forests

MDPI

Article

Research on Single-Tree Segmentation Method for Forest 3D Reconstruction Point Cloud Based on Attention Mechanism

Lishuo Huo [1,2,3], Zhao Chen [1,2,3,*], Lingnan Dai [1,2,3], Dianchang Wang [1,2,3] and Xinrong Zhao [1,2,3]

[1] School of Information Science and Technology, Beijing Forestry University, Beijing 100083, China; hls_0851@163.com (L.H.); daidaidai0426@163.com (L.D.); 17864286091@163.com (D.W.); zxr7230428@bjfu.edu.cn (X.Z.)
[2] Engineering Research Center for Forestry-Oriented Intelligent Information Processing, National Forestry and Grassland Administration, Beijing 100083, China
[3] Hebei Key Laboratory of Smart National Park, Beijing 100083, China
* Correspondence: cz71@bjfu.edu.cn; Tel.: +86-13641197031

Abstract

The segmentation of individual trees holds considerable significance in the investigation and management of forest resources. Utilizing smartphone-captured imagery combined with image-based 3D reconstruction techniques to generate corresponding point cloud data can serve as a more accessible and potentially cost-efficient alternative for data acquisition compared to conventional LiDAR methods. In this study, we present a Sparse 3D U-Net framework for single-tree segmentation which is predicated on a multi-head attention mechanism. The mechanism functions by projecting the input data into multiple subspaces—referred to as "heads"—followed by independent attention computation within each subspace. Subsequently, the outputs are aggregated to form a comprehensive representation. As a result, multi-head attention facilitates the model's ability to capture diverse contextual information, thereby enhancing performance across a wide range of applications. This framework enables efficient, intelligent, and end-to-end instance segmentation of forest point cloud data through the integration of multi-scale features and global contextual information. The introduction of an iterative mechanism at the attention layer allows the model to learn more compact feature representations, thereby significantly enhancing its convergence speed. In this study, Dongsheng Bajia Country Park and Jiufeng National Forest Park, situated in Haidian District, Beijing, China, were selected as the designated test sites. Eight representative sample plots within these areas were systematically sampled. Forest stand sequential photographs were captured using an iPhone, and these images were processed to generate corresponding point cloud data for the respective sample plots. This methodology was employed to comprehensively assess the model's capability for single-tree segmentation. Furthermore, the generalization performance of the proposed model was validated using the publicly available dataset TreeLearn. The model's advantages were demonstrated across multiple aspects, including data processing efficiency, training robustness, and single-tree segmentation speed. The proposed method achieved an F1 score of 91.58% on the customized dataset. On the TreeLearn dataset, the method attained an F1 score of 97.12%.

Keywords: point cloud; individual tree segmentation; multi-head attention; sparse convolutional; 3D U-Net

check for updates

Received: 11 June 2025
Revised: 16 July 2025
Accepted: 17 July 2025
Published: 19 July 2025

Citation: Huo, L.; Chen, Z.; Dai, L.; Wang, D.; Zhao, X. Research on Single-Tree Segmentation Method for Forest 3D Reconstruction Point Cloud Based on Attention Mechanism. *Forests* **2025**, *16*, 1192. https://doi.org/10.3390/f16071192

Copyright: © 2025 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

Surveys of forest resources are crucial for maintaining ecological balance, mitigating climate change, and ensuring the sustainable management of forest ecosystems [1]. In order to assess the structural characteristics of a forest or a single tree, LiDAR is currently a mainstream tool for acquiring 3D point cloud models of forests [2]. However, the equipment is costly and not very portable, and the required postprocessing process for the acquired point cloud data is cumbersome and time-consuming [3]. With the improvement in the accuracy of 3D reconstruction technology, image-based 3D reconstruction technology provides a low-cost way to acquire data [4]. Therefore, 3D reconstruction data will become another data source for sample plot investigation. The single-tree segmentation algorithm can separate a point cloud into single trees, which makes it easier to obtain the spatial coordinates, canopy structure model, and health of the trees [5]. Meanwhile, as the challenges and costs associated with data acquisition decline, the availability of relevant open-source datasets is expected to rise, thereby facilitating advancements in single-tree segmentation accuracy.

Presently, single-tree segmentation techniques utilizing point cloud data are primarily categorized into two distinct classes: algorithms based on manual feature extraction and those based on automatic feature extraction [6]. Among these, algorithms reliant on manual feature extraction necessitate the laborious extraction and input of numerous structural parameters of trees, thereby increasing the time and complexity associated with data preprocessing [7]. Currently, two primary algorithms employing manual feature extraction exist: The first involves determining the position of the apex of a tree crown using the Canopy Height Model (CHM), followed by the application of a segmentation algorithm (e.g., a watershed algorithm [8]) to isolate individual trees. Chen et al. [9] employed a marker-controlled watershed algorithm to detect and extract single-tree information from fruit trees, demonstrating that the watershed algorithm applied to the Canopy Height Model exhibits high feasibility in addressing issues related to overlap and occlusion within tree canopies. Additionally, Chen et al. [10] proposed a similar marker-controlled watershed algorithm to segment individual trees by identifying the apex of a tree through a variable-sized dynamic window. The other algorithm is single-tree segmentation directly based on the original point cloud data, which can directly utilize the original point cloud data without constructing additional 3D models, thus reducing the complexity of data processing [11]. This category primarily encompasses point cloud distance clustering discriminant algorithms [6], k-means clustering algorithms [12], and others. Jiang et al. [13] performed a sensitivity analysis on the point cloud distance discriminative clustering algorithm, revealing that setting the distance threshold to the average crown radius of the sample site yields optimal single-tree segmentation. In addition, Vega et al. [14] introduced a multi-scale dynamic point cloud segmentation method which utilizes diverse evaluation criteria to ascertain the optimal tree-top location as the initial clustering center for crown clustering, achieving a correct segmentation rate of 82%.

Automatic feature extraction methods typically leverage deep learning techniques, which autonomously extract features via neural networks, significantly diminishing the reliance on manual feature extraction [15]. Moreover, this method can autonomously complete segmentation upon the conclusion of model training, eliminating the necessity for further personnel involvement, thereby minimizing redundant operations and optimizing the overall workflow [15]. Currently, this category comprises two primary methodological approaches: The first involves employing deep learning techniques (e.g., PointNet [16] and PointNet++ [17]) to conduct semantic segmentation, effectively isolating the desired tree species from the dataset, followed by manual feature extraction for single-tree segmentation. While this approach does enhance the accuracy of single-tree segmentation, it merely employs deep learning as a data-cleaning tool, failing to capitalize on deep learning's

potential for feature extraction and neglecting to fully demonstrate its advantages, including automatic feature extraction and end-to-end processing capabilities. For instance, Chen [18] employed the PointNet network model for tree identification on voxelized data. Following the acquisition of classification results for each voxel point cloud, the segmentation of the tree-crown boundary was subsequently refined based on highly correlated gradient information. Another category of methodologies accomplishes single-tree segmentation by advancing instance segmentation models within the deep learning paradigm. Typically, this approach utilizes neural networks to extract salient data features, which are subsequently employed to perform clustering or generate predictive frames, thereby facilitating the precise segmentation of individual instances.

For example, Henrich et al. [19] introduced a deep learning-based single-tree segmentation approach, incorporating enhancements to the PointGroup model. This technique utilizes an end-to-end training strategy to extract multi-scale geometric features from point cloud data and performs clustering based on these features. Consequently, this results in a substantial enhancement of the accuracy in single-tree segmentation.

Nonetheless, deep learning models typically demand extensive labeled datasets for training and entail prolonged training durations, thereby presenting notable challenges for real-world applications [20].

