate in environments from 0 to 40 °C. The two datasets were stitched by RIEGL's RISCAN PRO software and are in two different coordinate systems and the position of the scanner at the first station is the starting point of the coordinates. RIEGL's unique LiDAR technology, based on waveform digitization, online waveform processing, and multi-echo period processing, enables high-speed, long-range, high-precision measurements in poor visibility conditions, such as dust, haze, rain, and high vegetation cover. The equipment was used to collect the point cloud 77 times at the experimental site during the Bajia Park plantation dataset trial and 98 times on the experimental site during the Gaotang plantation trial. The final point clouds were stored in LAS 1.4 format.

2.3.1. Beijing's Dongsheng Bajia Park Dataset

Beijing's Dongsheng Bajia Park is located at No. 5 Shuangqing Road, Dongsheng Township, Haidian District, Beijing, China (N: 40°01′4.78″ E: 116°20′40.63″). Dongsheng Bajia Park is the largest country park in Beijing, in the temperate monsoon zone, with an average annual rainfall of 688.26 mm and an average annual temperature of 13.1 °C. It covers an area of about 1521.75 mu and is rich in plant species, with more than 21,700 trees and greenery covering over 90% of the area. The plantation forest studied in this paper covers an area of approximately 20 mu, and the tree species planted include mainly tsubaki and poplar. As shown in Figure 2, trees (mainly tsubaki) were used as the object of study to validate our method. The scene consisted mainly of trees (including foliage and stem), the ground, and a number of objects considered to be distracting, including human shadows and light emitters.

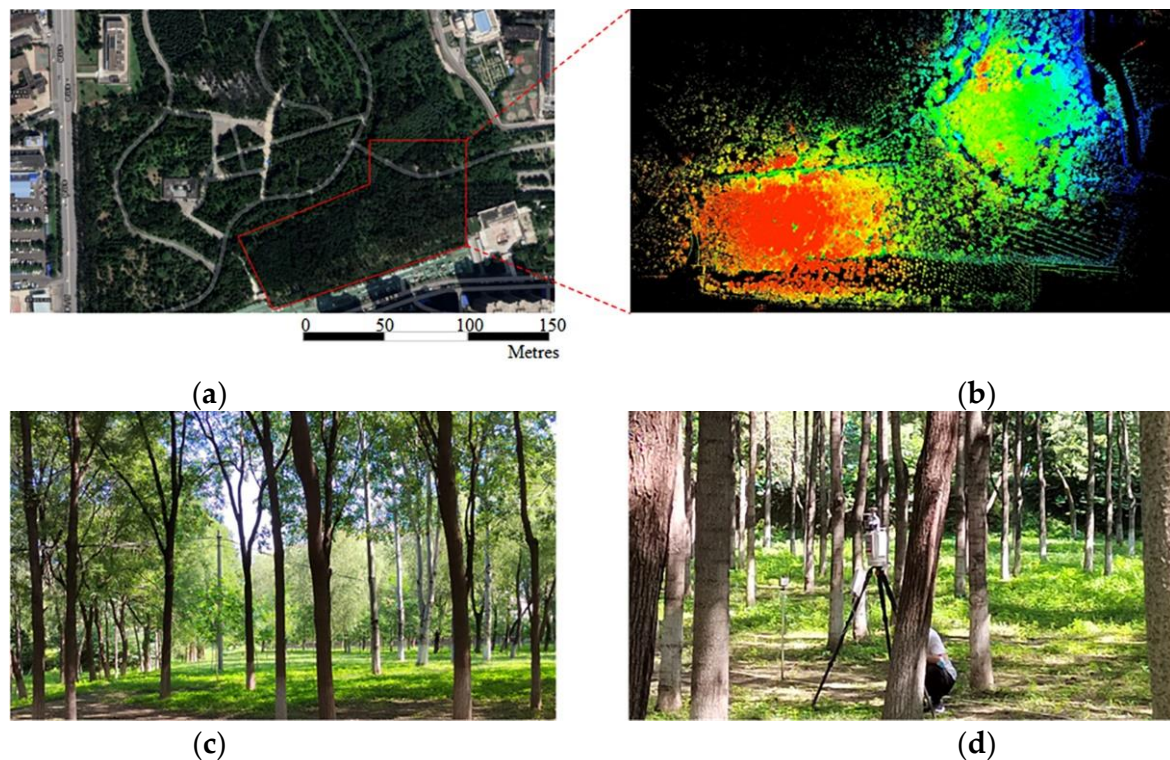



Figure 2. Study area of Bajia Park in Dongsheng, Beijing, China: (**a**) RGB aerial view of Bajia Park; (**b**) Aerial view of the point cloud map of the sample site; (**c**) RGB image of the white sample site shown in the figure, which clearly shows the relationship between tree plant distribution; (**d**) Our use of TLS to collect data.

2.3.2. Gaotang Triploid Populus Tometosa Dataset

The triploid Populus tometosa plantation is located in Qingping Town National Ecological Park, Gaotang County, Liaocheng City, Shandong Province, China (N: 36°48′46.33″ E: 116°05′23.00″). Qingping Town National Ecological Park is the largest plain forest park in Shandong Province, in the temperate monsoon zone, with an average annual rainfall of 589.3 mm and an average annual temperature of 13.0 °C. It covers an area of about 50,000 mu, with a forest coverage rate of over 80% and rich flora and fauna resources. The trees were planted in the spring of 2015 using triploid Populus asexualis B301 ((*P. tomentosa* × *P. bolleana*) × *P. tomentosa*) with an average diameter at breast height of 2 cm and a height of 3 m. The trees were fertilized with conventional fertilizer (170 g of urea per plant per year, in four applications) during the growing season, and a completely randomized group design with three treatments: full irrigation (FI), controlled irrigation (CI), and no irrigation (CK). As shown in Figure 3, 20 mu of poplar was chosen to validate our method. The whole scene consisted of trees (including foliage and stem), the ground, some human shadows that are considered as human disturbance, dwarf shrubs, and grasses on the ground.

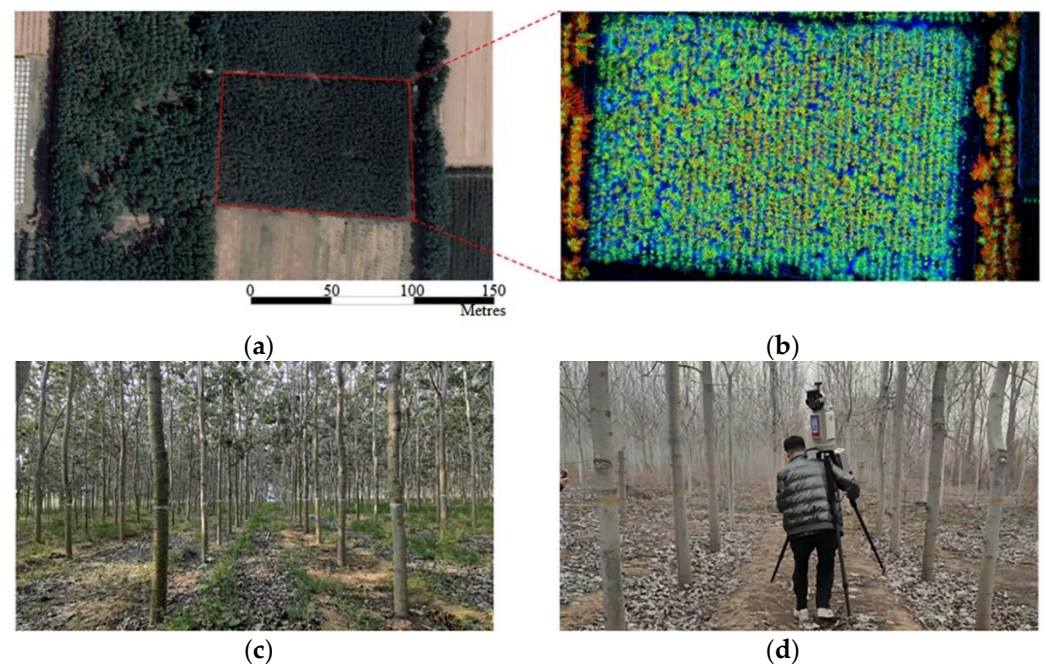



Figure 3. Condition of the Populus tometosa study area: (**a**) RGB aerial view of hairy poplar; (**b**) Aerial view of cloud map of hairy poplar sample site; (**c**) RGB image of hairy poplar sample site clearly illustrating the inter-plant spacing and inter-row relationship; (**d**) Our use of TLS to collect data.

2.4. Data Pre-Processing

2.4.1. Training and Validation Data

The first step in training was to obtain sufficient and high-quality training, validation, and test data. Since the manual labeling of point cloud data is time-consuming and monotonous, especially trunk and foliage labeling that requires a skilled and patient operator, the data were expanded by random rotations of the X and Y axes and multiplication of the axes by random scale variations of 0.6–1.4 times for the training set samples. A large number of samples helps to avoid overfitting, allowing the network to be trained with as much data as possible. We then converted the point clouds in the training and validation sets into the HDF5 [42] format for training and validation.

