

Article

Pruning Branch Recognition and Pruning Point Localization for Walnut (*Juglans regia* L.) Trees Based on Point Cloud Semantic Segmentation

Wei Zhu ^{1,2,3}, Xiaopeng Bai ^{1,2,3}, Daochun Xu ^{1,2,3} and Wenbin Li ^{1,2,3,*}

¹ School of Technology, Beijing Forestry University, Beijing 100083, China; zhuwei07020026@bjfu.edu.cn (W.Z.); xiaopengbai@bjfu.edu.cn (X.B.); xudaochun@bjfu.edu.cn (D.X.)

² State Key Laboratory of Efficient Production of Forest Resources, Beijing 100083, China

³ Key Laboratory of National Forestry and Grassland Administration on Forestry Equipment and Automation, Beijing 100083, China

* Correspondence: leewb@bjfu.edu.cn; Tel.: +86-13910659735

Abstract: Intelligent pruning technology is significant in reducing management costs and improving operational efficiency. In this study, a branch recognition and pruning point localization method was proposed for dormant walnut (*Juglans regia* L.) trees. First, 3D point clouds of walnut trees were reconstructed from multi-view images using Neural Radiance Fields (NeRFs). Second, Walnut-PointNet was improved to segment the walnut tree into Trunk, Branch, and Calibration categories. Next, individual pruning branches were extracted by cluster analysis and pruning rules were adjusted by classifying branches based on length. Finally, Principal Component Analysis (PCA) was used for length extraction, and pruning points were determined based on pruning rules. Walnut-PointNet achieved an OA of 93.39%, an ACC of 95.29%, and an mIoU of 0.912 on the walnut tree dataset. The mean absolute errors in length extraction for the short-growing branch group and the water sprout were 28.04 mm and 50.11 mm, respectively. The average success rate of pruning point recognition reached 89.33%, and the total time for pruning branch recognition and pruning point localization for the entire tree was approximately 16 s. This study provides support for the development of intelligent pruning for walnut trees.

Keywords: walnut tree; intelligent pruning; 3D point cloud; semantic segmentation; PointNet++; branch recognition; pruning point localization

Academic Editor: Roberto Alves Braga Júnior

Received: 19 March 2025

Revised: 4 April 2025

Accepted: 8 April 2025

Published: 9 April 2025

Citation: Zhu, W.; Bai, X.; Xu, D.; Li, W. Pruning Branch Recognition and Pruning Point Localization for Walnut (*Juglans regia* L.) Trees Based on Point Cloud Semantic Segmentation. *Agriculture* **2025**, *15*, 817. <https://doi.org/10.3390/agriculture15080817>

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

Walnut (*Juglans regia* L.) is an important woody oil species, with a wide planting area. In walnut orchard management, pruning is still primarily performed manually. However, issues such as the shortage of agricultural labor, rising labor costs, and an aging workforce are limiting the walnut industry. Therefore, the development of intelligent pruning technology to achieve selective pruning is crucial [1]. The keys to intelligent pruning for fruit trees are identifying pruning branches and locating pruning points. At present, there are two main methods for identifying branches in fruit trees: 2D image and 3D point cloud.

With the development of image processing technology, image-based methods for fruit tree branch identification have made significant progress. Amatya et al. [2] employed a Bayesian classifier to categorize cherry tree image pixels into four classes, achieving a