The applicability of these methodologies to three-dimensional reconstructed datasets warrants further exploration and rigorous validation. This is attributed to the fact that point cloud data generated via 3D reconstruction display a higher density and more heterogeneous spatial distributions relative to the original LiDAR-acquired point clouds.

The Sparse 3D U-Net architecture represents a novel variant of the U-Net framework originally introduced by Ronneberger et al. [21]. The original U-Net architecture features a symmetric encoder–decoder structure that effectively captures the contextual information of an image while simultaneously preserving intricate details. Consequently, this architecture demonstrates superior performance in medical image segmentation tasks, particularly when the available dataset is limited. Nonetheless, it possesses the potential to lose critical information during the processing of 3D data, thereby failing to fully exploit the spatial dimensions. Çiçek et al. [22] introduced the 3D U-Net by extending the convolution operation into three dimensions, employing both 3D convolution and 3D pooling operations, thereby enabling the network to capture more intricate spatial features. Furthermore, the Sparse 3D U-Net is a framework introduced by Sun et al. [23] which builds upon the principles of the 3D U-Net integrated with sparse convolution techniques proposed by Graham et al. [24]. This framework significantly enhances computational efficiency and memory utilization while preserving the spatial structural features of 3D data, thereby improving segmentation performance. However, due to its high network complexity and dependence on the sparsity of the data, it may lead to slow convergence of the network.

In the realm of deep learning, attention mechanisms have been extensively employed across a variety of tasks [25]. The fundamental attention mechanism allows the model to concentrate on the most pertinent information pertaining to the current task through the computation of the weighted sum of various segments of the input sequence [26]. Nevertheless, despite its ability to capture contextual information, the conventional single-head attention mechanism is limited to aggregating information within a subspace and may fail to encapsulate multiple distinct semantic features. In contrast, multi-head attention seeks to address this limitation by concurrently computing several attention heads in parallel [26]. By partitioning the input representation into multiple linearly transformed subspaces, this approach enables each attention head to independently compute the attention weights; subsequently, the outputs of each head are concatenated and linearly transformed to yield the final output [26]. While the attention mechanism has the capacity to enhance model perfor-

mance, relying solely on this mechanism may not fully leverage the structural information inherent within the data.

To address the above key challenges in practical applications, this study proposes a deep learning model based on the attention mechanism. This model effectively integrates single-tree segmentation with deep learning techniques and incorporates an attention mechanism for algorithmic optimization grounded in the U-Net network architecture, thereby enhancing the feature extraction capabilities and training speed of the model. Notably, this approach is end-to-end, requiring no professional intervention to operate the model once training is complete. The segmented data can be automatically outputted by simply inputting the forest data intended for segmentation into the model. In the experiments, not only the performance of our method and several existing methods on LiDAR point cloud datasets, but also the applicability of these methods to 3D reconstruction datasets is verified. The results show that our method not only has good performance on the LiDAR point cloud dataset, but also performs well on the 3D reconstruction dataset and is still able to output high-quality single-tree segmentation results. The specific contributions are as follows:

1. The Sparse 3D U-Net and multi-head attention mechanism techniques are introduced into the domain of single-tree segmentation for the first time, thereby broadening the application prospects of deep learning in this area.
2. A novel model architecture is proposed to extract both the spatial and offset information of points within point cloud data using the Sparse 3D U-Net. The offset information refers to the difference between the projected plane and the original coordinates. Subsequently, the multi-head attention mechanism is employed to separately aggregate this information, culminating in the use of an iterative approach to expedite the convergence of the model. These innovative designs effectively address the challenges posed by the excessive density and uneven distribution of point clouds in data acquired through image-based 3D reconstruction methods, thereby significantly enhancing both the efficiency and accuracy of single-tree segmentation.
3. We investigated the feasibility of an efficient and cost-effective forest inventory methodology, circumventing the reliance on expensive equipment. Furthermore, we executed single-tree segmentation in two representative forest scenarios—a nature reserve in North China and an urban ecological green space—utilizing solely image-generated point cloud data. A comprehensive series of experiments, including ablation and comparative studies, was conducted, thereby thoroughly validating the efficacy and superiority of our model with respect to single-tree segmentation.

2. Dataset and Evaluation Methods

2.1. Dataset

To assess the robustness of the model, this study initially performed comprehensive experiments utilizing the open-source forest point cloud dataset TreeLearn, followed by an evaluation of the model's generalization capabilities on small-scale samples derived from a custom dataset generated through 3D reconstruction. It is worth noting that the two datasets belong to different research regions.

2.1.1. Study Area

The open-source forest point cloud dataset, TreeLearn, employed in this study comprises 19 tree plot-level point clouds sourced from Germany. For further details regarding this dataset, readers are referred to the pertinent original study [19]. Additionally, Dongsheng Bajia Country Park and Jiufeng National Forest Park, located in the Haidian District of Beijing, China, were designated as the study areas for this research. The geographic coor-

dinates of Dongsheng Bajia Country Park are delineated between 116.2815° and 116.2958° E and 39.9712° and 39.9795° N. The elevation within the park varies between 40 m and 60 m. The predominant tree species within the park comprise poplar, willow, locust, and pine trees, collectively representing a typical urban plantation forest ecosystem. The geographic coordinates of Jiufeng National Forest Park extend from 116.1134° to 116.1419° E longitude and from 39.9971° to 40.0216° N latitude. This region is characterized by a complex topography and considerable elevation variation, ranging from 100 m to 1153 m above sea level. The predominant tree species in this park include greasy pine, larch, ginkgo, and oak forests, collectively illustrating a typical nature reserve in North China. The data collection period for this study spanned March and August 2024, during which the weather conditions were predominantly sunny and aligned with the spring season, contributing to a relatively stable climate that facilitated the accurate acquisition of image data pertaining to the forest landscape (Figure 1).

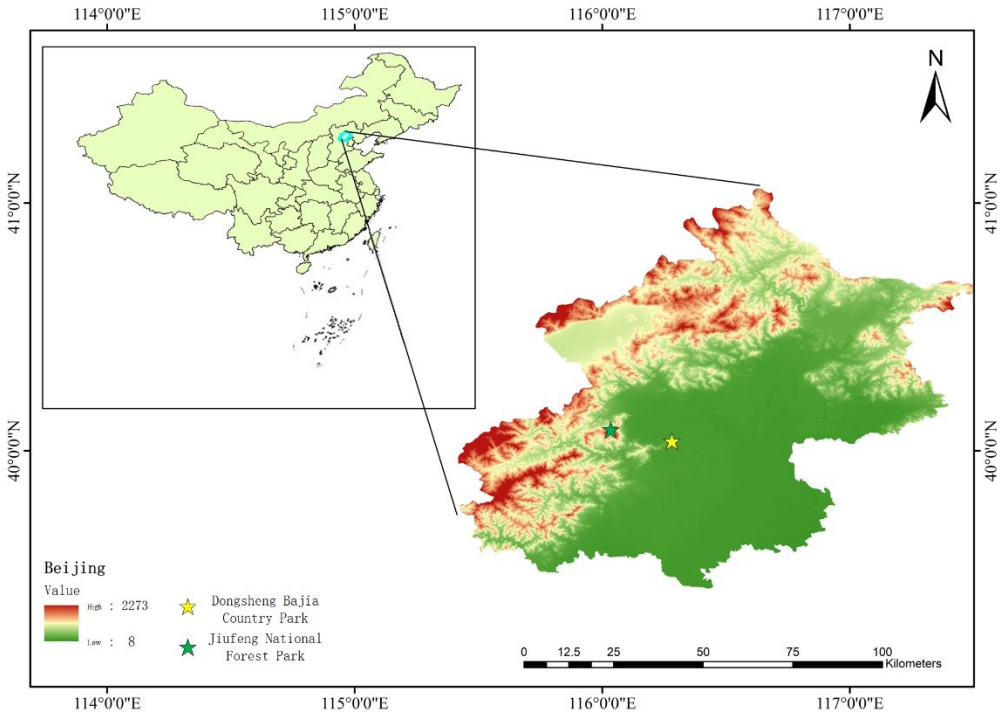


Figure 1. Overview of the study area [27].