At the same time, we also generated different training and validation datasets for the two datasets due to their different characteristics. For the Bajia Park dataset, four types of datasets were manually generated during semantic labeling, including (1) Complete and partially mutilated single tree stems; (2) Crown and foliage; (3) Complete ground; (4) Other point clouds in the scene, such as human shadows (operators) and experimental equipment (solar radiometers). In addition to the trait characteristics of the trees themselves being different from the Bajia dataset, the Gaotang poplar dataset also had more plants on the ground in the point cloud map than the Bajia dataset. As we only focused on stand information for trees in this paper, these plants were grouped into other categories in this dataset. The total number of trees in the Bajia dataset was approximately 230 complete tree point clouds and some fragmented tree point clouds, while the number of trees in the Gaotang dataset was approximately 215. The two datasets were divided into a training set and a validation set according to a 7 to 3 ratio of the holdout method, so the Bajia dataset contained 158 trees for training and 72 trees for validation and the Gaotang dataset contained 154 trees for training and 61 trees for validation. Figure 4 shows the details of single tree labeling of the dataset, which illustrates the differences in traits between the poplar and Tsubaki trees in the two datasets. All datasets were annotated according to this criterion during labeling in this work. As shown in Figure 5, our training and validation datasets consisted of 3 to 20 trees in each point cloud; a part of them are shown here.

Figure 4. Schematic diagram of single tree annotation for the Bajia and Gaotang datasets: (**a**,**d**) The whole tree after labeling; (**b**,**e**) Labeling details of the foliage and stem of a single tree; (**c**,**f**) Clear stem of a single tree after labeling.

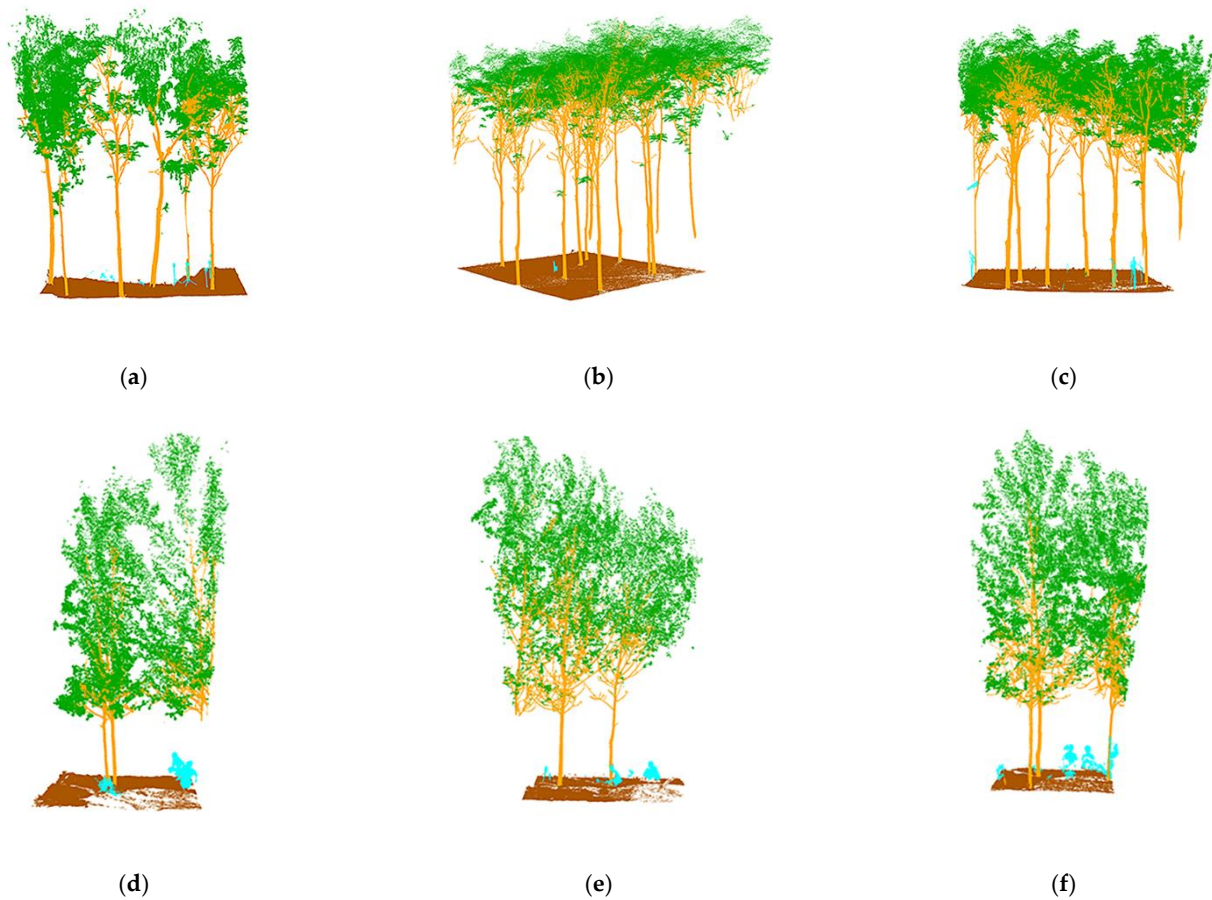



Figure 5. Schematic diagram of dataset labeling data: (**a**–**c**) Labeling of part of the Dongsheng Bajia dataset; (**d**–**f**) Labeling of part of the Gaotang triploid Populus tometosa dataset.

2.4.2. Testing Data

The accuracy and robustness of our semantic segmentation method were tested by taking a block of sample data from the Bajia Park and Gaotang poplar data. Detailed parameters of the test set are shown in Table 1.

Table 1. Detailed parameters for the test set of our two self-built datasets.

Data Name	Sensing Method	Forest Type	NT	NP	RTP	Area
Baijia Park	Terrestrial Laser Scanner	Artificially	74	28,648,947	97.99%	916.424 m^2
Gaotang Park	(RieglVZ-2000i)	planted woods	57	22,782,508	55.66%	388.370 m^2

NT: Number of trees; NP: Number of scanned points; RTP: Ratio of tree point cloud to scanned points.

3. Methods

The deep learning framework we used in this work was based on energy segmentation, our proposed geometric feature balance model (GFBM), and PointCNN, as shown in Figure 6. The energy segmentation function was used as a pre-segmentation framework for point clouds, allowing point clouds to be efficiently segmented into smaller geometric partitions based on object geometry without losing major fine details, and its energy segmentation method for forming geometric partitions is mainly listed here. Each geometric partition matched the geometric features after entering the GFBM so that the geometric features of the whole scene were balanced. A PointCNN based on TensorFlow [43] was embedded in the subsequent semantic segmentation using a convolutional network to learn the input point cloud features.

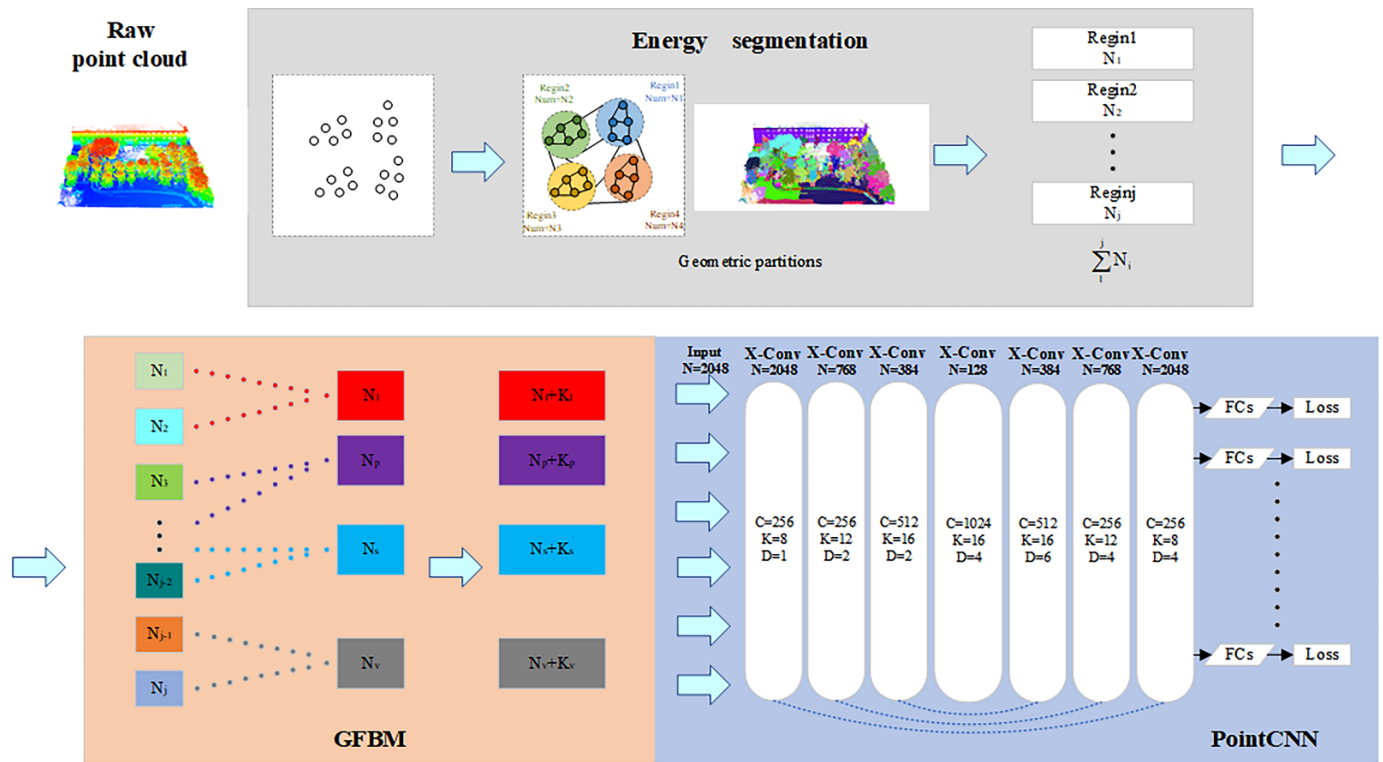



Figure 6. The deep learning network structure in this paper, where N_1–N_j represent each geometric partition; N_l, N_p, N_s, and N_v represent the point clouds of linear, planarity, scattering, and vertical as the main features, respectively; K_l, K_p, K_s, and K_v represent the point clouds of each feature added after the geometric features are balanced. In PointCNN, in each X-Conv operation, N represents the number of points in the next layer, C represents feature dimensionality, K rep-resents the number of nearest neighbors, and D represents the dilation rate.

