



Article

A Multiple Criteria Decision-Making Method Generated by the Space Colonization Algorithm for Automated Pruning Strategies of Trees

Gang Zhao ^{1,2} and Dian Wang ^{1,2,*} ¹ School of Technology, Beijing Forestry University, Beijing 100083, China; zg1999@bjfu.edu.cn² Key Laboratory of Forestry Equipment and Automation of State Forestry and Grassland Bureau, Beijing Forestry University, Beijing 100083, China

* Correspondence: wangdian@bjfu.edu.cn

Abstract: The rise of mechanical automation in orchards has sparked research interest in developing robots capable of autonomous tree pruning operations. To achieve accurate pruning outcomes, these robots require robust perception systems that can reconstruct three-dimensional tree characteristics and execute appropriate pruning strategies. Three-dimensional modeling plays a crucial role in enabling accurate pruning outcomes. This paper introduces a specialized tree modeling approach using the space colonization algorithm (SCA) tailored for pruning. The proposed method extends SCA to operate in three-dimensional space, generating comprehensive cherry tree models. The resulting models are exported as normalized point cloud data, serving as the input dataset. Multiple criteria decision analysis is utilized to guide pruning decisions, incorporating various factors such as tree species, tree life cycle stages, and pruning strategies during real-world implementation. The pruning task is transformed into a point cloud neural network segmentation task, identifying the trunks and branches to be pruned. This approach reduces the data acquisition time and labor costs during development. Meanwhile, pruning training in a virtual environment is an application of digital twin technology, which makes it possible to combine the meta-universe with the automated pruning of fruit trees. Experimental results demonstrate superior performance compared to other pruning systems. The overall accuracy is 85%, with mean accuracy and mean Intersection over Union (IoU) values of 0.83 and 0.75. Trunks and branches are successfully segmented with class accuracies of 0.89 and 0.81, respectively, and Intersection over Union (IoU) metrics of 0.79 and 0.72. Compared to using the open-source synthetic tree dataset, this dataset yields 80% of the overall accuracy under the same conditions, which is an improvement of 6%.

Keywords: modeling trees; space colonization algorithm; dormant pruning; point cloud segmentation



Citation: Zhao, G.; Wang, D. A Multiple Criteria Decision-Making Method Generated by the Space Colonization Algorithm for Automated Pruning Strategies of Trees. *AgriEngineering* **2024**, *6*, 539–554. <https://doi.org/10.3390/agriengineering6010033>

Academic Editor: Bugao Xu

Received: 2 December 2023

Revised: 2 February 2024

Accepted: 22 February 2024

Published: 26 February 2024



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

1. Introduction

At present, the pruning of dormant trees is constrained by factors such as production methods, levels of mechanized automation, and farming practices, leading to the persistence of traditional manual pruning methods in some planting regions. This traditional approach exhibits drawbacks, such as low automation levels and inefficient management, which significantly hinder the economic development of the fruit tree industry. Particularly concerning is the continuous increase in manual labor costs associated with traditional pruning methods. In light of this, the mechanization and automation of fruit tree pruning can greatly enhance orchard management efficiency, reduce the workload of farmers, and enable precise pruning with the automated selection of pruning methods, thereby minimizing damage to the trees. Thus, the rapid realization of mechanized and automated fruit tree pruning is an urgent requirement for current agricultural economic development [1]. The establishment of a simulation platform is one of the fundamental tasks in the practical implementation of intelligent pruning automation. Utilizing virtual simulation technology

can effectively reduce the debugging and training costs of such products while enhancing the pruning accuracy of the robots [2].

Before guiding the pruning robot in performing pruning operations, it is essential to create a three-dimensional model of the target trees. Generating three-dimensional trees within a virtual environment has long been a significant challenge in the field of 3D reconstruction [3]. Runions et al. [4] introduced an algorithm that simulates spatial competition among tree branches to generate models of trees and shrubs. The model's attributes and parameters in the space colonization algorithm (SCA) specify the shape and granularity of space, branch visibility, and penetration levels. Additional parameters control the spatial and temporal distribution of attractor points, allowing for increased branch density near the tree crown boundaries and the formation of hierarchical branch structures of varying sizes [5]. These model parameters are closely related to the concepts used to describe the appearance of trees and shrubs in the landscape and can be adjusted to produce different branching structures. Xu et al. [6] focused on cherry and crabapple trees in a sample area, applying the SCA directly to Terrestrial Laser Scanning (TLS) point cloud data. The proposed method extracted skeletons from the point clouds of 100 reference trees. The measured experiments were compared, and the Root Mean Square Errors (RMSEs) of the branch angles, main trunk length, and branch length were found to be 3.309 degrees, 0.069 m, and 0.051 m, respectively. Guo et al. [7] presented an inverse procedural modeling method that uses deep learning to discover atomic structures, such as line segments or branches, and determine their orientations and scaling ratios. The detected structures are combined to form a tree. In a review, Crimaldi et al. [8] gave an overview of the most common approaches used to generate 3D tree models, discussing both methodologies and available commercial software. He presented a study that proposes a biomathematical model for tree growth and presents a 3D output of its structure through a real-time 3D-rendering environment (Unity©). The model is based on Ordinary Differential Equations (ODEs), which are used to calculate the growth of each individual internode in terms of length (primary growth) and width (secondary growth), as well as the accumulation of growth inhibitors that regulate the seasonal cyclicity of the tree. The feasibility of the model was verified through virtual experiments on environmental conditions (the amount of light and temperature) and the species characteristics of simulated trees (the number of buds and branching angle) [9]. Anastasiou et al. [10] assessed a laser scanner sensor for accurate estimation of pruning wood parameters in vineyards and found that the laser scanner emerged as a valuable tool for vineyard applications. It also shows that the point cloud data acquired by LiDAR are crucial for tree pruning.

Intelligent pruning techniques have emerged as a promising field in recent years. After creating three-dimensional models of the target trees, determining the pruning points and selecting appropriate pruning strategies for the pruning robot are critical issues in this domain [11]. Botterill et al. [12] proposed an electro-hydraulic control system for a grape pruning mobile platform. By illuminating the vine with LED lights and controlling the background, the system achieved a higher success rate in identifying pruning points. Karkee et al. [13] developed a machine vision system for the automated pruning of spindle-shaped apple trees. By processing tree images, the system identified pruning points and applied pruning principles, resulting in a recognition rate of 85% for long branches and a removal rate of 70% for branches. Jia et al. [14] studied the pruning point localization method for grape pruning robots and designed a grape pruning robot system. The system acquired grape tree images through a CCD camera, processed the images, and developed a pruning point localization algorithm based on bud information. Huang et al. [15] used a wide-angle camera to capture images of loquat branches and proposed a method for determining and locating the cutting points of loquat branches. The system achieved pruning point localization, branch pruning, and pulverization with a single branch operation time of 55 s. Liu et al. [16] presented an algorithm for extracting three-dimensional tree skeletons using an SR4000 depth sensor and conducted research on a pruning method based on a Backpropagation (BP) neural network. They proposed a pruning method for

fruit trees based on the BP neural network, which could be used for intelligent pruning decision making by pruning robots. Molaei et al. [17] outline a systematic approach for designing ‘Prubot,’ a highly efficient 7R manipulator tailored for automated grapevine cane pruning, demonstrating impressive kinematic performance in simulations. You et al. [18] designed an autonomous pruning robot system and developed a novel planning framework specifically designed for high-throughput operations. This framework reduced the motion planning time and skillfully selected cutting points, achieving high success rates in end-to-end experiments with ten different branching structures.