2.1.2. Data Acquisition

The datasets employed for model training and validation in this study comprised the open-source forest point cloud dataset TreeLearn, as well as a customized dataset. Eight sample plots measuring 15 m × 15 m were selected for analysis within the customized dataset. Four of these plots were situated in Dongsheng Bajia Country Park, characterized by relatively flat terrain, uniform tree heights, and moderate tree overlap. Close-up photography was employed to accurately capture tree height information, given the open terrain and minimal shading from surrounding trees. The remaining four plots were situated within Jiufeng National Forest Park, two of which featured steep terrain and significant variations in tree heights. Using an iPhone 15 smartphone, we circled around the forest from the periphery toward the interior, gradually walking to the plot's center. Subsequently, we performed a controlled, small circular movement from the inside to the outside to record a video at a consistent speed. Afterwards, a custom script was developed to extract frame indices from the video, remove redundant and blurry frames, and ultimately generate a sequence of images representing the sample plot. Following the completion of data collec-

tion, the eight sample plots were modeled utilizing the 3D reconstruction methodology proposed by Dai et al. [27], ultimately yielding the point cloud data for these plots (Table 1).

Table 1. Introduction to the sample plot.

Location	Sample Plot	Principal Species	Number of Trees	Type
Dongsheng Bajia Country Park	1	Poplar	25	Planted forest
	2	Poplar	23	
	3	Poplar	28	
	4	Poplar	21	
Jiufeng National Forest Park	5	Red pine	35	Natural forests
	6	Red pine	29	
	7	Red pine	30	
	8	Red pine	26	

3. Methods

This study proposes a deep learning-based model for single-tree segmentation, comprising three distinct modules: a Sparse 3D U-Net, an attention layer, and a prediction head, utilizing forest point cloud data as input [28]. Each of the three modules performs a unique function and collaboratively operates to achieve single-tree segmentation. Initially, the Sparse 3D U-Net is employed to extract bottom-up, point-by-point features from the point cloud data. Subsequently, the attention layer incorporates a multi-head attention mechanism, a self-attention mechanism, a feedforward neural network, and a learnable query vector to augment the model's global perceptual and semantic parsing capabilities. Ultimately, two independent Multi-Layer Perceptron (MLP) networks are utilized within the prediction head section to forecast both semantic information and offset information, respectively.

Specifically, the first phase is the preparation phase, wherein the input forest point cloud data are divided into smaller overlapping rectangular slices based on the x and y coordinates to mitigate memory constraints; the second phase is the feature extraction phase, during which point-by-point features are extracted using the Sparse 3D U-Net network and subsequently fed into the next layer; following this, the feature enhancement phase involves the attention layer performing relational modeling based on the features extracted in the previous layer. The extracted features facilitate relationship modeling and feature enhancement; this is followed by the semantic prediction and offset prediction phases, wherein the prediction head outputs the semantic prediction results and offset prediction results based on the outputs from the preceding layer. Subsequently, leveraging the semantic predictions and offset predictions, points sharing identical semantics are projected onto the same plane for clustering, and the slices are merged; finally, unallocated points from the previous step are assigned to their nearest instances. The flow of the specific session is illustrated in Figure 2. It should be noted that the core of this study lies in the design of the proposed network's model architecture. Unlike the model adopted by TreeLearn, we introduced an attention mechanism into the model to achieve dynamic and flexible feature selection and reweighting, which effectively improves the performance of the model. In addition, without any modification to the original framework, we directly applied the network model proposed in this study to the segmentation pipeline proposed by TreeLearn to accomplish the corresponding segmentation tasks.

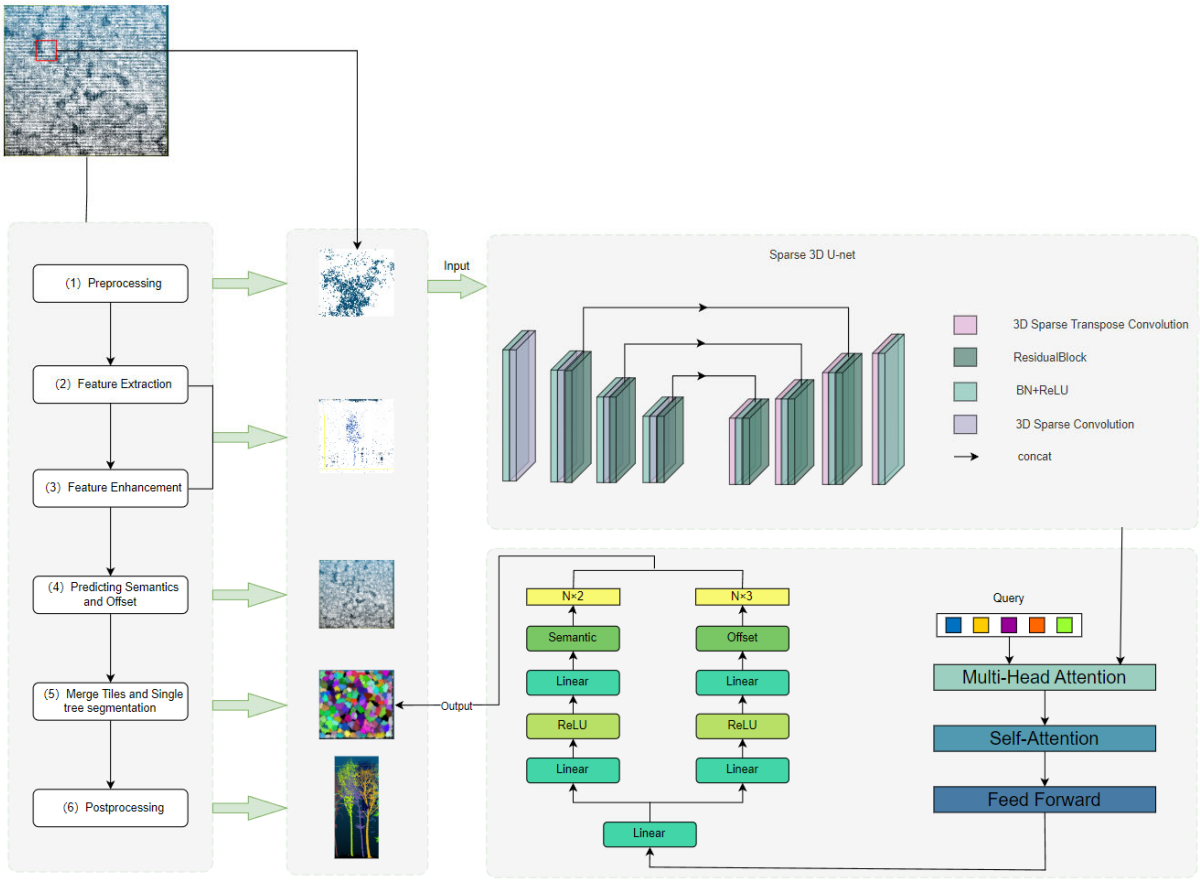


Figure 2. Flowchart of method.

3.1. Data Preprocessing

The TreeLearn dataset encompasses a point cloud representing the entire forest, comprising approximately 20 million points. Owing to memory constraints, directly processing point cloud data of this magnitude with neural networks is not feasible. Drawing inspiration from the work of Ronneberger et al. [21], this study employed a slicing approach to address this challenge. The degree of overlap between neighboring slices can be regulated by adjusting the hyperparameters. This treatment was also used for custom datasets.