3.1. Energy Segmentation Network

We describe the process of energy segmentation network in this section, where the input raw point cloud was computationally energy segmented, allowing the transformation of raw input point cloud data of millions of points into a few hundred geometric partitions, where the local geometry of the points within each partition was similar.

For the input point cloud P, the geometric partitioning was calculated based on the features of its 3D geometry. The point cloud was geometrically partitioned according to the above four features: linearity, planarity, scattering, and verticality. Each point will only belong to one geometric partition.

According to [44], these features were defined by the local domain of each point in the point cloud. The eigenvalues for each point $\lambda_1 \geq \lambda_2 \geq \lambda_3$ were calculated of the covariance matrix of the positions of the neighbors. The neighborhood size was chosen such that it minimized the eigentropy E of the vector $(\lambda_1/\Lambda, \lambda_2/\Lambda, \lambda_3/\Lambda)$, where E represents the point cloud adjacency relationship. According to the best neighbor principle proposed by Weinmann et al. [44], $\Lambda = \sum_{i=1}^{3} \lambda_i$, which is in accordance with the optimal adjacency:

$$E = -\sum_{i=1}^{3} \frac{\lambda_i}{\Lambda} \log(\frac{\lambda_i}{\Lambda}) \tag{1}$$

According to the findings of [45], a formula for the linearity, planarity, and scattering of the local neighborhood can be presented based on these eigenvalues.

$$\text{linearity} = \frac{\lambda_1 - \lambda_2}{\lambda_1} \tag{2}$$

$$\text{planrity} = \frac{\lambda_2 - \lambda_3}{\lambda_1} \tag{3}$$

$$\text{scattering} = \frac{\lambda_3}{\lambda_1} \tag{4}$$

Linearity describes how elongated the neighborhood is, planarity describes how well it is fitted by a plane, and high-scattering values correspond to an isotropic and spherical neighborhood. These three characteristics combine to form dimensionality. Verticality can also be obtained from the definition of eigenvectors and the values defined above. Let μ_1, μ_2, μ_3 be the three eigenvectors associated with, respectively, λ_1, λ_2, λ_3. We then define the unary vector of principal direction in R^3 as the sum of the absolute values of the coordinate of the eigenvectors weighted by their eigenvalues.

$$[\hat{u}]_i \propto \sum_{j=1}^{3} \lambda_j \left| [u_j]_i \right|, \text{pour} i = 1, 2, 3 \ et \ \|\hat{u}\| = 1 \tag{5}$$

We considered that the vertical part of this vector characterizes the verticality of a point field.

In this article, the generalized minimal partition problem was studied by referring to the partition problem of global properties. For each point i, we computed a vector of geometrical features and associated its local geometric feature vector $f_i \in R^4$ (dimensionality and verticality) to calculate piecewise constant approximation g*, where g* is defined as the vector of $R^{4 \times P}$ minimizing the following Potts segmentation energy. We obtained the point cloud geometric partition by solving this optimization problem.

$$g* = \underset{g \in R^{4 \times P}}{\operatorname{argmin}} \sum_{i \in P} \|g_i - f_i\|^2 + \rho \sum_{(i,j) \in E} \omega_{i,j} [g_i - g_j \neq 0] \tag{6}$$

In the above equation, [ ] is an Iverson bracket; for any point i belonging to P, ρ is the regularization factor and influences the coarseness of the partition; $\omega_{i,j}$ is the edge weight, equal to 0 in 0 and 1 everywhere else. For the partition, l_0-cut pursuit [46] was used to solve this energy partitioning problem. The advantage of this method is that it does not require the definition of the size of the point cloud and the different energy partitions of the whole scene are obtained quickly after calculation.

3.2. Geometric Feature Balance Model

The main function of this module is to balance the geometric features of the point cloud input to the network. For example, for the trees in the point cloud, most of the branches have scattering geometric features predominantly, but there are also branches with more vertical or planarity geometric features, and increasing the number of these branches in the training set is beneficial to enhance the network's learning of the details of the stem label. In this paper, the geometric features of the whole scene are balanced in-stead of a particular label, which is beneficial for global features and less computationally intensive. The overall process is shown in Figure 7.

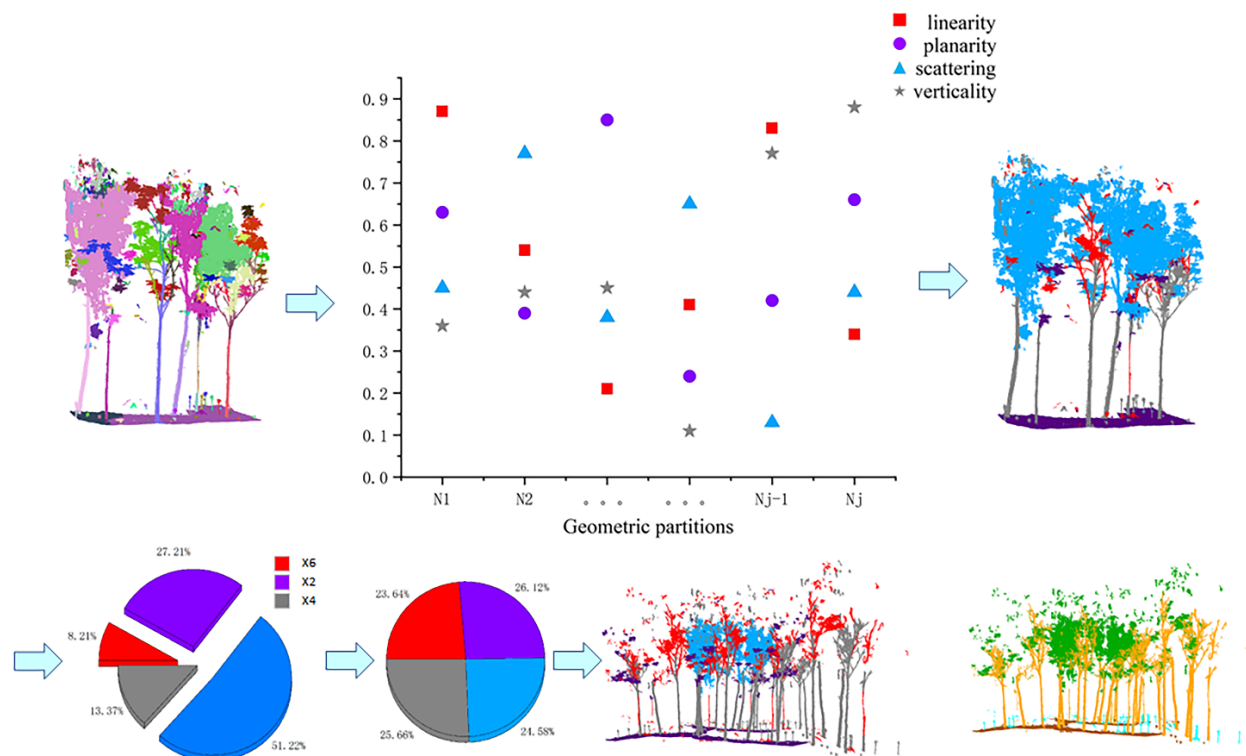



Figure 7. Using Figure 5a as an example, the main geometric features selected for each geometric partition are visualized.

The formula for calculating the four geometric features of the local neighborhood was shown in the previous Section 3.1. For each geometric partition, the average value of its four geometric features was calculated separately, and then the most representative feature of each geometric partition was selected as the geometric feature of this partition so that the number of point clouds with the four features as the main features in the whole scene could be obtained, respectively.

After the above operation, we obtained four types of point clouds with linearity, planarity, scattering, and verticality as the main features, namely, P_l, P_p, P_s, and P_v. The geometric feature balancing strategy was to use the largest number of point clouds in the four-point cloud feature sets as the quantitative benchmark, and the rest of the geometric feature sets were aligned by this benchmark in order of magnitude, as shown in Equations (7)–(9) below. The geometric features were balanced by copying and panning the point cloud, adding a random angle of rotation to the panning to increase the generalizability of the network. Meanwhile, to ensure that the overall geometric features of the scene do not change as a result of the rotation operation, the rotation was performed with a reference axis paired with Z.

If

$$P_l > P_i (i = p, s, v) \tag{7}$$

obtain

$$K_i = [\frac{P_l - P_i}{P_i}] \bullet P_i \tag{8}$$

or obtain

$$K_i = [\frac{P_l - P_i}{P_i} + 1] \bullet P_i \tag{9}$$

3.3. PointCNN Deep Learning Network

PointCNN solves the point cloud disorder problem by employing transpose matrices. Compared to PointNet, which uses symmetric functions to deal with point cloud disorder,

PointCNN can reduce feature loss. In PointCNN, we used an encoder–decoder paradigm, it is called X-conv, where the encoder reduces the number of points while increasing the number of channels. Then, the decoder part of the network increases the number of points, and the number of channels is incrementally reduced. The network also uses the same "skip connection" architecture as U-Net [47]. The most important characteristic of X-conv is that it can both weight and guarantee the invariance of the input features, and then apply the traditional convolution to the features, it is the basic block of PointCNN.