The goal of this work is to create a bridge between 3D tree modeling and automated pruning in a virtual environment, as the framework of our method is shown in Figure 1. To create this connection, the space colonization algorithm was used to generate the models of trees, which were subsequently imported into Blender© for tree segmentation. The segmented tree branches were classified with different labels, and annotated point cloud data were exported as datasets for automated pruning training in neural networks. The process of generating 3D trees and the architecture of neural networks used in this method are explained in more detail in Sections 2.1 and 2.2. The multiple criteria decision-making method is explained in Section 2.3. The processing of the experimental datasets is presented in Section 3, while the application of the datasets in the pruning neural network and the metrics for evaluating pruning performance are provided in Section 3.2, with the respective results of different cases shown in Section 4. Finally, in Section 5, the authors provide the conclusion and detail planned future enhancements to the method in robotic pruning.

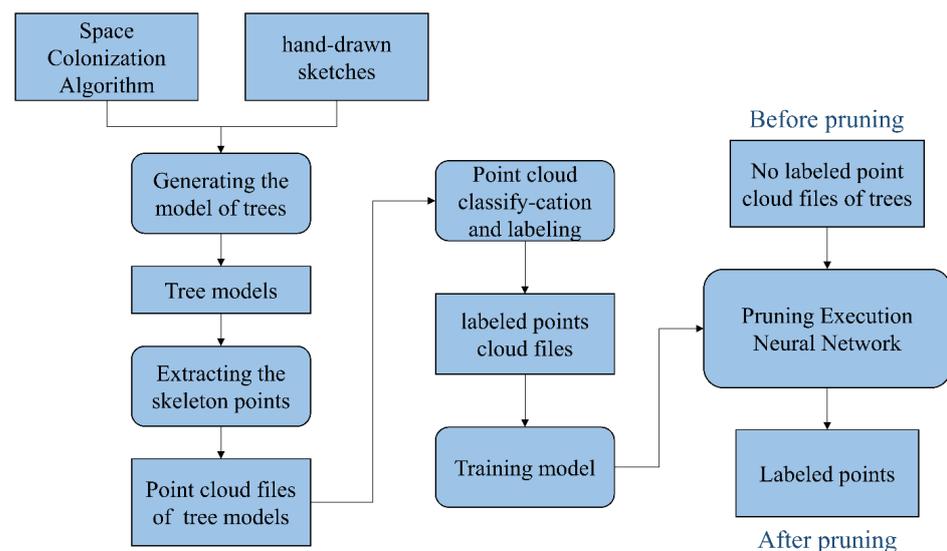


Figure 1. Framework of our modeling trees and pruning system.

2. Methodology

2.1. Space Colonization Algorithm for 3D Tree Generation

This algorithm, based on the principle of spatial competition in tree growth, enables the effective extraction of tree skeleton data even in obstructed scenarios [19]. The basic process of the whole algorithm is shown in Figure 2, with the following parts respectively: During the skeleton extraction process, the skeleton’s trajectory is influenced by spatial voids within the neighborhood of the growth points, following the principle of “nearest influence.” The skeleton is generated sequentially, with the algorithm terminating when no points have an impact on the skeleton [4]. Taking into account the upward phototropic growth characteristic of trees, the angle between branches and secondary branches is approximately maintained at 60 degrees. Hence, a growth angle constraint of 60 degrees is applied.

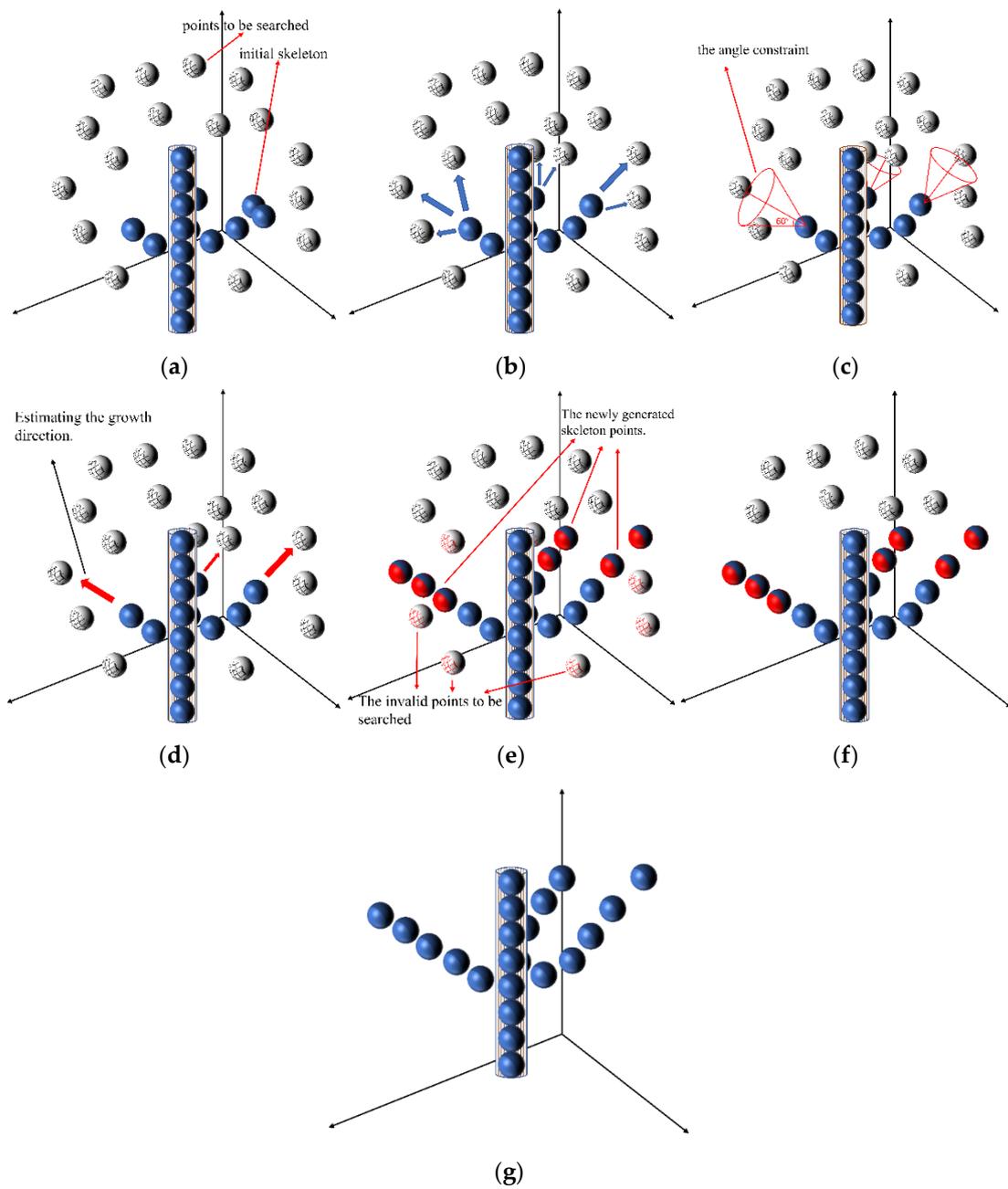


Figure 2. (a) The initial skeleton and the set of points to be searched; (b) Calculating the growth vector; (c) The angle constraint; (d) Estimating the growth direction; (e) Generating a new skeleton; (f) Deleting the invalid points to be searched; (g) Results of the First Iteration.