In the customized dataset, this study employed 3D reconstruction techniques to generate the corresponding point cloud data from the acquired image data, followed by point cloud filtering to mitigate the interference of isolated noise during feature extraction [29]. Subsequently, the data underwent normalization relative to its coordinates, and the point cloud was aligned with the origin through translation and rotation. This process mitigates the interference of absolute coordinates on the deep learning model and enhances the algorithm's robustness against scale variations. Finally, the data were manually segmented utilizing CloudCompare(version 2.13), with seven of the samples designated as the training set and the remaining sample allocated as the test set.

3.2. Sparse 3D U-Net

To address the significant computational burden associated with traversing all spatial locations when processing sparse data (such as point cloud data) using traditional convolution methods, this study adopted the approach delineated by Graham et al. [24], who developed a U-Net architecture incorporating sub-stream form sparse convolution (SSC) and sparse convolution (SC). This module serves as the cornerstone of the model, tasked with extracting point-wise and offset features from the point cloud data. Let us assume that

the input point cloud comprises N points, denoted as P∈RN×3, where each point possesses three coordinate attributes: x, y, and z. Subsequently, the input point cloud undergoes voxelization, after which the voxelized point cloud is input into the Sparse 3D U-Net network. The encoder component of this network primarily comprises four 3D sparse convolutions, which are chiefly responsible for downsampling the data, whereas the decoder component predominantly consists of four 3D sparse transpose convolutions tasked with upsampling the data. Additionally, batch normalization and ReLU activation functions are applied following all convolution operations, thereby enhancing the stability of model training. Furthermore, following each ReLU activation, a ResidualBlock is integrated, providing a direct pathway for the gradient to propagate through the network. Concurrently, the residual connections enable the encoder to capture features at varying scales, while the decoder fuses low- and high-level features owing to the skip connections. The extraction of pointwise features, denoted as P∗∈RN×C, is ultimately achieved. The specific structure of this framework is illustrated in Figure 2.

3.3. Attention Layer

The attention module predominantly comprises multi-head attention, self-attention, a feedforward neural network (FFN), and a learnable query vector, which constitutes a pivotal component of the model [30]. Its primary objective is to iteratively refine the query features while enhancing the multimodal information fusion capabilities through the attention mechanism. Among these components, the multi-head attention layer and the self-attention layer serve as the core modules facilitating the internal interaction of query vectors, employed to uncover long-range dependencies within the query sequence.

$$\text{Attention}(\mathrm{Q}, \mathrm{K}, \mathrm{V}) = \left(\frac{QK^T}{\sqrt{d_k}}\right)\mathrm{V} \tag{1}$$

where Q, K, and V are derived from query vectors through linear mapping and d_k represents the deflation factor. By employing multi-head parallel computation, the model effectively captures features from diverse subspaces. Upon completion of the attentional interaction, the features undergo nonlinear augmentation via a feedforward neural network (FFN). The FFN comprises two tiers of fully connected layers, each equipped with ReLU activation functions:

$$\text{FFN}(\mathrm{x}) = \mathrm{W}_2(\text{ReLU}(\mathrm{W}_1\mathrm{x} + \mathrm{b}_1)) + \mathrm{b}_2 \tag{2}$$

where W_1 and W_2 represent the learnable weights, while b_1 and b_2 denote the learnable bias vectors. The feedforward neural network (FFN) further amplifies the expressive capacity of the model and integrates residual connections along with normalization operations subsequent to each layer. The module takes as input point-by-point features and learnable query vectors such that each query vector corresponds to a potential instance prediction based on point-by-point features. Moreover, the query vectors progressively learn the feature templates and distributions of diverse instances within the space via the multi-layer attention mechanism. The specific architecture is illustrated in Figure 3.

Furthermore, during the training phase, it was observed that incorporating certain true values into the outputs of the attention layer can enhance the model's performance. Let us denote the presence of M query vectors, defining the query vector as $Z_\theta \in R^{M\times D}$, where D represents the embedding dimension and $\theta = 0, 1, 2, 3, \cdots$ denotes the index. Here, Z0 signifies the query vector initialized at the outset, while $Z_\theta = 1, 2, 3, \cdots$ corresponds to the feature query vectors that are learned to represent various instance types following the training process. Subsequently, Z1 is reintroduced into the attention layer, and after numer-

ous iterations, only the output from the final attention layer is utilized in the prediction head for generating predictions. The formula is presented as follows:

$$\widehat{Z_\theta} = softmax\left(\frac{QK^T}{\sqrt{D}}\right)\mathrm{V} \tag{3}$$

where $\widehat{Z_\theta}$ denotes the output of the attention layer and Q represents a linear projection of $Z_{\theta-1}$, while K and V correspond to linear projections capturing distinct features. Empirical evidence confirms that this approach facilitates an enhancement in the rate of model convergence when addressing a variable number of instance projections.

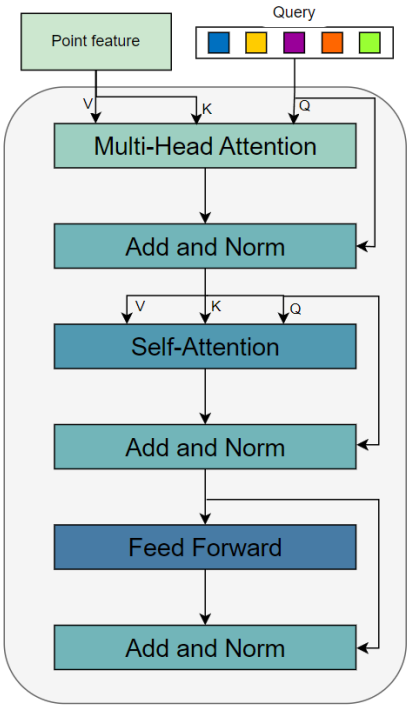


Figure 3. The architecture of the attention layer.

3.4. Prediction Head

The prediction head primarily functions to generate the final predictions based on the query features produced by the preceding layer, encompassing both semantic labeling and offset prediction. In the specific implementation, the input query features undergo layer normalization as an initial step to enhance training stability. Subsequently, the normalized features are mapped onto the semantic category space via a fully connected layer, which produces the target category prediction labels. Concurrently, an additional fully connected layer is employed to derive bias values. Furthermore, instance features are obtained by analyzing the relationships between query features and point-by-point features. Ultimately, the output of the prediction header amalgamates semantic information, bias information, and instance features to furnish dependable support for subsequent tasks.

3.5. Merge Tiles and Single-Tree Segmentation

Upon completion of the predictions for all slices, the slices are amalgamated according to a tailored overlap region. For multiple predictions generated in overlapping sections, the average is computed, thereby minimizing artifacts introduced by the tiling process. Should the offset predictions prove accurate, the coordinates projected from the offset values will yield x and y coordinates that correspond to the trunk. These coordinates will subsequently

serve as inputs to a density-based clustering algorithm, which executes the final single-tree segmentation [31].

3.6. Postprocessing

In the preceding step, the tree instances are identified; however, a subset of points situated at the peripheries of the clusters remains unassigned. To effectively assign these points, a straightforward strategy is employed: calculating their distances from neighboring points based on the projected coordinates, followed by assigning them to the instances corresponding to the nearest point according to the nearest neighbor principle.