PointCNN differs from traditional grid-based CNNs in two main ways. First, the method of local region extraction is different. While CNN extracts local features directly through K × K blocks, PointCNN extracts local features by representing K neighboring points on a point and then fusing the features in the K neighborhood by weighting the sum, enabling it to achieve the same effect as a convolution operator fusing domain features in regular data. Second, the method of local region information learning is also different. CNN usually extracts image features by Conv and then pools downsampling, while PointCNN uses X-Conv to extract features, aggregating them into fewer points to increase the channels to recursively learn the correlation with the surrounding points. A comparison of the two methods is shown in Figure 8.

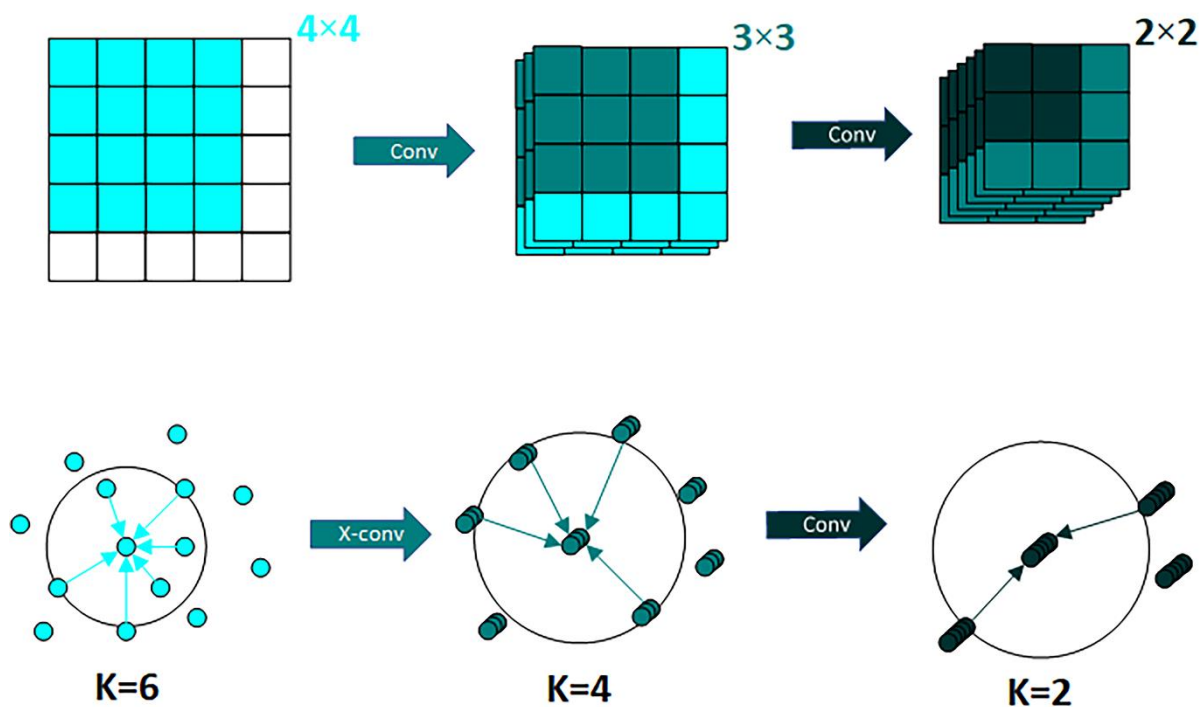



Figure 8. The different processes of CNN and PointCNN feature fusion.

3.4. Training Details and Performance Measures

We provide more details of the training in this section. All training and testing were conducted on a personal computer, with CUDA-accelerated computation using an Nvidia 3060 GPU during the training process. A development environment of Python 3.6 and TensorFlow GPU 2.4.1 was set up on Ubuntu 18.04, with a basic learning rate of 0.0002 and batch study size of 8. The network was trained over 150 rounds, and all were trained using a randomized dropout method, which was applied before the last fully connected layer to reduce over-fitting. This method can effectively improve the generalization of the training process and make the algorithm perform well on sparse point clouds.

A comparison between the manually measured real values and the QSM measurements was carried out as a reference for the effectiveness of standing wood feature information extraction. The Softmax cross-entropy function was used as the loss function of the deep learning network. To evaluate the performance of the semantic segmentation model,

Python packages Numpy and Seaborn were used to evaluate our results and generate confusion matrices. *IoU* is the evaluation index for each category, and *OA* is the overall precision evaluation index of the dataset. Precision indicates the proportion of actual positives to predicted positives and Recall indicates the proportion of actual positives that are correctly predicted.

$$IoU = \frac{TP}{TP + FP + FN} \tag{10}$$

$$OA = \frac{TP + TN}{TP + TN + FP + FN} \tag{11}$$

$$Precision = \frac{TP}{TP + FP} \tag{12}$$

$$Recall = \frac{TP}{TP + FN} \tag{13}$$

where *TP* is the true positive, *TN* is the true negative, *FP* is the false positive, and *FN* is the false negative.

3.5. QSM Formation and Feature Parameter Extraction

The QSM of a tree is the structural model of the tree, describing its basic branch structure and geometric and volumetric properties. These properties also include the total number of branches of the tree and the parent–child relationship of the branches, the length, volume, and angle of individual branches, and the branch size distribution. There are other properties and distributions that can be easily calculated from the QSM. The QSM consists of construction blocks, usually of some geometric shape, such as cylinders and cones. The cylinder was used here as it is the most reliable and is highly accurate for estimating diameters, lengths, orientations, angles, and volumes in most situations. A QSM consisting of cylinders provides a downsampled representation of the tree and can store a lot of information about the tree, as mentioned previously.

In actual cases, using the semantically segmented tree point cloud followed by QSM and standing wood feature information extraction can reduce the interference of the rest of the point cloud in the environment on the accuracy of the tree information and also prove the necessity and accuracy of our point cloud segmentation method.

In this paper, 3D Forest software was used to instance segment the semantic segmented tree point clouds, and then TreeQSM was used to extract standing wood feature information from our segmented point clouds by fitting columns to convert the point clouds into QSM models, which can represent over 99% of our segmented tree trunk point clouds with an accuracy of over sub-millimeter. TreeQSM has two key steps in extracting tree parameters; the first step is the topological reconstruction of branching structure, which is the segmentation of the point cloud into stem and individual branches. The second step is the geometrical reconstruction of branch surfaces, which is realized by fitting cylinders. For more details on the original TreeQSM method, please refer to the original articles [35–38].

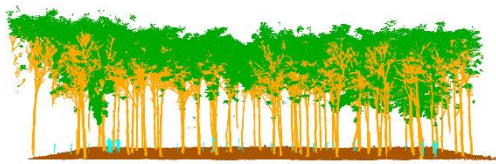
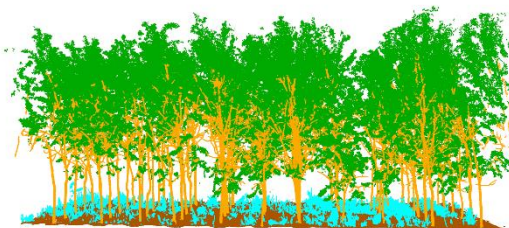
4. Results

The segmentation results for the two test datasets are shown here, including the energy segmentation results, the semantic segmentation results, and the results of standing wood feature parameter extraction for the processed tree point clouds.

4.1. Semantic Segmentation Results

The energy partitioning results for the test set are shown in Figure 9a,b, where the left-hand side shows the partitioning results for the Bajia dataset and the right-hand side shows the partitioning results for the Gaotang dataset. It can be seen that the morphology of the poplar and Tsubaki trees in the two datasets are completely different; the Gaotang dataset that has more branches and leaves has significantly more geometric partitioning. Due to the disorderly nature of the point cloud, the energy partitioning of the test set does

not affect the final result of the semantic partitioning and is only shown here to demonstrate the process of unsupervised segmentation.

(**a**) Bajia energy segmentation   (**b**) Gaotang energy segmentation

(**c**) Bajia Reference   (**d**) Bajia Prediction

(**e**) Gaotang Reference   (**f**) Gaotang Prediction

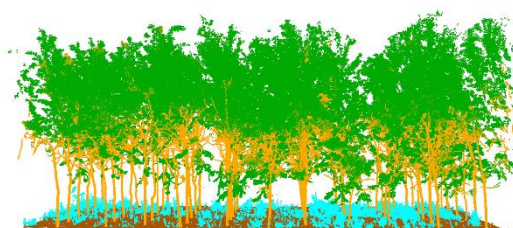



Figure 9. (**a–f**) Results of the energy and semantic segmentation of two different datasets, in the semantic segmentation results. The left graph shows the manually calibrated reference and the right shows the predicted results.

The semantic segmentation results are shown in Figure 9c–f, with the manually annotated point clouds on the left and the segmented point clouds on the right, which are the result of energy segmentation into geometric partitions followed by semantic segmentation. These point cloud inspection sets are visually very similar to the reference data for manual segmentation. For the segmentation results, the tree point cloud is well-segmented from the whole point cloud. While some stems and foliage are misclassified into ground and other points, and some ground is misclassified into foliage, which is not common. In the Bajia test set, a small number of points of trees are misclassified as ground, which occurs in point clouds located at the edges of the data sample and in the Gaotang dataset. While some of the shrubs in the other point categories are misclassified as stems mainly in the root point cloud, the number of misclassifications is not significant.