Initialization of the Tree Skeleton Array was performed to store the coordinates of the tree skeleton. A search radius, denoted as R_s , was set. The collection $S(p)$ was utilized to store the candidate points within the search radius of the skeleton point p , as illustrated in Figure 2a. Each candidate point could only influence the nearest skeleton point p , while p could be influenced by multiple candidate points within different search radii. The mathematical relationship can be expressed as follows:

$$|q - p| < R_s \tag{1}$$

$$|p - q| = \min(|p - q_x|, q_x \in S(p)) \tag{2}$$

where q is the set of points that are within a distance smaller than the search radius R_S from the skeleton point p .

Estimate the growth direction. Connect the skeleton point with all the candidate points that can influence the skeleton point, as shown in Figure 2b. Calculate the direction vector, represented by the black arrow in Figure 2c, by summing up the direction vectors. Normalize the summed vector to obtain the growth direction of the skeleton, as shown in Figure 2d. The specific mathematical relationship is as follows:

$$d = \sum_{p \in S(p)} \frac{p - q}{\|p - q\|} \quad (3)$$

$$n = \frac{d}{\|d\|} \quad (4)$$

where d is the direction vector and n is the normalized result vector.

To compare the estimated skeleton growth direction with the direction vector, the angle between them is calculated. If the angle is greater than 60° , the corresponding pending search point is removed, and the estimation of the growth direction is recalculated. This process is illustrated in Figure 2e. The mathematical relationship is as follows:

$$t = p - q \quad (5)$$

$$\arccos = \frac{dt}{|d||t|} \leq 60^\circ \quad (6)$$

where t is the growth vector.

The position of the new skeleton point, denoted as P_j , is determined by setting the spacing between skeleton points as D_p . In this case, the position of P_j is calculated as the product of P_j and $n \times D_p$. By multiplying f by the spacing D_p , the position of P_j can be obtained. This process is illustrated in Figure 2f.

To remove the candidate points, a deletion threshold R_d is set. If the spatial distance between a candidate point and the new skeleton point is less than the deletion threshold, the candidate point is deleted. The new skeleton points are represented by red solid circles, while the points to be deleted are enclosed in white solid circles. This process is illustrated in Figure 2f. The mathematical relationship is defined as follows:

$$|q - P_j| \leq R_d \quad (7)$$

The next iteration begins, continuing until there are no candidate points left to be found by the skeleton points. In other words, when $S(p)$ becomes empty, the iteration process is complete. Figure 2g shows the results after one iteration of the algorithm. It can be observed that three major lateral branches are growing diagonally upwards. Six candidate points have been removed, and the algorithm is ready for the next iteration.

During the process of constructing a 3D model of the tree, the morphology and growth direction of the tree can be influenced by adjusting the parameters in the tree growth process. The space colonization algorithm can be utilized to search for the optimal combination of parameters, aiming to achieve the best tree morphology.

By utilizing the space colonization algorithm, we have generated the three-dimensional skeleton of the tree. Next, we can proceed to fill this skeleton with solid structures by combining it with freehand sketches. This process involves using freehand sketches as references to add volume and shape to the skeleton, effectively transforming it into a complete and visually realistic tree model. The sketches provide guidance for the placement of branches and other structural elements, ensuring that the filled tree model aligns with the intended design or representation [20], as Figure 3. shows.



Figure 3. The automatically generated three-dimensional tree models.

To facilitate training for pruning simulation, the generation of a substantial dataset of tree models becomes essential. Employing the aforementioned approach significantly alleviates the data collection burden. As a result, we utilized the same methodology to generate a diverse set of 200 unique tree models, laying the foundation for subsequent utilization as inputs for the automatic pruning neural network. This approach not only streamlines the data generation process but also ensures the availability of a comprehensive and representative dataset for training purposes.

2.2. Proposed Deep Learning Architecture for Automated Pruning

We selected the PointNet++ network as our experimental model, which consists of multiple layers, each containing a certain number of neurons [21]. With appropriate modifications, the architecture of PointNet++ applicable for branch segmentation is shown in Figure 4. below. Specifically, we utilized the following network architecture: (a) Input layer: 4 input features (x , y , z coordinates and categories); (b) MLP (Multi-Layer Perceptron) layers: 3 MLP layers, each with 64 neurons; (c) Feature propagation layers: 2 feature propagation layers; (d) Segmentation layer: 2 output neurons for segmenting the point cloud into different categories (0, 1) [22].

To train the PointNet++ network, we chose appropriate hyperparameters. We set the learning rate to 0.001, batch size to 8, and utilized Adam optimizer as the optimizer. During the training process, we employed the following training strategy: (a) Number of epochs: 200; (b) Loss function: Cross-entropy loss; (c) Regularization: L2 regularization with a regularization parameter of 0.001; (d) Dropout: 0.5 dropout rate to prevent overfitting.

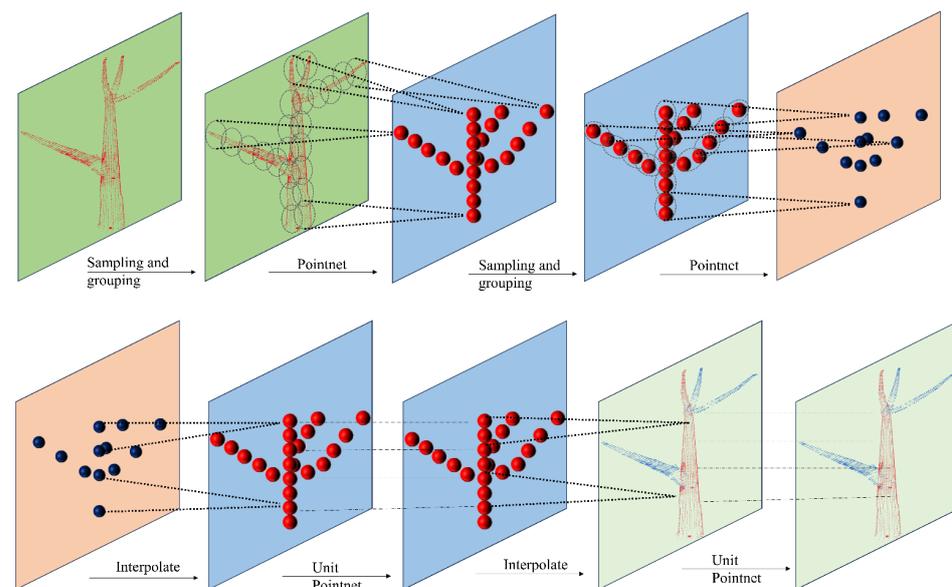


Figure 4. The architecture of PointNet++.