3.7. Assessment Criteria

To assess the performance of the proposed methodology, the matching strategy delineated by Zhao et al. [32] was employed, which correlates the true values with the predicted values, as illustrated in the following equation:

$$\mathrm{D} = \sqrt{\left((x_i - x_j)^2 + (y_i - y_j)^2 + k(z_i - z_j)^2\right)} \tag{4}$$

where D represents the distance between the true and predicted values, while (x_i, y_i, z_i) and (x_j, y_j, z_j) denote the trunk coordinates corresponding to the true and predicted values, respectively. Estimated trees i and j are deemed successfully matched when the two three-dimensional spatial points that represent the trees' location information are in close proximity. k signifies the weight attributed to the height difference, with a default value set at 0.5.

In evaluating the results of semantic segmentation, this study adopted accuracy, defined as the ratio of correctly predicted points to the total number of points. To mitigate potential bias in the results arising from the over-weighting of regions characterized by high point densities, this study employed point cloud data that had been secondarily sampled using voxels of size 10 cm^3 for evaluation.

In assessing the results of instance segmentation, this study utilized recall, precision, and F1 scores to evaluate the performance of the proposed methodology, as demonstrated in the following equation:

$$\mathrm{r} = \frac{TP}{TP + FN} \tag{5}$$

$$\mathrm{p} = \frac{TP}{TP + FP} \tag{6}$$

$$\mathrm{F1} = 2\frac{p \times r}{p + r} \tag{7}$$

where TP denotes that the true value and the predicted value align accurately, indicating that the instance was successfully detected; FN signifies that the true value and the predicted value do not correspond, such that the instance was undetected or erroneously partitioned into other instances; and FP indicates that the true value and the predicted value exhibit a one-to-many relationship, whereby a single instance is fragmented into multiple instances. r is defined as the proportion of true-positive samples correctly identified by the model out of all actual positive samples, thereby highlighting the model's sensitivity in capturing positive instances. p refers to the proportion of samples predicted as positive by the model that are truly positive, reflecting the model's accuracy in covering positive samples and its reliability in positive predictions.

4. Experiment and Results

4.1. Experimental Details

During the training phase, the specifics of the experimental setup utilized in the subsequent investigations of this study are presented in Table 2. For the TreeLearn dataset, we employed a training methodology consistent with the procedures outlined in its original publication. For the custom dataset, we initially pre-trained the model on the TreeLearn dataset, followed by fine-tuning through additional training exclusively on our own dataset. To enhance the convergence speed of the model, inspired by the work of Chandrabanshi et al. [33], a learning rate scheduler was incorporated during the training process. This approach facilitates accelerated convergence of the model by dynamically adjusting the learning rate.

Table 2. Runtime environment.

	Name	Details
Hardware	CPU	20 vCPU Intel(R) Xeon(R) Platinum 8457C
	Memory	150 GB RAM
	GPU	L20(48 GB)
Software	Operating System	Ubuntu 20.04
	Programming Language	Python 3.8
	Deep Learning Framework	PyTorch 1.9.0
	CUDA Version	CUDA 11.3
	cuDNN Version	cuDNN 8.2.0
Learning Rate Scheduler	t_initial	1300
	lr_min	0.0001
	cycle_decay	1
	warmup_lr_init	0.00001
	warmup_t	50
	cycle_limit	1
	t_in_epochs	True

4.2. TreeLearn Dataset Segmentation Results

In order to verify the generalization ability of the model, in this section we compare our method with several existing kinds of methods on the TreeLearn dataset.

The experimental results for each method are presented in Table 3, wherein the K-Means method serves as a manual feature extraction technique, whose accuracy is significantly influenced by the selection of initial seed points. Conversely, PointNet++ (MSG) and PCT are semantic segmentation models that leverage deep learning techniques. We assessed the semantic segmentation capabilities of these two models and ultimately opted for the PCT model, which demonstrated superior accuracy for semantic segmentation of the data. Subsequently, the K-Means method was employed for clustering. The results indicate that, while the recall is marginally lower, the overall segmentation accuracy surpasses that achieved by directly applying the K-Means method. The TreeLearn method represents an automated feature extraction for the single-tree segmentation approach, and the results clearly indicate that its segmentation accuracy markedly exceeds that of the first two methods. Moreover, our method, while exhibiting a slightly lower performance than the TreeLearn method, demonstrates a significantly improved accuracy compared to the first two methods.

Table 3. Overall results. Instance segmentation results averaged across predictions.

	Semantic Segmentation	**Instance Segmentation**		
	Accuracy	**Precision**	**Recall**	**F1**
K-Means [34]	\	67.12	92.87	77.92
PointNet++(MSG) [17]	85.53	\	\	\
PCT [35]	88.31	89.85	90.12	89.98
TreeLearn [19]	99.88	97.57	98.28	97.84
Ours	98.93	96.64	97.61	97.12

Furthermore, a detailed comparison between the TreeLearn method and our proposed method was conducted, analyzing the experiments from three perspectives: data preprocessing, model training, and real-time segmentation. The results are shown in Table 4.

Table 4. Detailed comparison of TreeLearn method and our method.

	Data		**Train**			**Segmentation**
	Number of Crops	**Occupied Space**	**Epoch**	**Time/Epoch**	**Total Epoch**	**Time**
TreeLearn	25,000	700 GB	2000	120 s	69 h	175 min
Ours	3000	80 GB	1400	140 s	54 h	179 min

During the data preprocessing stage, both methods necessitate the segmentation of the entire forest point cloud dataset to address memory constraints. However, our approach requires fewer segments and occupies less disk space, thereby demonstrating that our model outperforms the TreeLearn method in feature processing. In the training phase, our model necessitates a reduced training duration and exhibits a more rapid convergence rate. In the real-time segmentation phase, our method demonstrates performance that is comparable to that of the TreeLearn method. The comparative results indicate that, although our method exhibits marginally lower segmentation accuracy than the TreeLearn method, it presents notable advantages in both feature processing and model convergence speed.

4.3. Customized Dataset Segmentation Results

In this section we validate the applicability of the methods of this paper on datasets produced using image-based 3D reconstruction methods. The segmentation results are shown in Figure 4.

Furthermore, we fine-tuned several methods to enable their application to the customized dataset that we produced. Figure 5 illustrates the varying results obtained from our tests conducted on the eight sample plots. The findings indicate that the segmentation outcomes for the planted forest are generally superior to those for the natural forest. This phenomenon can be attributed to the relatively flatter and more uniformly distributed topography characteristic of planted forests. Furthermore, we conducted a comparative analysis of four methods, as shown in Table 5. In terms of semantic segmentation, our method achieved the highest accuracy, with an improvement of 18.77%, reaching 99.37%. In the case of instance segmentation, our method outperformed other typical approaches, with accuracy, recall, and F1 scores improving by 39.72%, 39.84%, and 44.87%, respectively. Overall, on the customized dataset, our method clearly outperformed the other approaches.