The training accuracy and loss curves during the training process of the Bajia and Gaotang data sets are shown in Figure 10. In the training process, the training accuracy tends to increase while the training loss function tends to decrease, which indicates that our network has a good learning ability for global features. For the Bajia dataset, the total training and validation time is approximately 120 h. Additionally, for the Gaotang dataset, the total training and validation time is approximately 110 h. After 130 rounds of training, the training accuracy and loss curve stabilize in the Bajia dataset, with the training accuracy and loss function converging to 0.95 and 0.14, respectively, and in the Gaotang dataset, these figures are closer to 0.93 and 0.18, respectively.

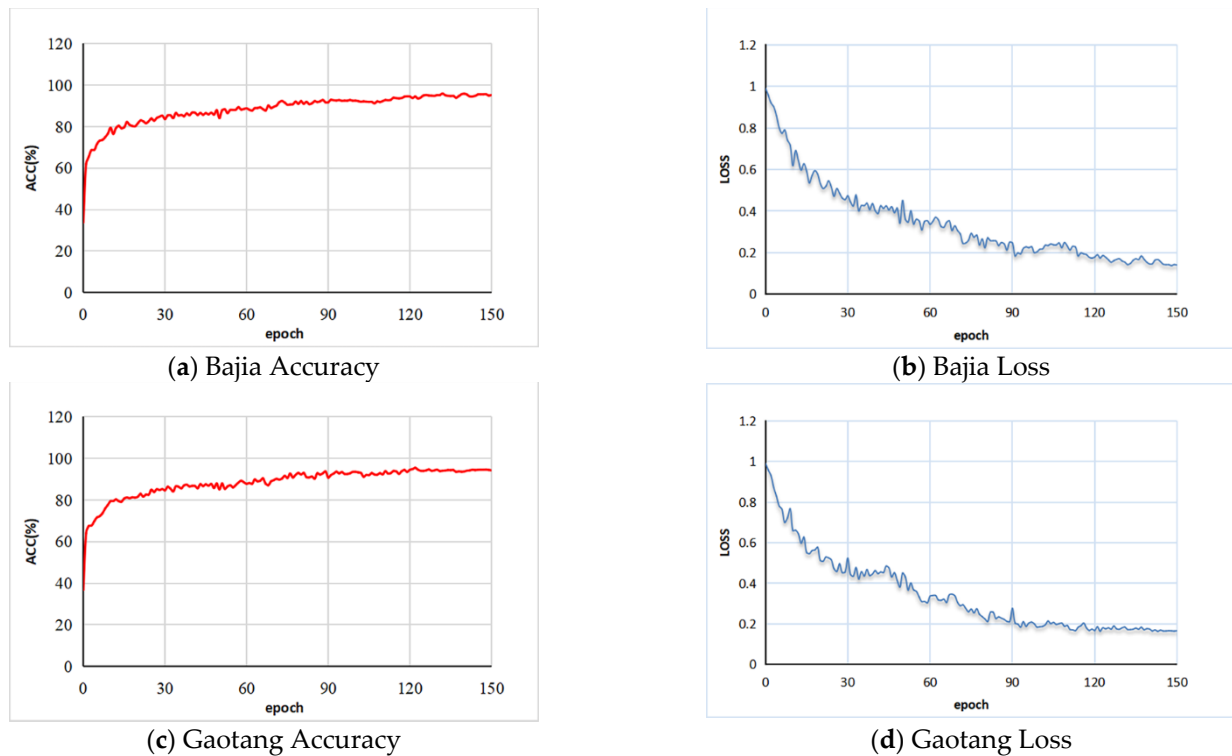



Figure 10. Training accuracy values and loss function curves.

For the Bajia and Gaotang datasets, the segmentation results achieve our goal of data processing, and the tree point cloud is well-segmented from the overall point cloud. Of the four labels, we observed that for the three categories of ground, foliage, and other points, the model has significantly higher accuracy than the stem label when predicting and that the other point category with the fewest points in the entire map has the highest precision. This is illustrated in Figure 11, which shows the semantic partitioning confusion matrix for the overall two test sets.

In both test datasets, the overall accuracy of trees (including stem and foliage) in the Bajia dataset is 0.94, significantly higher than the 0.90 of the Gaotang dataset. This may be because the leaves of the Tsubaki trees in the Bajia dataset are mainly concentrated at the top of the trees, and the branch angle is large, so the leaf feature network is easier to learn. In contrast, the triploid poplar planted in the Gaotang dataset has a large number of branches and a large branch-to-diameter ratio, the tree canopy envelope is tighter, the leaves are smaller and more numerous, and there are a large number of ground plants that affect the segmentation accuracy of the tree point cloud and cause the trunk to be misclassified into leaves more often.

In these two datasets, the ground class had the highest Recall, reaching 0.983 and 0.988, respectively. The other classes have high Precision but low Recall in the Bajia dataset, probably because the other point classes contain several types of unappreciated points, including light-emitting instruments and human shadows. As these objects are more different and less numerous in the overall point cloud map, their features are not well learned by the network, which leads to their misclassification as stems. The detailed metric parameters are shown in Table 2.

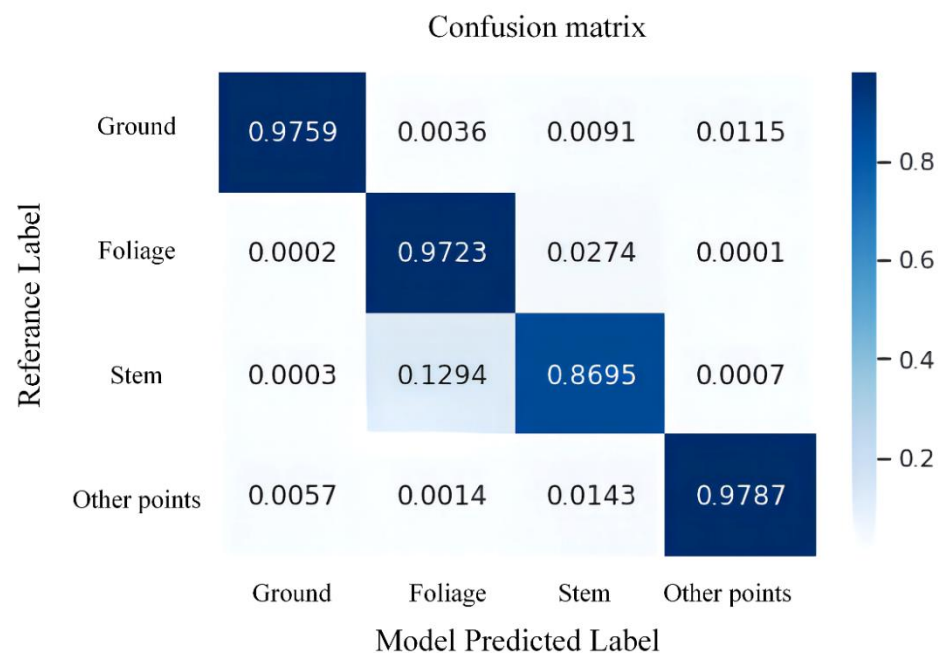



Figure 11. A confusion matrix showing the results of manual labeling compared with semantic segmentation.

Table 2. The Recall, Precision, Overall Accuracy, and *IoU*.

Dataset	Indicators	Ground	Foliage	Stem	Others
Bajia Park	Precision	0.932	0.983		0.890	0.959
	Recall	0.983	0.903		0.980	0.618
	IoU	0.917	0.889		0.875	0.602
	OA			0.934		
Gaotang Park	Precision	0.979	0.952		0.816	0.979
	Recall	0.988	0.886		0.898	0.935
	IoU	0.977	0.848		0.747	0.916
	OA			0.938		

4.2. Comparison of QSM Results with Manual Measurements

The segmented point cloud was used to generate a QSM model using 3D Forest software and MATLAB-based TreeQSM and then extracted to analyze the tree feature information, which was an important step in our overall workflow. Semantic segmentation was carried out to prepare for the extraction of stand information to deal with the interference of distracting objects in the artificially planted woods on the extraction of tree feature information.

A brief demonstration of the extracted tree height and diameter at breast height data and its comparison with the actual measurements after forming the QSM model for the test dataset is shown in Figure 12. The correlation coefficient R^2 and the root-mean-square error were calculated separately for both datasets to qualitatively evaluate our results.

As shown in Figure 12, a total of 74 trees from the Bajia dataset and 57 trees in the Gaotang dataset were counted for both height and diameter at breast height compared to the manual measurements. It can be seen that the accuracy of both height and diameter at breast height is higher in the Bajia dataset, likely because the branches of the Tsubaki tree branch are at large angles and cross over less, and the diameter at breast height is relatively larger.

In both datasets, the accuracy of tree height seems better than diameter at breast height, the point clouds of the trees are fitted frame by frame, and some deviations may occur during the fitting process, but this is acceptable and the QSM algorithm still does

not fit the diameter at breast height well enough, increasing the measurement error. The results of the standing wood information extraction achieved by QSM are briefly presented here, illustrating the importance and effectiveness of the semantic segmentation work in this paper.