2.3. Multiple Criteria Decision-Making Method

A significant aspect of the pruning dilemma concerning dormant trees revolves around a decision-making process regarding the necessity of performing pruning actions on the tree's branches. Historically, this decision was typically made through manual pruning. However, in order to integrate intelligent pruning robots into the process, we propose a multi criteria decision-making approach to address this situation, as Figure 5 shows. This method amalgamates the knowledge gained from manual pruning. For instance, when dealing with a dormant cherry tree, criterion 1 pertains to whether the tree is mature or young. The primary purpose of pruning a young tree is to establish its height and shape, either as a sparsely dispersed structure or a naturally shaped heart. Criterion 2, applicable to mature trees, divides the tree's growth into three phases: early fruiting stage, fruiting stage, and senescence stage. Criterion 3 is implemented according to the specific growth stages. During the early fruiting stage, shortcuts and thinning of fruiting trunks are performed, retaining 3–5 branches of the upright trunks while removing dense peripheral branches and ineffective branches growing between the nodes. Throughout the fruiting period, the primary focus is on improving ventilation and light transmission within the canopy, with constant renewal of fruiting branches through pruning. In the senescence period, the main trunk is renewed by sawing off the entire crown at the appropriate location. Subsequently, a robust main trunk is selected, and approximately 50–100 cm of its length is retained to facilitate branching and cultivation as a high-quality lateral branch. These aforementioned guidelines are not exclusive to cherry trees but also have applicability in the pruning of most fruit trees. The establishment of such guidelines serves not only as a foundation for our simulation-based pruning work but also as a viable pruning logic framework for future automated pruning robots [23–25].

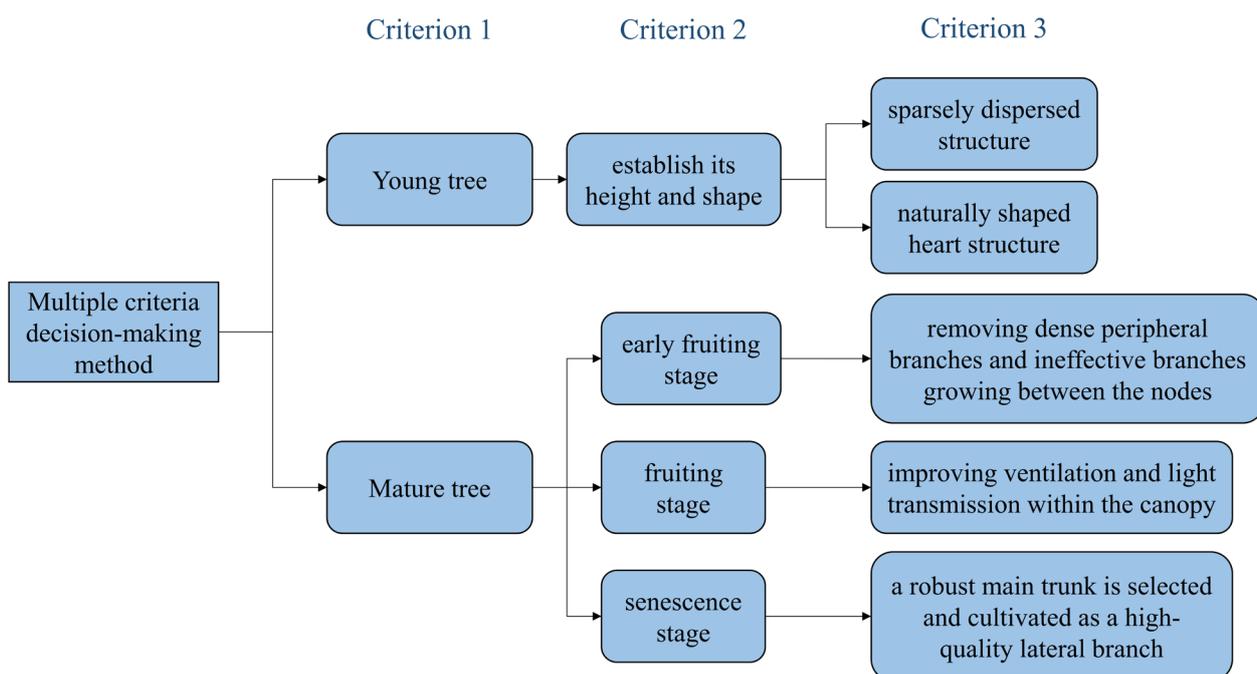


Figure 5. The multiple criteria decision-making method of pruning a dormant cherry tree.

3. Experimental Setup

3.1. Dataset

To train an automatic pruning deep learning model, it is necessary to prepare a suitable training dataset. The tree models generated using the space colonization algorithm mentioned earlier serve as a large virtual 3D tree dataset for automatic pruning. These tree models are further simplified and parameterized through hand-drawn sketches to obtain diverse and realistic tree samples. Each tree model in the dataset is labeled, including the main trunk, the branches to be pruned, and pruning nodes.

Prior to conducting experiments using the dataset, we applied a series of data preprocessing steps to ensure the quality and consistency of data. Firstly, we performed point cloud sampling to extract a specified number of points from the original data as experimental samples. Next, we normalized the coordinates of the point clouds, scaling them to a range of $[-1, 1]$ for improved network processing. Additionally, we encoded the labels, converting them into a numerical form, using 0 and 1 to represent different categories [26]. A set of 200 reconstructed 3D models depicting cherry trees underwent both training and testing processes. Among these, 150 models (75%) were randomly allocated for training purposes, while the remaining 50 trees (25%) were reserved for testing. Concurrently, the point clouds corresponding to the trees were meticulously classified into two categories, namely trunks and branches, serving as the ground truth values. This categorization was executed using the CloudCompare software v2.11, adopting a manual approach. The labeling process involved cropping individual branches using the polygon selection function within the software, followed by the addition of a label for each branch. The display of both branches and trunks was then hidden. The trunks and branches are annotated in red and blue, respectively, as exemplified in Figure 6. Consequently, all 200 trees in the dataset were annotated into two distinct categories. Each tree comprised approximately 9789–37,032 points, depending on its architectural complexity. To accommodate computational performance, point clouds for individual trees were subsampled using a voxel size set at 0.006 during the extensive experiments conducted for network training [27].

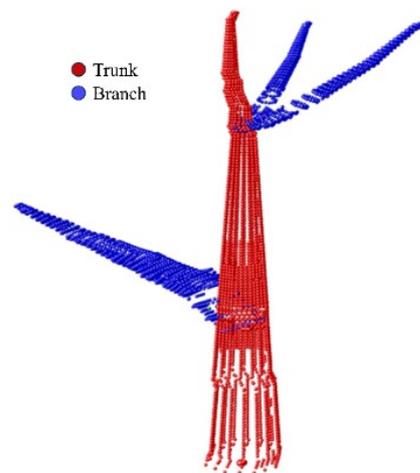


Figure 6. Example of labeled point clouds of the tree.

Our experimental environment is implemented using the PyTorch deep learning framework and Python language. We conducted training and testing on a computer equipped with an NVIDIA GPU. We utilized GPU acceleration based on CUDA and cuDNN to enhance the efficiency of model training and inference. The hardware facilities included a computer with an i5-11400 CPU, 3060Ti GPU, and 16 GB memory, with the software running on an Ubuntu 20.04 64bit system to ensure efficient and stable model training. Simultaneously, the hyperparameters employed in this study for PointNet++ were chosen, and the details can be found in Table 1.

Table 1. Hyperparameters used in PointNet++ for segmentation.