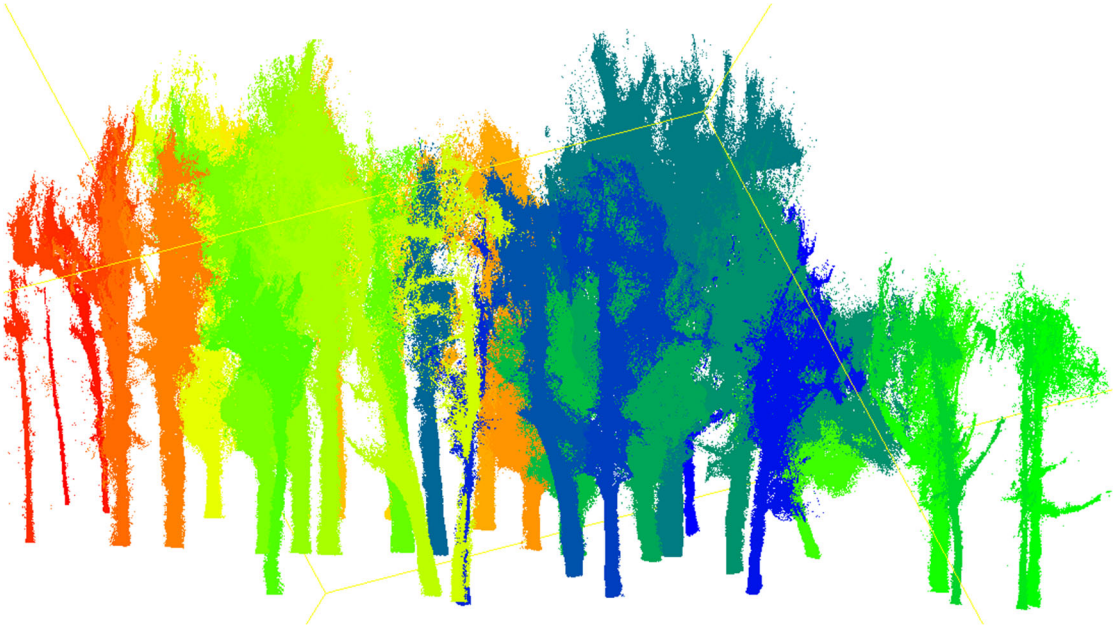


Figure 4. Segmentation results.

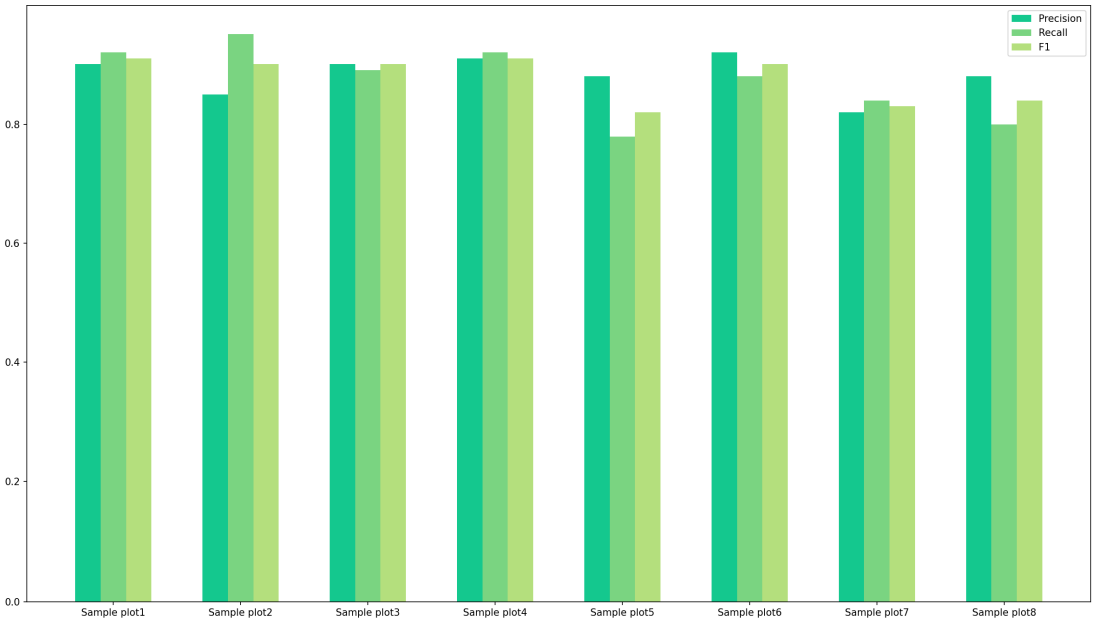


Figure 5. Comparison of sample segmentation indicators.

Table 5. Comparison of our and other methods on custom datasets.

	Semantic Segmentation	**Instance Segmentation**		
	Accuracy	**Precision**	**Recall**	**F1**
K-Means	\	41.97	52.64	46.71
PCT	80.60	60.76	71.92	65.87
TreeLearn	84.91	86.77	89.65	88.19
Ours	99.37	90.69	92.48	91.58

Figure 5 illustrates the comparative results of various sample segmentation metrics. Given the complexity inherent in sample segmentation, we evaluated the performance of three metrics—precision, recall, and F1 score—across the different samples. Through the

evaluation and analysis of the generated segmentation results, we present the comparative outcomes of these metrics across eight samples, as depicted in Figure 5. Specifically, owing to the minimal differences between trees within the sample plots, sample 2 demonstrates the highest recall, suggesting that the model is capable of recognizing the majority of actual positive samples in this instance. Conversely, owing to the significant differences between trees in the sample plots, sample plot 5 exhibits a relatively low recall, implying that the model may have overlooked some actual positive samples in this particular plot. This phenomenon may be attributed to the highly consistent morphological, textural, and spectral characteristics of the tree crowns in sample 2, which led to a reduced inter-class distance within the feature space of the model, thereby facilitating identification via clustering algorithms or threshold-based segmentation. Sample 5 likely comprises trees at various developmental stages, with overlapping canopies and shading effects leading to feature fragmentation, thereby posing challenges for the model in accurately distinguishing neighboring instances. Additionally, due to insufficient terrain leveling in sample plots 7 and 8, these plots also exhibit relatively low recall. Moreover, the accuracy and F1 scores exhibit a similar trend. Overall, our model demonstrates commendable performance across several sample plots.

4.4. Ablation Experiment

To elucidate the contributions of individual modules within our model, we devised an ablation experiment. This experiment seeks to analyze the contributions of these critical components to the overall performance of the model in instance segmentation through the systematic removal of the Sparse 3D U-Net, attention mechanism, and clustering modules. We established the baseline model, along with three comparative scenarios, each involving the removal of a single key module for the corresponding ablation model, as detailed in Table 6.

Table 6. Model ablation experiment and indicator comparison.

Sparse 3D U-Net	Attention	Clustering	Precision	Recall	F1	Overall
√	√	√	96.64%	97.61%	97.12%	Baseline
√	√		87.07%	85.26%	86.15%	↓10.97% (F1)
√		√	91.80%	76.39%	83.38%	↓2.77% (F1)
	√	√	79.67%	72.82%	76.09%	↓7.29% (F1)

In the ablation experiments, the baseline model unequivocally demonstrated superior performance, consistently outperforming the ablation models across all evaluated metrics. This singular result underscores the critical importance of the synergistic interactions among the modules within the complete model, collectively yielding substantial performance enhancements for the intricate task of forest 3D reconstruction. The overall performance of the model following the removal of the Sparse 3D U-Net is markedly diminished. This observation implies a pivotal role for the feature extraction capabilities of the Sparse 3D U-Net. The substantial decline in recall subsequent to the removal of the attention module indicates its efficacy in mitigating model overfitting. Following the removal of the clustering module, the indexes exhibit a decrease; however, they become more uniform, suggesting that this module enhances the overall performance of the model. The removal of the Sparse 3D U-Net results in the most pronounced decline in performance, underscoring the irreplaceable nature of this module, which demonstrates its unique advantages in the single-tree segmentation task. In summary, the findings from these ablation experiments not only substantiate the necessity of the individual modules within the model but also highlight their synergistic effects in enhancing single-tree segmentation performance. By

preserving the integrity of the model, researchers can fully leverage the strengths of each module to attain optimal segmentation outcomes.

5. Discussion

This study investigated the viability of a novel approach to single-tree segmentation technology by creating point cloud datasets suitable for deep learning applications through an image-based 3D reconstruction methodology, culminating in the utilization of the trained model for single-tree segmentation. This approach offers greater convenience and reduced costs in comparison to traditional methods utilizing LiDAR for data collection. Relative to alternative methodologies, our model demonstrates superior performance in processing datasets generated through image-based 3D reconstruction techniques, while also exhibiting high accuracy when handling LiDAR data.