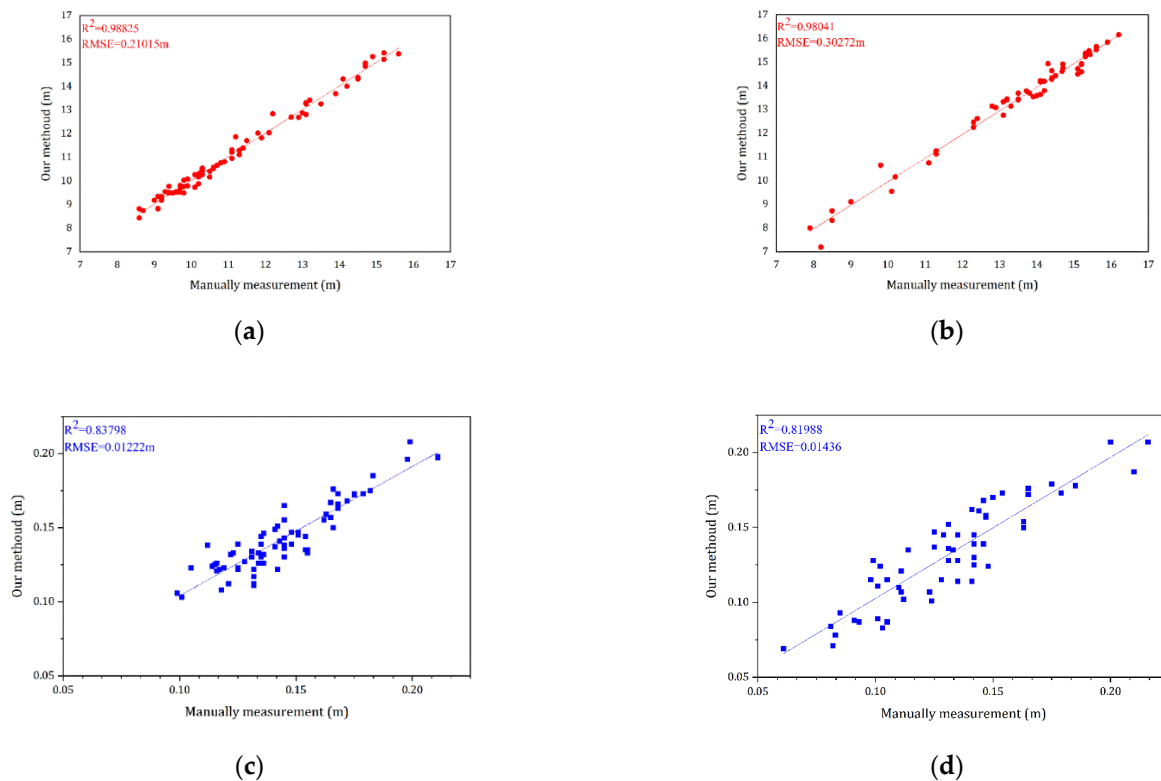



Figure 12. (**a**,**c**) Comparison of tree height and diameter at breast height with manual measurements for the Bajia test set. (**b**,**d**) Comparison of tree height and diameter at breast height with manual measurements for the Gaotang test set.

5. Discussion

5.1. Evaluation of Our Approach

Extracting information on standing tree characteristics in forestry environments through LiDAR scanning techniques is of great importance for forestry automation. Analyzing the physical parameters of trees can be very helpful for studying the relationship between the standing canopy and sap flow, light, soil, etc. Our approach provides a new method to reliably extract tree feature information from TLS point clouds of artificially planted woods. The automatic extraction of tree point clouds from laser scanned data is an important prerequisite for standing feature information extraction, tree phenotype, and biophysical parameter estimation. The semantic segmentation method in this paper provides a new and reliable method for extracting tree point clouds from LiDAR-scanned forestry point clouds. This semantic segmentation method enhances the learning of the network for artificially planted woodland object features by balancing the geometric features in the scene, and the point clouds are segmented by PointCNN. The semantic segmentation in both forestry scenes obtains good segmentation results, and this method has high robustness.

However, there are still some limitations to this work. For example, the manual labeling of point clouds remains highly subjective. While the ground and other points are generally correctly labeled, when labeling tree stem and foliage, although the majority of points are labeled accurately, a small number of points are inevitably incorrectly labeled due to the limited time available and the fact that some of the blurred points are difficult to distinguish manually. In Section 2.4.1, we also showed the scale of our labeling of point

cloud categories, where the fuzzy stem-like point clouds are labeled as foliage-like so that the network can learn the features of the complete stem point cloud to the maximum ex-tent and better segment the stem point cloud from the overall point cloud. We believe that the obtained overall segmentation accuracy of 0.9 for the tree point cloud achieves our de-sired goal. According to our segmentation study on the two datasets, the accuracy of tree segmentation in the Bajia dataset is 4% higher than that in the Gaotang dataset. The main reason for this phenomenon may be the fact that the physical parameters of the Tsubaki tree in the Bajia dataset are completely different from those of the triploid aspen in the Gaotang dataset. The Tsubaki tree has an oblate crown with many branches, and the leaves are relatively large, whereas the long branch leaves of the triploid hairy poplar are broadly ovate or triangular–ovate leaf-shaped and relatively small.

Additional results are also significant. For example, in the Bajia dataset, some tree point clouds on the upper boundary of the point cloud edge are misclassified as ground class, although the number of misclassifications is very small, which may be due to the similarity between the point cloud boundary features and ground features. In the Gaotang dataset, leaves on branches are misclassified as stems, which may be due to the lack of de-tailed labeling or poor learning of network features. However, this does not have a negative impact on the extraction of standing tree information.

Overall, the method of extracting information from artificially planted woods explored in this paper effectively extracts standing tree feature information, and the semantic segmentation method maximizes the preservation of spatial features of the point cloud and achieves good performance in the final test. The network obtains optimal weights through iteration during the training process, making the model robust in identifying the point clouds that form the structure of tree trunks and leaves.

5.2. Comparison with Similar Methods

We also compared our experimental results with the results of other papers. It is worth noting that as different data and methods of labeling the data were employed, these values do not necessarily characterize the absolute performance of the algorithm, but still provide a certain reference for our research.

A comparison of our study with other papers can be seen in Table 3, where the definitions of the classes are different in each paper, but essentially all contain two classes: leaves and trunk. In the above study, of the four classes counted, [29] performs best in terms of overall accuracy, which employs a method based on PointNet++. By comparison, our method performs better in the ground and other point classes, with similar accuracy in the foliage and lower accuracy in the stem classes. In contrast, [48] compares a variety of methods, applying a 3D convolutional neural network on voxels and PointNet to segment the dataset, and also compares data with intensity and without intensity, respectively. An overall accuracy of 0.925 was obtained in [49], which used an approach based on unsupervised learning.

Although there are a number of limitations to our comparisons here, our method obtained more accurate results. This comparison provides a clear understanding of the differences between methods, which will remain a reference for our future research work.

We also compared different semantic segmentation methods on both the Bajia and Gaotang datasets using three algorithms, including the MVF CNN [50] (which also uses CNNs), the point-based method PointNet, and the original unaltered PointCNN network.

MVF CNN is a deep learning-based multi-scale voxel and feature fusion algorithm for large-scale point cloud classification. First, the point cloud is transformed into two different-sized voxels. The voxels are then fed into a three-dimensional convolutional neural network (3D CNN) to extract features. Next, the output features are fed into a recommended global and local feature fusion classification network (GLNet), and the multi-scale features of the main branch are finally fused to obtain their classification results.

PointNet uses the input of the original point cloud to maximize the spatial characteristics of the point cloud, partitioning the input into voxels of uniform size. The input

of the network is the 3D coordinates ($n \times 1024 \times 3$) of the three-dimensional point cloud containing n voxels and 1024 points within a voxel. This is then fed into the network for training.

Table 3. Comparison of segmentation results against similar studies. Bold numbers denote the top score in each metric.

Study	Method	Ground Precision	Foliage Precision	Stem Precision	Others Precision	Overall Accuracy
48	3D Fully Convolutional Network	-	0.985	0.595	-	-
	Convolutional Network (with LiDAR intensity)	-	0.985	0.652	-	-
	PointNet	-	0.976	0.517	-	-
	PointNet (with LiDAR)	-	0.985	0.554	-	-
49	Unsupervised Learning	-	-		-	0.925 *
29	Modified PointNet++ approach	0.926	0.974	0.948	0.610	0.961 * 0.954 **
Ours	Improved Point CNN Method	0.976	0.972 0.970 ***	0.870 0.934 ***	0.979	0.925 * 0.936 ** 0.958 ***

* Including stem and foliage classes; ** Including all of our classes (ground, foliage, stem, and others). *** We compared against the TLS_1 and TLS_2 datasets from [29]; the datasets contain foliage and stem.

PointCNN takes the structure from the original paper and does not change it, partitioning the input point cloud into uniformly sized voxels to feed into the network for training and prediction of the test set.

In this work, the network was trained, the test set was partitioned, and the maximum epoch was set to 200. The batch size used eight samples, and the learning rate was 0.0002. The segmentation results are shown in Figure 13, and quantitative evaluations of the results are provided in Table 4.

(a)   (b)   (c)

(d)   (e)   (f)

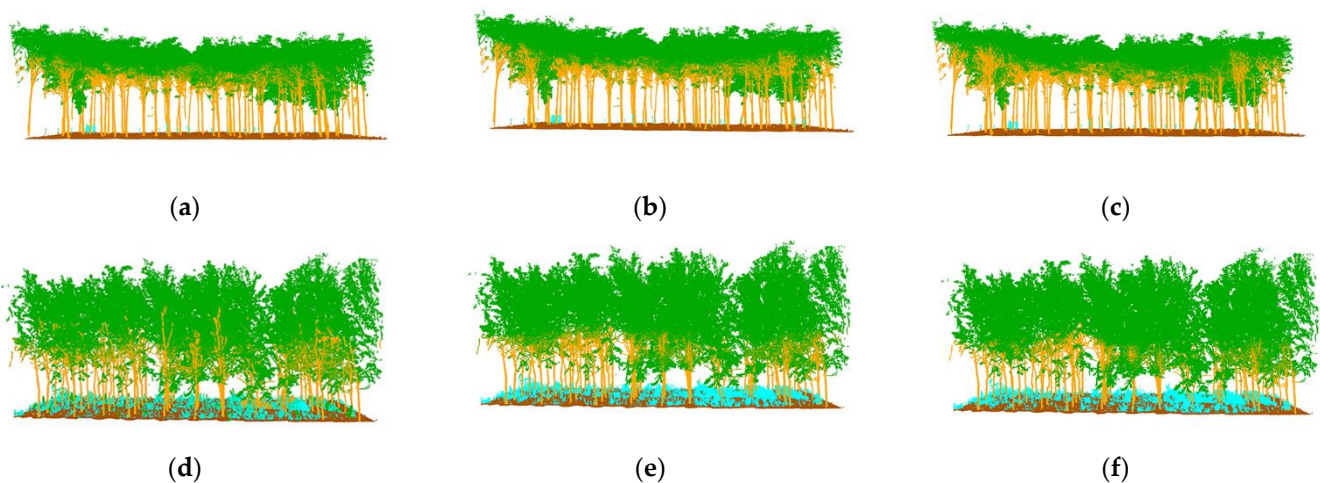



Figure 13. (**a**,**d**) MVF CNN segmentation; (**b**,**e**) PointNet segmentation; (**c**,**f**) PointCNN segmentation.