Training Hyperparameters	Values
Learning Rate	0.001
Epochs	200
Batch size	8
Optimizer	Adam

3.2. Metrics for Evaluating Pruning Performances

To evaluate our experimental results, we utilized accuracy as the primary metric. The assessment of network performance in segmenting the trunks and branches of modeled trees in the test dataset involved the utilization of various metrics. These metrics encompassed the overall accuracy (OA), class accuracy, Intersection over Union (IoU), mean accuracy (mAcc), and Mean Intersection over Union (mIoU). Overall accuracy (OA), determined by dividing the correctly classified point clouds by the total number of point clouds (Equation (8)), gauged the general accuracy of the segmentation. Class accuracy, expressing the likelihood of accurately classifying a specific category, was computed using Equation (9). The IoU, or Jaccard Index (Equation (10)), quantified the intersection area between the predicted and ground truth values. Additionally, traditional metrics such as mIoU (Equation (11)), indicating the average IoU across all classes, and mAcc (Equation (12)), representing the average accuracy across all classes, were considered in the evaluation process.

$$OA = 100\% \times \frac{\sum_{i=1}^N P_{ii}}{\sum_{j=1}^N \sum_{k=1}^N P_{jk}} \quad (8)$$

$$\text{ClassAccuracy} = \frac{TP}{(TP + FN)} \quad (9)$$

$$IoU = \frac{TP}{(TP + FP + FN)} \quad (10)$$

$$mIoU = \frac{\sum_{i=1}^n IoU}{N} \tag{11}$$

$$mAcc = \frac{\sum_{i=1}^n ClassAccuracy}{N} \tag{12}$$

4. Result and Discussion

Before the results are presented, we review and summarize the workflow depicted in Figure 7. First, the valid virtual tree models were generated using SCA and hand-drawn sketches. Next, the skeletons of these models were extracted as point cloud files. A subset of these files underwent the label classification process in CloudCompare, primarily classifying them into trunks and branches. The pruning execution model, based on the PointNet++ network, was trained using labeled tree point cloud files. After training, another subset of the unlabeled tree model point cloud files served as the test set to assess the pruning model’s results. The figure also demonstrates the changes observed in an unlabeled point cloud tree model before and after it was input into the pruning model, resulting in the separation of the tree into its main trunk and branches. This process accomplishes a pruning operation. Once the workflow is clarified, the presentation of the results naturally divides into two primary sections. The first section encompasses the outcomes of tree model generation, while the second section entails the segmentation results of the pruning execution model.

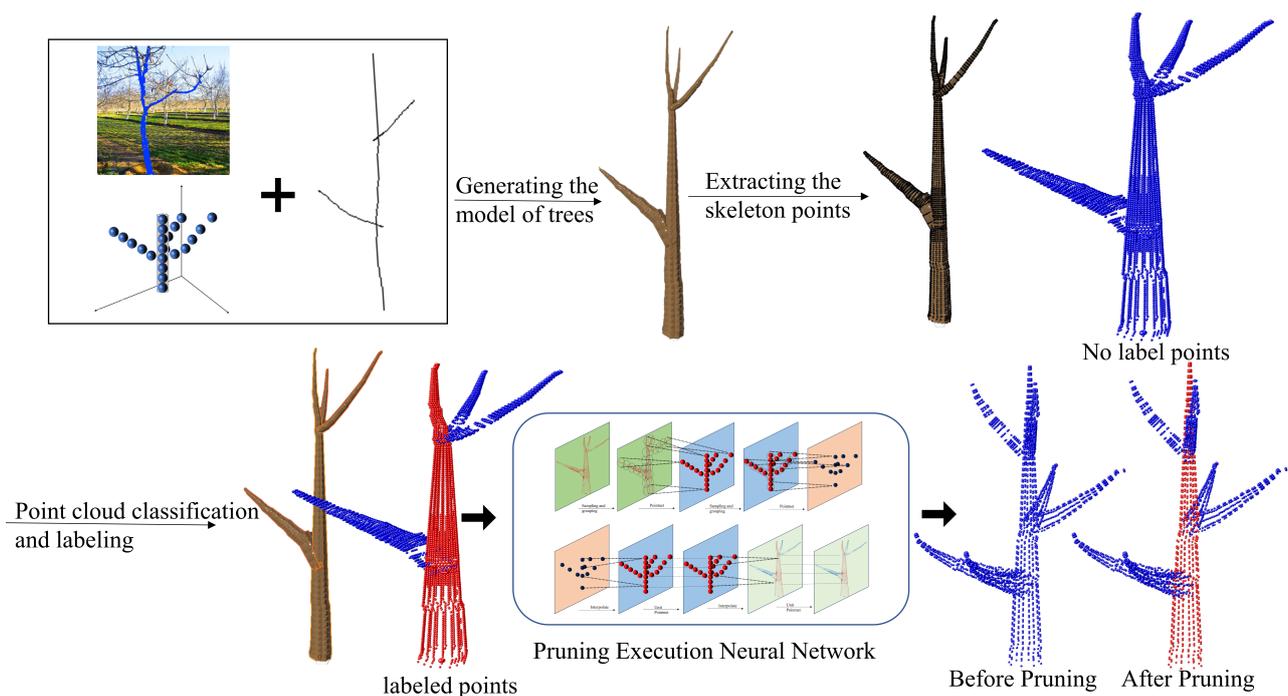


Figure 7. Workflow of our modeling trees and pruning system.

4.1. The Invalid Tree Models Generated by SCA

Dormant tree models, generated through the amalgamation of space colonization algorithms and hand-drawn sketches, do not always possess direct applicability in pruning experiments, as illustrated previously. Figure 8a–c showcase several invalid models that distinctly deviate from tree characteristics. These models exhibit excessive bending, incorrect iteration quantities, unrealistic branch competition, and other discrepancies. It is evident from the aforementioned case that this method has some drawbacks, with around one-fifth of the spanning trees deemed invalid. To obtain the required dataset of 200 trees for the experiment, 55 invalid models were discarded, resulting in an effective success rate

of 79%. This success rate was deemed feasible for generating trees specifically for pruning purposes.

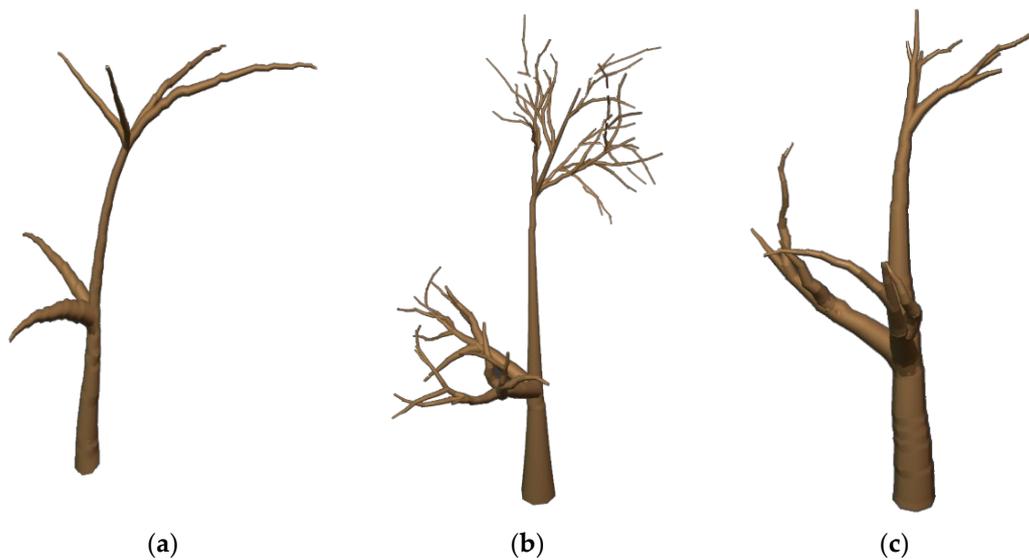


Figure 8. (a) Excessive bending; (b) incorrect iteration quantities; and (c) unrealistic branch competition.