The fundamental innovation of the model lies in the incorporation of a Sparse 3D U-Net architecture tailored for the specific characteristics of image-based 3D reconstruction data, along with the integration of modules including a multi-head attention mechanism, learnable query vectors, and cyclic prediction, which markedly enhances its capacity to process 3D reconstruction data. Specifically, the Sparse 3D U-Net architecture significantly enhances both the efficiency and accuracy of the model in processing large-scale forest point clouds by minimizing redundant parameters and computational complexity. Meanwhile, inspired by the pioneering work of Vaswani et al. [26], the multi-head attention mechanism allows the model to simultaneously process multiple feature subspaces during the feature extraction process, thereby adaptively enhancing the expressiveness of key features and effectively mitigating the impact of extraneous information, which notably improves the model's performance in scenarios characterized by an uneven point cloud distribution. The cyclic prediction module accelerates the convergence speed of the model by iteratively refining the prediction results and retaining only the final output. Furthermore, the overall performance of the model is enhanced by the synergistic interaction among these modules, facilitating efficient and accurate single-tree segmentation across a variety of forest scenarios.

When compared to TreeLearn, our approach demonstrates substantial improvements in feature representation and convergence efficiency. This advantage primarily stems from the multi-head attention mechanism integrated into our model, which allows for simultaneous focus on various segments of input features across multiple subspaces, thereby effectively capturing multi-scale and intricate dependency patterns within the data. Such comprehensive feature representation significantly improves the model's capacity to identify salient features, mitigate information loss and sparsity issues, and consequently enhance the accuracy and robustness of feature extraction.

Furthermore, the iterative prediction module enables the model to progressively refine its outputs through successive feedback cycles, thereby enhancing feature representations, capturing finer details, and correcting errors more effectively. This iterative mechanism not only amplifies the model's expressive capacity but also accelerates convergence during training, as each cycle iteratively refines the previous predictions, thereby substantially reducing the training duration required to attain optimal performance.

However, it is possible that the introduction of additional complexity during the optimization process adversely impacts the model's capacity to fit fine-grained details, thereby leading to a marginal decline in overall accuracy. And owing to limited project funding, the plot size employed in this study (15×15 m, totaling 225 m^2) is substantially smaller than the plots within the TreeLearn dataset, which span from 1.0 to 2.2 hectares. This disparity in spatial scale imposes notable constraints on direct dataset comparisons.

Specifically, smaller plots tend to provide a more localized snapshot of forest structure and species composition, which may not adequately reflect the heterogeneity inherent at broader spatial scales. As a result, models developed or validated using data from these smaller plots may demonstrate diminished generalizability when applied to larger-scale datasets, such as TreeLearn.

In this study, the TreeLearn dataset was employed for pre-training the model; however, it originates from a different geographical region than the customized dataset. Consequently, this regional discrepancy may adversely affect the model's performance, exemplified by decreased recall rates observed when substantial heterogeneity exists among trees within the sample plots. When point cloud data collected via LiDAR and image-based 3D reconstruction are obtained from the same region, it can facilitate more effective feature transfer for the target task, thereby enhancing both model performance and training efficiency. In future research, we will consider using both LiDAR and image-based 3D reconstruction techniques to collect data for the same study area in order to improve the performance of the model.

In the context of deep learning models, both the quantity and quality of data are pivotal factors influencing model performance. Therefore, a reduction in the difficulty of data acquisition will contribute positively to the enhancement of model performance. Certainly, alongside the necessity for high-quality data, model optimization represents another critical avenue for enhancing performance. Building upon the existing model, we are contemplating the integration of superpoint features, which are defined as points aggregated from homogeneous neighboring points according to geometric principles [36]. This methodology effectively circumvents the need for supervised features by utilizing undirected semantic and center distance labels [37]. Moreover, our approach employs a clustering-based strategy, with potential future explorations incorporating techniques such as bounding box detection or mask prediction.

In conclusion, our methodology adeptly integrates multiple innovative techniques, exploring and validating a new pathway for single-tree segmentation while addressing the challenges encountered by existing single-tree segmentation models in the context of 3D reconstruction data. Despite certain limitations inherent in the model, it is anticipated that, through subsequent optimization and refinement, it will assume a broader role in forest resource management and related fields.

6. Conclusions

In this study, we introduce a Sparse 3D U-Net framework augmented with an attention mechanism for the task of single-tree segmentation. The framework enables effective instance segmentation of forest point clouds by integrating multi-scale features with global contextual information. Experimental results demonstrate that the method achieved an F1 score of 97.12% on the TreeLearn benchmark dataset, surpassing the traditional K-Means method by 24.3 percentage points. Moreover, it attained an F1 score of 91.58% on the customized dataset, thereby validating its robust cross-scene generalization capability. The applicability of the single-tree segmentation approach was further validated using the 3D reconstruction dataset. In comparison to traditional manual feature extraction methods, deep learning approaches leveraging automatic feature extraction not only exhibit substantial advantages in terms of accuracy, but also effectively reduce labor costs, thereby playing a pivotal role in future intelligent processes.

Image-based 3D reconstruction techniques present a promising solution by mitigating the cost of data acquisition and enhancing the availability of open-source datasets. Forest inventory personnel can leverage affordable devices, such as smartphones or cameras, to collect data; these data can then be reconstructed into three-dimensional datasets, which

are subsequently processed by models for individual tree segmentation to accurately quantify tree counts, spatial distribution, and structural attributes, among other relevant features. It is anticipated that the proliferation of such datasets will empower deep learning-based methods to attain higher accuracy and enhanced generalization capabilities, thereby facilitating the establishment of a standardized framework for forest inventories.

Author Contributions: L.H. completed the experiments and wrote the paper. L.H. and Z.C. designed the specific scheme. L.H. and D.W. completed the result data analysis. X.Z. and L.D. collected the field data. Z.C. and X.Z. modified and directed the writing of the paper. X.Z. provided guidance in planning the experimental design. 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: Intelligent Forest Field Observation Equipment and Precision Extraction Technology for Tree Parameters, grant number 2023YFD2201701.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