Table 4. Comparison of the Precision of each method.

Method	Ground Precision	Foliage Precision	Stem Precision	Others Precision	Overall Accuracy
MVF CNN	0.908	0.832	0.870	0.679	0.850
PointNet	0.873	0.840	0.857	0.828	0.851
PointCNN	0.913	0.902	0.909	0.938	0.908
Improved Point CNN Method	0.976	0.972	0.870	0.979	0.936

Among them, MVF CNN and PointNet show a similar accuracy of 0.85, and both methods have good accuracy in trunk and foliage classes. The original PointCNN has a higher accuracy of 0.91 compared to the previous two methods, and its stem precision is the best among the four methods. The overall accuracy of these three methods is not as good as our method. This suggests that our deep learning framework performs better in extracting spatial features of trees when processing point clouds of artificially planted trees captured by TLS.

5.3. Future Research Directions

In future work on this project, our focus will be on improving the accuracy of network segmentation. In practice, the higher the accuracy of the semantic segmentation, the less manual correction will be required, which will significantly reduce manual effort and facilitate the fully automated segmentation of the forest point cloud. We intend to increase the amount of data in the training set, add the manually corrected test set after segmentation to the training set, and iterate the training model again to enhance the recognition capability of the network. We will also explore the applicability of our method in different acquisition techniques, such as backpack radar and aerial radar. This will enable us to determine its applicability on point clouds collected by more devices, test its effectiveness in different forestry contexts, such as primary forest or urban greenery, and explore its segmentation effectiveness in larger contexts. The problem of shading between branches can affect the accuracy of manual labelling and the classification results of trees, just as shading can also have an impact on leaf area calculations [51]. In future work, we will also consider how to address this issue, for example, by considering whether the use of aerial radar data [52] would reduce this effect or by considering some graph-based deep learning networks [53].

We will also explore the effectiveness of different neural networks for segmenting point clouds of artificially planted trees in future work. The varied results of the two datasets in this paper indicate that different tree species may behave differently on the same network when using the same tree feature information extraction method, thus whether different neural networks are suitable for different tree species point cloud data. Exploring whether there is a relationship between the choice of neural network and the ability to segment tree point clouds in a forestry environment is of great significance for establishing a fully automated method for extracting stand information.

6. Conclusions

This work aimed to obtain a complete ground-based radar point cloud tree information extraction method of artificially planted trees to help us to better investigate the relationship between the 3D physical information of trees, tree growth, and tree cultivation practices. We divided the work into forestry point cloud map building, deep learning-based semantic segmentation, and QSM-based tree feature information extraction.

The forestry map was built using RIEGL equipment and this paper focused on our proposed semantic segmentation method based on deep learning as the forestry point cloud collected by LiDAR has noise, and point clouds of other objects are not relevant. Semantic segmentation is an extremely important component for excluding the influence of other point clouds on the tree point cloud. Although the need for extracting tree point

clouds in some practical applications can also be solved by direct manual segmentation, as human energy is limited and the forestry environment has a vast volume of data, semantic segmentation based on deep learning is undoubtedly the best approach. The point clouds are then processed using existing QSM methods to effectively obtain information on the standing wood features of the target.

Our method showed good segmentation results on the dataset, with RMSEs of 0.30272 and 0.21015 m for tree height and 0.01436 and 0.01222 m for diameter at breast height in both datasets, respectively, with a high overall accuracy of 0.95 for semantic segmentation and 0.93 for trees. Compared to the manual segmentation of point clouds, our method has considerable advantages as an automated process for extracting feature information from artificial woodland point clouds collected by TLS, providing a strong foundation for creating a fully automated and high-precision method for extracting information from artificial woodland stands.

Author Contributions: Conceptualization, X.S., Q.H. and J.L.; Data curation, X.W. and B.X.; Investigation, X.S., Q.H., J.L., X.W. and B.X.; Project administration, Q.H.; Resources X.S., Q.H., J.L., X.W. and B.X.; Software, X.S.; Writing—original draft, X.S.; Writing—review and editing, X.S., Q.H. and J.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by The National Key Research and Development Program of China: 2021YFD2201203.

Data Availability Statement: Some restrictions apply to the availability of these data. The artificially planted forests in this article are of great research value, and the dataset obtained is a trade secret, but we will publish some of the data in our subsequent work.

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

References

1. Bogdanovich, E.; Perez-Priego, O.; El-Madany, T.S.; Guderle, M.; Pacheco-Labrador, J.; Levick, S.R.; Moreno, G.; Carrara, A.; Pilar Martín, M.; Migliavacca, M. Using terrestrial laser scanning for characterizing tree structural parameters and their changes under different management in a Mediterranean open woodland. *For. Ecol. Manag.* **2021**, *486*, 118945. [CrossRef]
2. Sun, Y.; Liang, X.; Liang, Z.; Welham, C.; Li, W. Deriving Merchantable Volume in Poplar through a Localized Tapering Function from Non-Destructive Terrestrial Laser Scanning. *Forests* **2016**, *7*, 87. [CrossRef]
3. Li, X.; Lin, H.; Long, J.; Xu, X. Mapping the Growing Stem Volume of the Coniferous Plantations in North China Using Multispectral Data from Integrated GF-2 and Sentinel-2 Images and an Optimized Feature Variable Selection Method. *Remote Sens.* **2021**, *13*, 2740. [CrossRef]
4. Luoma, V.; Yrttimaa, T.; Kankare, V.; Saarinen, N.; Pyörälä, J.; Kukko, A.; Kaartinen, H.; Hyyppä, J.; Holopainen, M.; Vastaranta, M. Revealing Changes in the Stem Form and Volume Allocation in Diverse Boreal Forests Using Two-Date Terrestrial Laser Scanning. *Forests* **2021**, *12*, 835. [CrossRef]
5. Shugart, H.H.; Saatchi, S.; Hall, F.G. Importance of structure and its measurement in quantifying function of forest ecosystems. *J. Geophys. Res. Biogeosci.* **2010**, *115*, G00E13. [CrossRef]
6. Zheng, J.; Tarin, M.W.K.; Jiang, D.; Li, M.; Ye, J.; Chen, L.; He, T.; Zheng, Y. Which ornamental features of bamboo plants will attract the people most? *Urban For. Urban Green.* **2021**, *61*, 127101. [CrossRef]
7. Burrascano, S.; Lombardi, F.; Marchetti, M. Old-growth forest structure and deadwood: Are they indicators of plant species composition? A case study from central Italy. *Plant Biosyst.* **2008**, *142*, 313–323. [CrossRef]
8. Shan, T.; Englot, B.; Meyers, D.; Wang, W.; Ratti, C.; Rus, D. LIO-SAM: Tightly-coupled Lidar Inertial Odometry via Smoothing and Mapping. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 25–29 October 2020; pp. 5135–5142.
9. Shan, T.; Englot, B. LeGO-LOAM: Lightweight and Ground-Optimized Lidar Odometry and Mapping on Variable Terrain. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 4758–4765.
10. Shan, T.; Englot, B.; Ratti, C.; Rus, D. LVI-SAM: Tightly-coupled Lidar-Visual-Inertial Odometry via Smoothing and Mapping. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, May 30–June 5 2021; pp. 5692–5698.
11. Wang, C.; Zhang, G.; Zhang, M. Research on improving LIO-SAM based on Intensity Scan Context. *J. Phys. Conf. Ser.* **2021**, *1827*, 012193. [CrossRef]
12. Wang, P.; Wang, L.G.; Leung, H.; Zhang, G. Super-Resolution Mapping Based on Spatial–Spectral Correlation for Spectral Imagery. *IEEE Trans. Geosci. Remote Sens.* **2021**, *59*, 2256–2268. [CrossRef]

13. Liu, J.; Wu, Y.; Gao, X.; Zhang, X. A Simple Method of Mapping Landslides Runout Zones Considering Kinematic Uncertainties. *Remote Sens.* **2022**, *14*, 668. [CrossRef]

14. Jacobs, M.; Rais, A.; Pretzsch, H. How drought stress becomes visible upon detecting tree shape using terrestrial laser scanning (TLS). *For. Ecol. Manag.* **2021**, *489*, 118975. [CrossRef]

15. O'Sullivan, H.; Raumonen, P.; Kaitaniemi, P.; Perttunen, J.; Sievanen, R. Integrating terrestrial laser scanning with functional-structural plant models to investigate ecological and evolutionary processes of forest communities. *Ann. Bot.* **2021**, *128*, 663–683. [CrossRef] [PubMed]