4.2. The Performance Metrics of the Deep Learning Method for Pruning

The learning accuracy and minimum batch loss curves throughout the training of point clouds representing the modeled trees are depicted in Figure 9a,b below. The attained training accuracies, reaching 83%, and the corresponding minimum batch losses, reaching 0.06, collectively signify that this network was effectively trained on the dataset. This outcome indicates a successful training process without overfitting the training dataset.

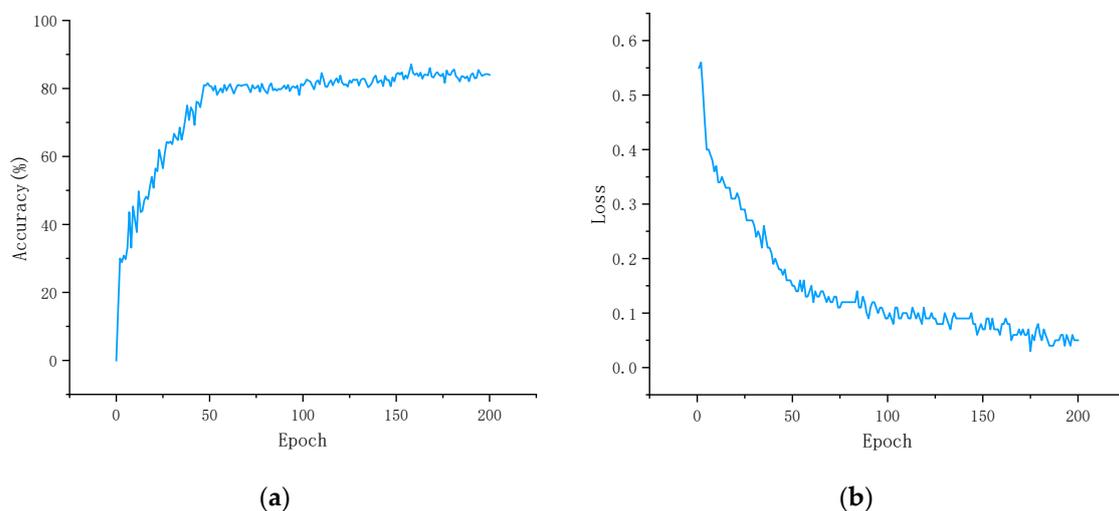


Figure 9. (a) Learning accuracy curve; (b) minimum batch loss curve of PointNet++.

As outlined in Section 3.1, a comprehensive evaluation of the network was conducted using a dataset comprising 200 trees. The test dataset, unseen by the network during training, produced an overall accuracy of 0.85, accompanied by mean accuracy and mean Intersection over Union (IoU) values of 0.83 and 0.75, respectively (refer to Figure 10a). Notably, the trunk of the tree achieved a class accuracy of 0.89 and an IoU of 0.79 (as illustrated in Figure 10b). Similarly, the tree branches exhibited a class accuracy of 0.81 and an IoU of 0.72 (also shown in Figure 10b). Despite the comparatively lower segmentation

accuracy for branches compared to trunks, this outcome underscores the satisfactory performance of the segmentation process for modeling trees.

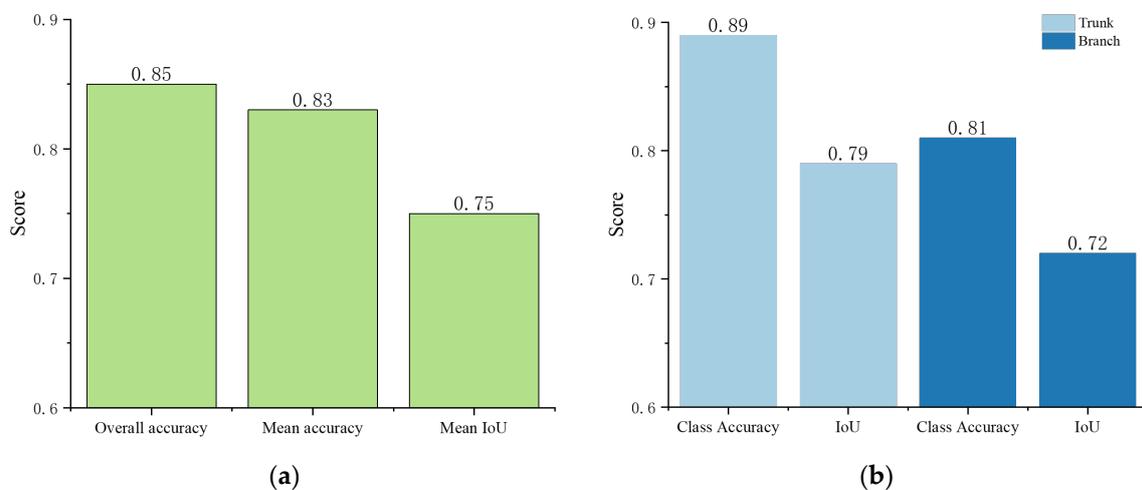


Figure 10. (a) Estimated performance over the entire test dataset; (b) comparison of performance measures for individual classes.

4.3. Comparison with Other Methods

A comparison was conducted between our experimental results and those reported in other papers. It should be emphasized that the utilization of diverse datasets and distinct data labeling methods renders these values not necessarily indicative of the algorithm's absolute performance. However, they do offer a valuable reference for our research.

Previous studies have employed various traditional approaches to generate 3D models of dormant trees for pruning, such as utilizing LiDAR or depth cameras. In our research, we adopted a different approach by generating our own tree model dataset instead of relying on vision hardware devices. This significant deviation reduced the time and labor costs associated with testing pruning robots. While acquiring a point cloud of a real-tree typically takes several hours, in our virtual environment, we could generate a tree model and convert it into a point cloud file within a mere ten minutes, with the speed primarily dependent on the computational capabilities of the computer. Consequently, our pruning-oriented modeling method significantly reduced the modeling time by at least 80% compared to other methods [13,28].

In order to illustrate the effectiveness of the method, we collected some cherry tree pictures in a cherry orchard in the Haidian District, Beijing, and compared them with the fruit tree model, as shown in Figure 11a below, which shows that the fruit tree model basically conforms to the growth characteristics of a real cherry tree.

Meanwhile, in order to compare and verify whether our generated dataset is valid, we also used the open-source point cloud dataset of synthetic trees for training [29]; this dataset provides 100 point clouds (600 in total) for each of the six tree species, which vary in intricacy and size. We chose the same kind of cherry tree point cloud model for training as Figure 11d shows, and it is shown that the cherry model generated by SCA achieves better performance in training, according to Table 2.