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

References

1. Geng, J.; Liang, C. Analysis of the Internal Relationship between Ecological Value and Economic Value Based on the Forest Resources in China. *Sustainability* **2021**, *13*, 6795. [CrossRef]
2. Calders, K.; Adams, J.; Armston, J.; Bartholomeus, H.; Bauwens, S.; Bentley, L.P.; Chave, J.; Danson, F.M.; Demol, M.; Disney, M.; et al. Terrestrial Laser Scanning in Forest Ecology: Expanding the Horizon. *Remote Sens. Environ.* **2020**, *251*, 112102. [CrossRef]
3. Hao, J.; Li, X.; Wu, H.; Yang, K.; Zeng, Y.; Wang, Y.; Pan, Y. Extraction and Analysis of Tree Canopy Height Information in High-Voltage Transmission-Line Corridors by Using Integrated Optical Remote Sensing and LiDAR. *Geod. Geodyn.* **2023**, *14*, 292–303. [CrossRef]
4. Dai, F.; Rashidi, A.; Brilakis, I.; Vela, P. Comparison of Image-Based and Time-of-Flight-Based Technologies for Three-Dimensional Reconstruction of Infrastructure. *J. Constr. Eng. Manag.* **2013**, *139*, 69–79. [CrossRef]
5. Bai, S. Research on Single Tree Segmentation and DBH Parameter Extraction Algorithm Based on Point Cloud Data. Master's Thesis, Beijing University of Civil Engineering and Architecture, Beijing, China, 2020.
6. Maggi, L.W.G.Q. A New Method for Segmenting Individual Trees from the Lidar Point Cloud. *Photogramm. Eng. Remote Sens.* **2012**, *78*, 75–84.
7. Guo, H.; Gelfand, S.B. Classification Trees with Neural Network Feature Extraction. In Proceedings of the Proceedings 1992 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Champaign, IL, USA, 15–18 June 1992; IEEE Computer Society: Los Alamitos, CA, USA, 1992; pp. 183–188.
8. Chen, Q.; Baldocchi, D.; Gong, P.; Kelly, M. Isolating Individual Trees in a Savanna Woodland Using Small Footprint Lidar Data. *Photogramm. Eng. Remote Sens. J. Am. Soc. Photogramm.* **2006**, *72*, 923–932. [CrossRef]
9. Chen, R.; Li, C.; Yang, G.; Yang, H.; Xu, B.; Yang, X.; Zhu, Y.; Lei, L.; Zhang, C.; Dong, Z. Extraction of Crown Information from Individual Fruit Tree by UAV LiDAR. *Trans. Chin. Soc. Agric. Eng.* **2020**, *36*, 50–59.
10. Chen, Q. Airborne Lidar Data Processing and Information Extraction. *Photogramm. Eng. Remote Sens. J. Am. Soc. Photogramm.* **2007**, *73*, 109–112.
11. Zhang, C.; Zhou, Y.; Qiu, F. Individual Tree Segmentation from LiDAR Point Clouds for Urban Forest Inventory. *Remote Sens.* **2015**, *7*, 7892–7913. [CrossRef]
12. Hui, Z.; Li, N.; Cheng, P.; Li, Z.; Cai, Z. Single Tree Segmentation Method for Terrestrial LiDAR Point Cloud Based on Connectivity Marker Optimization. *Chin. J. Lasers* **2023**, *50*, 155–163.
13. Jiang, Z.; Chen, J.; Tang, L.; Yu, C.; Xie, R.; Huang, D.; Su, S. Tree Parameter Extraction in Fokienia Hodginsii Plantation Based on Airborne LiDAR Data. *Chin. J. Appl. Ecol.* **2024**, *35*, 321–329. [CrossRef]
14. Vega, C.; Hamrouni, A.; El Mokhtari, S.; Morel, J.; Bock, J.; Renaud, J.-P.; Bouvier, M.; Durrieu, S. PTrees: A Point-Based Approach to Forest Tree Extraction from Lidar Data. *Int. J. Appl. Earth Obs. Geoinf.* **2014**, *33*, 98–108. [CrossRef]
15. Shaheen, F.; Verma, B.; Asafuddoula, M. Impact of Automatic Feature Extraction in Deep Learning Architecture. In Proceedings of the 2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA), Gold Coast, QLD, Australia, 30 November–2 December 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 1–8.

16. Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. *arXiv* **2016**, arXiv:1612.00593.
17. 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.
18. Chen, X. Individual Tree Crown Segmentation Directly from UAV-Borne LiDAR Data Using the PointNet of Deep Learning. Master's Thesis, Nanjing Forestry University, Nanjing, China, 2021.
19. Henrich, J.; van Delden, J.; Seidel, D.; Kneib, T.; Ecker, A.S. TreeLearn: A Deep Learning Method for Segmenting Individual Trees from Ground-Based LiDAR Forest Point Clouds. *Ecol. Inform.* **2024**, *84*, 102888. [CrossRef]
20. Najafabadi, M.M.; Villanustre, F.; Khoshgoftaar, T.M.; Seliya, N.; Wald, R.; Muharemagic, E. Deep Learning Applications and Challenges in Big Data Analytics. *J. Big Data* **2015**, *2*, 1. [CrossRef]
21. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. *arXiv* **2015**, arXiv:1505.04597.
22. Çiçek, Ö.; Abdulkadir, A.; Lienkamp, S.S.; Brox, T.; Ronneberger, O. 3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation. *arXiv* **2016**, arXiv:1606.06650.
23. Sun, J.; Qing, C.; Tan, J.; Xu, X. Superpoint Transformer for 3d Scene Instance Segmentation. In Proceedings of the 37th AAAI Conference on Artificial Intelligence, Washington DC, USA, 7–14 February 2023; Volume 37, pp. 2393–2401.
24. Graham, B.; Engelcke, M.; van der Maaten, L. 3D Semantic Segmentation with Submanifold Sparse Convolutional Networks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 9224–9232.
25. Brauwers, G.; Frasincar, F. A General Survey on Attention Mechanisms in Deep Learning. *IEEE Trans. Knowl. Data Eng.* **2021**, *35*, 3279–3298. [CrossRef]
26. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. *arXiv* **2017**, arXiv:1706.03762.
27. Dai, L.; Chen, Z.; Zhang, X.; Wang, D.; Huo, L. CPH-Fmnet: An Optimized Deep Learning Model for Multi-View Stereo and Parameter Extraction in Complex Forest Scenes. *Forests* **2024**, *15*, 1860. [CrossRef]
28. Thang, V.; Kookhoi, K.; Thanh, N.; Luu, T.M.; Junyeong, K.; Yoo, C.D. Scalable SoftGroup for 3D Instance Segmentation on Point Clouds. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *46*, 1981–1995. [CrossRef] [PubMed]
29. Wolff, K.; Kim, C.; Zimmer, H.; Schroers, C.; Botsch, M.; Sorkine-Hornung, O.; Sorkine-Hornung, A. Point Cloud Noise and Outlier Removal for Image-Based 3D Reconstruction. In Proceedings of the 2016 Fourth International Conference on 3D Vision (3DV), Stanford, CA, USA, 25–28 October 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 118–127.
30. Leng, X.-L.; Miao, X.-A.; Liu, T. Using Recurrent Neural Network Structure with Enhanced Multi-Head Self-Attention for Sentiment Analysis. *Multimed. Tools Appl.* **2021**, *80*, 12581–12600. [CrossRef]
31. Ester, M.; Kriegel, H.P.; Sander, J.; Xu, X. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, Portland, OR, USA, 2–4 August 1996.
32. Zhao, K.; Suarez, J.C.; Garcia, M.; Hu, T.; Wang, C.; Londo, A. Utility of Multitemporal Lidar for Forest and Carbon Monitoring: Tree Growth, Biomass Dynamics, and Carbon Flux. *Remote Sens. Environ.* **2018**, *204*, 883–897. [CrossRef]
33. Chandrabanshi, V.; Domnic, S. A Novel Framework Using 3D-CNN and BiLSTM Model with Dynamic Learning Rate Scheduler for Visual Speech Recognition. *Signal Image Video Process.* **2024**, *18*, 5433–5448. [CrossRef]
34. Qian, Y.; Wang, J.; Zheng, X. Improved K-Means Clustering Method Based on Spectral Clustering and Particle Swarm Optimization for Individual Tree Segmentation of Airborne LiDAR Point Clouds. *J. Geo-Inf. Sci.* **2024**, *26*, 2177–2191.
35. Hao, G.M.; Xiong, C.J.; Ning, L.Z.; Jiang, M.T.; Martin, R.R.; Min, H.S. PCT: Point Cloud Transformer. *Comput. Vis. Media* **2021**, *7*, 187–199. [CrossRef]
36. Landrieu, L.; Simonovsky, M. Large-Scale Point Cloud Semantic Segmentation with Superpoint Graphs. *arXiv* **2017**, arXiv:1711.09869.
37. Liang, Z.; Li, Z.; Xu, S.; Tan, M.; Jia, K. Instance Segmentation in 3D Scenes Using Semantic Superpoint Tree Networks. *arXiv* **2021**, arXiv:2108.07478.

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.