16. Muumbe, T.P.; Baade, J.; Singh, J.; Schmullius, C.; Thau, C. Terrestrial Laser Scanning for Vegetation Analyses with a Special Focus on Savannas. *Remote Sens.* **2021**, *13*, 507. [CrossRef]

17. Ko, C.; Lee, S.; Yim, J.; Kim, D.; Kang, J. Comparison of Forest Inventory Methods at Plot-Level between a Backpack Personal Laser Scanning (BPLS) and Conventional Equipment in Jeju Island, South Korea. *Forests* **2021**, *12*, 308. [CrossRef]

18. Jafri, S.R.u.N.; Rehman, Y.; Faraz, S.M.; Amjad, H.; Sultan, M.; Rashid, S.J. Development of Georeferenced 3D Point Cloud in GPS Denied Environments Using Backpack Laser Scanning System. *Elektronika ir Elektrotechnika* **2021**, *27*, 25–34. [CrossRef]

19. Roy, K.; Chaudhuri, S.S.; Pramanik, S. Deep learning based real-time Industrial framework for rotten and fresh fruit detection using semantic segmentation. *Microsyst. Technol.* **2021**, *27*, 3365–3375. [CrossRef]

20. Wu, H.; Liang, C.; Liu, M.; Wen, Z. Optimized HRNet for image semantic segmentation. *Expert Syst. Appl.* **2021**, *174*, 114532. [CrossRef]

21. Kim, W.S.; Lee, D.H.; Kim, T.; Kim, H.; Sim, T.; Kim, Y.J. Weakly Supervised Crop Area Segmentation for an Autonomous Combine Harvester. *Sensors* **2021**, *21*, 4801. [CrossRef]

22. Zhang, Y.; Lu, Z.; Zhang, X.; Xue, J.-H.; Liao, Q. Deep Learning in Lane Marking Detection: A Survey. In *IEEE Transactions on Intelligent Transportation Systems*; IEEE: Piscataway, NJ, USA, 2021; pp. 5976–5992.

23. Zhang, M.; Li, Z.; Wu, X. Semantic Segmentation Method Accelerated Quantitative Analysis of the Spatial Characteristics of Traditional Villages. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2021**, *46*, 933–939. [CrossRef]

24. Boulch, A.; Guerry, J.; Le Saux, B.; Audebert, N. SnapNet: 3D point cloud semantic labeling with 2D deep segmentation networks. *Comput. Graph.* **2018**, *71*, 189–198. [CrossRef]

25. Tchapmi, L.; Choy, C.; Armeni, I.; Gwak, J.; Savarese, S. SEGCloud: Semantic Segmentation of 3D Point Clouds. In Proceedings of the 2017 International Conference on 3D Vision (3DV), Qingdao, China, 10–12 October 2017; pp. 537–547.

26. Charles, R.Q.; Su, H.; Kaichun, M.; Guibas, L.J. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 77–85.

27. Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. PointNet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv* **2017**, arXiv:1706.02413.

28. Chen, X.; Jiang, K.; Zhu, Y.; Wang, X.; Yun, T. Individual Tree Crown Segmentation Directly from UAV-Borne LiDAR Data Using the PointNet of Deep Learning. *Forests* **2021**, *12*, 131. [CrossRef]

29. Krisanski, S.; Taskhiri, M.S.; Gonzalez Aracil, S.; Herries, D.; Turner, P. Sensor Agnostic Semantic Segmentation of Structurally Diverse and Complex Forest Point Clouds Using Deep Learning. *Remote Sens.* **2021**, *13*, 1413. [CrossRef]

30. Guinard, S.; Landrieu, L. Weakly supervised segmentation-aided classification of urban scenes from 3D lidar point clouds. In Proceedings of the ISPRS Workshop 2017, Hannover, Germany, 6–9 June 2017.

31. Landrieu, L.; Simonovsky, M. Large-Scale Point Cloud Semantic Segmentation with Superpoint Graphs. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4558–4567.

32. Li, Y.; Bu, R.; Sun, M.; Wu, W.; Di, X.; Chen, B. Pointcnn: Convolution on x-transformed points. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 828–838.

33. Han, T.; Sánchez-Azofeifa, G.A. A Deep Learning Time Series Approach for Leaf and Wood Classification from Terrestrial LiDAR Point Clouds. *Remote Sens.* **2022**, *14*, 3157. [CrossRef]

34. Krůček, M.; Král, K.; Cushman, K.C.; Missarov, A.; Kellner, J.R. Supervised segmentation of ultra-high-density drone lidar for large-area mapping of individual trees. *Remote Sens.* **2020**, *12*, 3260. [CrossRef]

35. Raumonen, P.; Casella, E.; Calders, K.; Murphy, S.; Åkerblom, M.; Kaasalainen, M. Massive-Scale Tree Modelling from Tls Data. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2015**, *2*, 189–196. [CrossRef]

36. Calders, K.; Newnham, G.; Burt, A.; Murphy, S.; Raumonen, P.; Herold, M.; Culvenor, D.; Avitabile, V.; Disney, M.; Armston, J.; et al. Nondestructive estimates of above-ground biomass using terrestrial laser scanning. *Methods Ecol. Evol.* **2014**, *6*, 198–208. [CrossRef]

37. Raumonen, P.; Kaasalainen, M.; Åkerblom, M.; Kaasalainen, S.; Kaartinen, H.; Vastaranta, M.; Holopainen, M.; Disney, M.; Lewis, P. Fast Automatic Precision Tree Models from Terrestrial Laser Scanner Data. *Remote Sens.* **2015**, *5*, 491–520. [CrossRef]

38. Markku, Å.; Raumonen, P.; Kaasalainen, M.; Casella, E. Analysis of Geometric Primitives in Quantitative Structure Models of Tree Stems. *Remote Sens.* **2015**, *7*, 4581–4603. [CrossRef]

39. Ye, X.; Li, J.; Huang, H.; Du, L.; Zhang, X. 3d Recurrent Neural Networks with Context Fusion for Point Cloud Semantic Segmentation. In Proceedings of the Computer Vision—ECCV 2018—15th European Conference, Munich, Germany, 8–14 September 2018.

40. Cai, J.-X.; Mu, T.-J.; Lai, Y.-K.; Hu, S.-M. LinkNet: 2D-3D linked multi-modal network for online semantic segmentation of RGB-D videos. *Comput. Graph.* **2021**, *98*, 37–47. [CrossRef]

41. Qiu, S.; Anwar, S.; Barnes, N. Semantic Segmentation for Real Point Cloud Scenes via Bilateral Augmentation and Adaptive Fusion. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Virtual, 19–25 June 2021; pp. 1757–1767.

42. Manduchi, G. Commonalities and differences between MDSplus and HDF5 data systems. *Fusion Eng. Des.* **2010**, *85*, 583–590. [CrossRef]

43. Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; et al. TensorFlow: A System for Large-Scale Machine Learning. In Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation, OSDI 2016, Savannah, GA, USA, 2–4 November 2016; Volume 16, pp. 265–283.

44. Weinmann, M.; Urban, S.; Hinz, S.; Jutzi, B.; Mallet, C. Distinctive 2D and 3D features for automated large-scale scene analysis in urban areas. *Comput. Graph.* **2015**, *49*, 47–57. [CrossRef]

45. Demantké, J.; Mallet, C.; David, N.; Vallet, B. Dimensionality Based Scale Selection in 3D LiDAR Point Clouds. ISPRS Int. Arch. Photogramm. *Remote Sens. Spat. Inform. Sci.* **2012**, *XXXVIII-5/W12*, 97–102.

46. Landrieu, L.; Obozinski, G. Cut pursuit: Fast algorithms to learn piecewise constant functions on general weighted graphs. *SIAM J. Imaging Sci.* **2017**, *10*, 1724–1766. [CrossRef]

47. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Munich, Germany, 5–9 October 2015; pp. 234–241.

48. Windrim, L.; Bryson, M. Detection, Segmentation, and Model Fitting of Individual Tree Stems from Airborne Laser Scanning of Forests Using Deep Learning. *Remote Sens.* **2020**, *12*, 1469. [CrossRef]

49. Wang, D.; Momo Takoudjou, S.; Casella, E. Chisholm, Ryan LeWoS: A universal leaf-wood classification method to facilitate the 3D modelling of large tropical trees using terrestrial LiDAR. *Methods Ecol. Evol.* **2020**, *11*, 376–389. [CrossRef]

50. Li, Y.; Tong, G.; Li, X.; Zhang, L.; Peng, H. MVF-CNN: Fusion of Multilevel Features for Large-Scale Point Cloud Classification. *IEEE Access* **2019**, *7*, 46522–46537. [CrossRef]

51. Yun, T.; Cao, L.; An, F.; Chen, B.; Xue, L.; Li, W.; Pincebourde, S.; Smith, M.J.; Eichhorn, M.P. Simulation of multi-platform LiDAR for assessing total leaf area in tree crowns. *Agric. For. Meteorol.* **2019**, *276*, 107610. [CrossRef]

52. Sun, C.; Huang, C.; Zhang, H.; Chen, B.; An, F.; Wang, L.; Yun, T. Individual Tree Crown Segmentation and Crown Width Extraction From a Heightmap Derived From Aerial Laser Scanning Data Using a Deep Learning Framework. *Front. Plant Sci.* **2022**, *13*, 914974.. [CrossRef]

53. Wang, J.; Chen, X.; Cao, L.; An, F.; Chen, B.; Xue, L.; Yun, T. Individual Rubber Tree Segmentation Based on Ground-Based LiDAR Data and Faster R-CNN of Deep Learning. *Forests* **2019**, *10*, 793. [CrossRef]