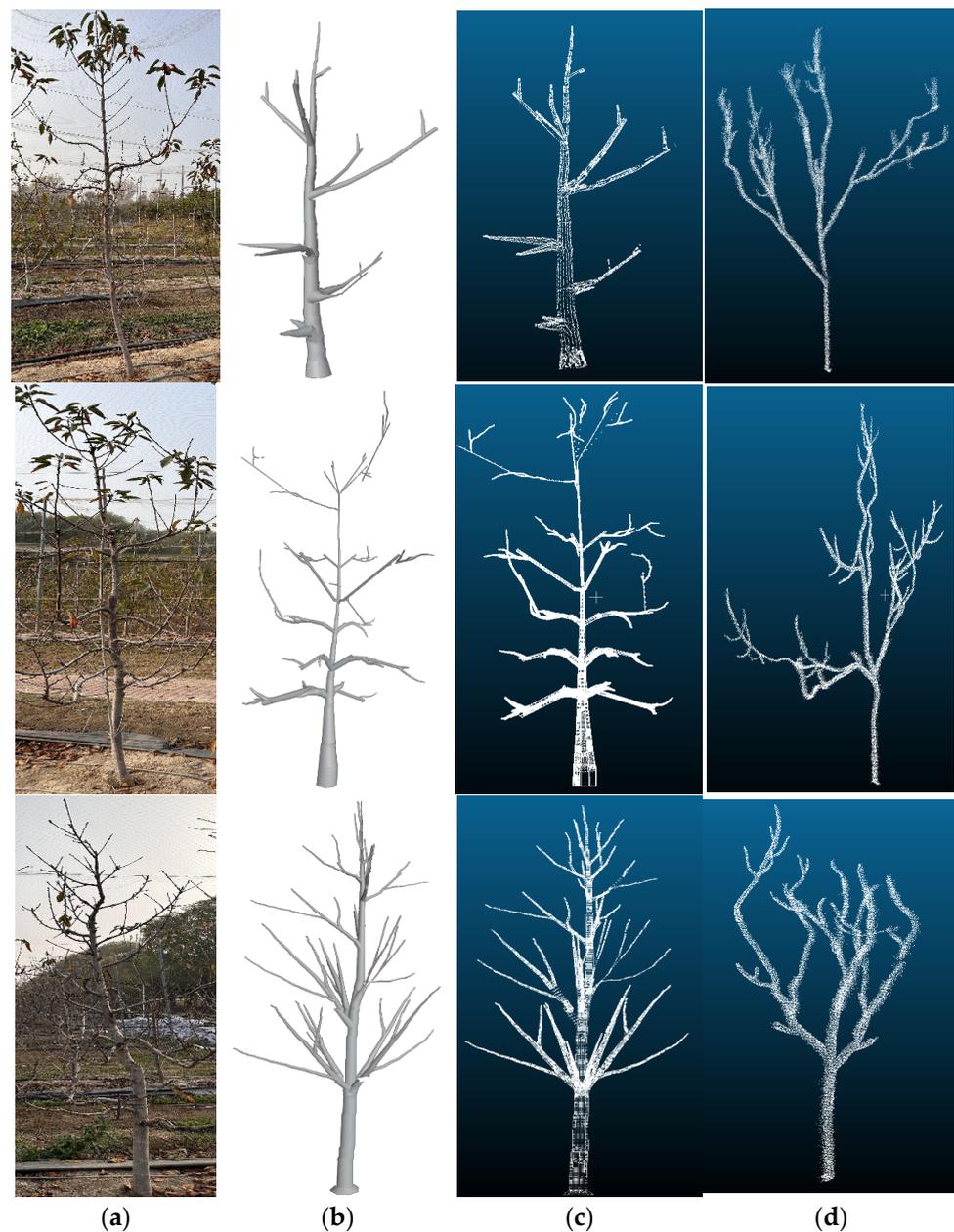


Figure 11. (a) Pictures of cherry trees in Beijing; (b) virtual cherry tree models generated by SCA; (c) virtual cherry tree point cloud models; and (d) synthetic cherry models.

Table 2. Comparison of cherry models generated by SCA and synthetic model training results.

Type of Model	Overall Accuracy	Mean Accuracy	Mean IoU
Cherry models generated by SCA	0.85	0.83	0.75
Synthetic cherry models	0.80	0.77	0.70

For the specific problem of tree pruning involving multi criteria decision making and machine learning techniques like support vector machines (SVMs) and random forest classifiers have traditionally been employed to segment reconstructed tree point clouds [5]. In our study, we utilize the PointNet++ network to segment tree point cloud files, achieving a true accuracy rate of 90% when compared to the SPGNet method without compromising recognition accuracy. Furthermore, the accuracy rate in the context of manual pruning reached 85%, although it fell short of the 93% accuracy achieved by the proposed method [26].

Nevertheless, the advantage lies in the convenience of obtaining the pruning targets in a virtual environment for subsequent real pruning. Simulating the real pruning environment within a virtual setting also holds significance in our paper's exploration of 3D modeling for dormant tree pruning.

5. Conclusions and Future Work

This research addresses the emerging requirement for autonomous tree pruning operations in the context of increasing mechanical automation within orchards. With the increasing popularity of digital twin technology and meta-universe concepts, pruning training in virtual environments can be applied to actual automated pruning. To fulfill this objective, we introduced a specialized tree modeling approach utilizing the space colonization algorithm (SCA) tailored explicitly for pruning purposes. By extending the SCA to a three-dimensional space, we successfully generated comprehensive cherry tree models, resulting in the foundation of our pruning system.

Our approach employed multiple criteria decision analysis to guide pruning decisions. This comprehensive approach ensures effective and efficient pruning operations. The results of our experiments demonstrate the superior performance of our proposed method in comparison to other pruning systems. We successfully segmented tree trunks and branches, achieving high-class accuracies of 0.89 and 0.81, respectively, along with Intersection over Union (IoU) metrics of 0.79 and 0.72. These metrics signify the accuracy and effectiveness of our approach in precisely identifying the pruning points. Furthermore, compared to synthetic tree models generated by other methods, the training results show that our method provides higher accuracy. When compared to manual pruning, our method achieves an impressive accuracy rate of 85%. These outcomes underscore the practical value and efficiency of our proposed approach.

Regarding future work, several potential avenues for further improvement and research exist. Firstly, enhancing pruning accuracy and efficiency through the exploration of advanced machine learning techniques and algorithms could prove fruitful. Additionally, investigating the integration of robotic control systems and feedback mechanisms to enable real-time adjustments and adaptive pruning strategies could further enhance the system's performance. Furthermore, expanding the scope of this study to encompass a wider variety of tree species and orchard environments could provide valuable insights for broader applicability.

In conclusion, this study lays the foundation for an autonomous pruning system for orchards in the context of digital twins, showcasing the potential of three-dimensional modeling, intelligent decision-making, and advanced segmentation techniques. Continued research and development in these areas will contribute to the advancement and practical implementation of autonomous pruning systems, resulting in increased efficiency, reduced labor costs, and improved pruning outcomes in agricultural practices.

Author Contributions: Conceptualization, G.Z. and D.W.; methodology, G.Z. and D.W.; software, G.Z.; validation, G.Z. and D.W.; formal analysis, G.Z. and D.W.; investigation, G.Z. and D.W.; resources, G.Z. and D.W.; data curation, G.Z. and D.W.; writing—original draft preparation, G.Z.; writing—review and editing, G.Z. and D.W.; visualization, G.Z. and D.W.; supervision, D.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research received funding by contract research of non-government funded projects (2022092803003).

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. He, L.; Schupp, J. Sensing and Automation in Pruning of Apple Trees: A Review. *Agronomy* **2018**, *8*, 211. [\[CrossRef\]](#)
2. Gongal, A.; Amatya, S.; Karkee, M.; Zhang, Q.; Lewis, K. Sensors and Systems for Fruit Detection and Localization: A Review. *Comput. Electron. Agric.* **2015**, *116*, 8–19. [\[CrossRef\]](#)
3. Tang, L.; Chen, C.; Huang, H.; Lin, D. An Integrated System for 3D Tree Modeling and Growth Simulation. *Environ. Earth Sci.* **2015**, *74*, 7015–7028. [\[CrossRef\]](#)
4. Runions, A.; Lane, B.; Prusinkiewicz, P. Modeling Trees with a Space Colonization Algorithm. *Nph* **2007**, *7*, 6.
5. Liu, Y.; Guo, J.; Benes, B.; Deussen, O.; Zhang, X.; Huang, H. TreePartNet: Neural Decomposition of Point Clouds for 3D Tree Reconstruction. *ACM Trans. Graph.* **2021**, *40*, 232. [\[CrossRef\]](#)
6. Xu, Y.; Hu, C.; Xie, Y. An Improved Space Colonization Algorithm with DBSCAN Clustering for a Single Tree Skeleton Extraction. *Int. J. Remote Sens.* **2022**, *43*, 3692–3713. [\[CrossRef\]](#)
7. Guo, J.; Jiang, H.; Benes, B.; Deussen, O.; Zhang, X.; Lischinski, D.; Huang, H. Inverse Procedural Modeling of Branching Structures by Inferring L-Systems. *ACM Trans. Graph.* **2020**, *39*, 115. [\[CrossRef\]](#)
8. Crimaldi, M.; Carteni, F.; Giannino, F. VISmaF: Synthetic Tree for Immersive Virtual Visualization in Smart Farming. Part I: Scientific Background Review and Model Proposal. *Agronomy* **2021**, *11*, 2458. [\[CrossRef\]](#)
9. Crimaldi, M.; Carteni, F.; Bonanomi, G.; Giannino, F. Integration of a System Dynamics Model and 3D Tree Rendering—VISmaF Part II: Model Development, Results and Potential Agronomic Applications. *Agronomy* **2023**, *13*, 218. [\[CrossRef\]](#)
10. Anastasiou, E.; Balafoutis, A.; Theocharis, S.; Theodorou, N.; Koundouras, S.; Fountas, S. Assessment of Laser Scanner Use under Different Settings in Two Differently Managed Vineyards for Estimating Pruning Wood Parameters. *AgriEngineering* **2022**, *4*, 733–746. [\[CrossRef\]](#)
11. Verbiest, R.; Ruysen, K.; Vanwalleghem, T.; Demeester, E.; Kellens, K. Automation and Robotics in the Cultivation of Pome Fruit: Where Do We Stand Today? *J. Field Robot.* **2021**, *38*, 513–531. [\[CrossRef\]](#)
12. Botterill, T.; Paulin, S.; Green, R.; Williams, S.; Lin, J.; Saxton, V.; Mills, S.; Chen, X.; Corbett-Davies, S. A Robot System for Pruning Grape Vines: A Robot System for Pruning Grape Vines. *J. Field Robot.* **2017**, *34*, 1100–1122. [\[CrossRef\]](#)
13. Karkee, M.; Adhikari, B.; Amatya, S.; Zhang, Q. Identification of Pruning Branches in Tall Spindle Apple Trees for Automated Pruning. *Comput. Electron. Agric.* **2014**, *103*, 127–135. [\[CrossRef\]](#)
14. Jia, T. Study on the Pruning Point Locating Method of the Grapevine Winter Pruning Robot. Master's Thesis, Zhejiang University of Technology, Hangzhou, China, 2012.
15. Huang, B.; Shao, M.; Song, L. Vision Recognition and Framework Extraction of Loquat Branch-Pruning Robot. *J. South China Univ. Technol. (Nat. Sci. Ed.)* **2015**, *43*, 114–119+126.
16. Liu, S.; Yao, J.; Li, H.; Qiu, C.; Liu, R. Research on a Method of Fruit Tree Pruning Based on BP Neural Network. *J. Phys. Conf. Ser.* **2019**, *1237*, 042047. [\[CrossRef\]](#)
17. Molaei, F.; Ghatrehsamani, S. Kinematic-Based Multi-Objective Design Optimization of a Grapevine Pruning Robotic Manipulator. *AgriEngineering* **2022**, *4*, 606–625. [\[CrossRef\]](#)
18. You, A.; Grimm, C.; Silwal, A.; Davidson, J.R. Semantics-Guided Skeletonization of Upright Fruiting Offshoot Trees for Robotic Pruning. *Comput. Electron. Agric.* **2022**, *192*, 106622. [\[CrossRef\]](#)
19. Shi, Y.; He, P.; Hu, S.; Zhang, Z.; Geng, N.; He, D. Reconstruction Method of Tree Geometric Structures from Point Clouds Based on Angle-Constrained Space Colonization Algorithm. *Trans. Chin. Soc. Agric. Mach.* **2018**, *49*, 207–216.
20. Okabe, M.; Owada, S.; Igarashi, T. Interactive Design of Botanical Trees Using Freehand Sketches and Example-Based Editing. In Proceedings of the ACM SIGGRAPH 2007 Courses. Association for Computing Machinery, New York, NY, USA, 5–9 August 2007; p. 26–es. [\[CrossRef\]](#)
21. Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 5099–5108.
22. Shen, X.; Huang, Q.; Wang, X.; Li, J.; Xi, B. A Deep Learning-Based Method for Extracting Standing Wood Feature Parameters from Terrestrial Laser Scanning Point Clouds of Artificially Planted Forest. *Remote Sens.* **2022**, *14*, 3842. [\[CrossRef\]](#)
23. Song, W. Walnut Tree Plastic Pruning Technology. *Rural Sci. Technol.* **2020**, *4*, 49–50. [\[CrossRef\]](#)
24. Balan, V.; Şarban, V. The Impact of The Cherry Tree Pruning Period on The Production and Quality of Fruit in an Intensive Cultivation System. In Proceedings of the International Agriculture Congress's (UTAK2021), Online, 16–17 December 2021; pp. 107–117.
25. Rutkowski, K.; Zydlik, Z.; Pacholak, E. Effect of Tree Pruning Intensity on the Yield and Fruit Quality of the Sour Cherry. *Zemdirbyste* **2015**, *102*, 417–422. [\[CrossRef\]](#)
26. Ma, B.; Du, J.; Wang, L.; Jiang, H.; Zhou, M. Automatic Branch Detection of Jujube Trees Based on 3D Reconstruction for Dormant Pruning Using the Deep Learning-Based Method. *Comput. Electron. Agric.* **2021**, *190*, 106484. [\[CrossRef\]](#)
27. Borrenpohl, D.; Karkee, M. Automated Pruning Decisions in Dormant Canopies Using Instance Segmentation. In Proceedings of the 2022 ASABE Annual International Meeting, Houston, TX, USA, 17–20 July 2022; American Society of Agricultural and Biological Engineers: St. Joseph, MI, USA, 2022.

28. Kolmanič, S.; Strnad, D.; Kohek, Š.; Benes, B.; Hirst, P.; Žalik, B. An Algorithm for Automatic Dormant Tree Pruning. *Appl. Soft Comput.* **2021**, *99*, 106931. [[CrossRef](#)]
29. Dobbs, H.; Batchelor, O.; Green, R.; Atlas, J. Smart-Tree: Neural Medial Axis Approximation of Point Clouds for 3D Tree Skeletonization. In Proceedings of the Iberian Conference on Pattern Recognition and Image Analysis, Alicante, Spain, 7–9 June 2023; Springer: Cham, Switzerland, 2023; pp. 351–362.

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.