branch detection accuracy of 89.2%. Tabb and Medeiros [3] captured fruit tree images in outdoor lighting conditions after removing complex backgrounds, located low-texture regions of the image using superpixels and applied a Gaussian mixture model for branch modeling and segmentation. With the continuous development of deep learning in image recognition, it has been widely applied to fruit tree branch identification. Various deep learning-based image segmentation models, such as U-Net [4], Deeplab v3 [5], YOLOv8 [6], YOLOv11 [7], Mask R-CNN [8,9], and Transformer-based models [10] have been analyzed and compared for segmenting branches and trunks in fruit tree images. Majeed et al. [11] utilized SegNet, a CNN-based segmentation model, to segment both simple and foreground RGB images of apple trees, achieving higher accuracies in foreground RGB images (0.91 for trunks, 0.92 for branches) than in simple RGB images. Borrenpohl and Karkee [12] used Mask R-CNN to segment the trunks and leaders of cherry trees in active and natural lighting images, demonstrating that active illumination increased the IoU of Mask R-CNN by 11%. The above-mentioned 2D image-based branch identification methods offer convenient data acquisition and fast processing. However, they are limited to planar information, making it difficult to capture branch spatial structures for pruning applications. To overcome this, researchers have integrated depth information into instance segmentation networks to enhance feature extraction and improve recognition accuracy [13–16]. However, the complex arrangement of fruit tree branches, including small gaps and occlusions, still challenges high segmentation accuracy for dense structures, even with depth information integration.

Three-dimensional point clouds have gained attention as a three-dimensional and intuitive representation, containing rich spatial structural information. Representing fruit trees as 3D point clouds for branch identification research has become a popular approach. Karkee et al. [17] used a depth camera and the medial axis thinning algorithm to extract the skeleton point cloud of apple trees, achieving 77% accuracy in branch identification. Li et al. [18] employed an improved Laplace algorithm for skeleton extraction, successfully performing branch segmentation. The above methods mainly rely on traditional algorithms and geometric properties, limiting their adaptability to complex tree structures. With significant advancements in deep learning techniques for 3D point cloud classification and semantic segmentation [19], researchers have extensively explored deep learning-based tree point cloud segmentation [20–22]. Ma et al. [23] used two Kinect DK depth cameras to acquire jujube tree point clouds and applied the SPGNet network in combination with the DBSCAN clustering algorithm for segmentation, achieving 93% accuracy for trunks and 84% for branches. Qi et al. [24] proposed the PointNet++ network, which enabled part segmentation for single objects and semantic segmentation for multi-object scenes. It has been used for apple tree trunk and branch segmentation, achieving an overall accuracy of 0.84 on uncolored point clouds [25]. Building on PointNet++, Jiang et al. [26] utilized PointNeXt, which achieved an mIoU of 0.9481 for apple tree trunk and branch segmentation, outperforming both PointNet and PointNet++. Compared to traditional methods, deep learning offers greater adaptability and robustness in processing diverse and complex branch structures [27].

After identifying tree branches, pruning points must be determined based on pruning rules. For 2D image-based branch identification, additional 3D information is often required for accurate localization. Integrating RGB images with depth data has been an effective solution [28]. By capturing both RGB and depth images of fruit trees with a depth camera, researchers have used instance segmentation models to segment trees in the RGB images. Through pixel coordinate transformation, the segmented information was mapped onto the depth images, allowing for the extraction of branch parameters and coordinates. This approach has been successfully used to identify pruning points for apple tree branches and grapevines [29,30]. However, the data captured by depth cameras are

often influenced by complex backgrounds and lighting conditions in outdoor environments. Additionally, the limited capture range only allows for the acquisition of local branch information, which limits the application to the structures of tall walnut trees. With the development of 3D reconstruction technology, a global 3D point cloud of the fruit tree better captures the branch structure, improving pruning point accuracy. The skeleton point cloud of apple trees was extracted to locate pruning positions, showing a good match with manual pruning [17]. Existing studies have explored 3D point cloud-based intelligent pruning for apple trees [11,31], jujube trees [23,32,33], sweet pepper plants [34], vines [35], and sweet cherry trees [36]. Different pruning rules should be adopted depending on the tree species. In existing studies, pruning points for apple trees were determined based on branch diameter and spacing [17,29]. For grapevines, pruning decisions relied on the number and position of buds on each cane [30,37]. Sweet cherry tree pruning locations were determined by stub length [38]. These factors were closely related to tree architecture and cultivation practices. Walnut trees typically exhibit a sparse-layered or open-center structure with slender, dense, and intertwined branches. Pruning point localization must be analyzed based on their specific growth characteristics.

In summary, to overcome the limitations of 2D images and the complexity of walnut tree branches, this study proposed a 3D point cloud-based method for pruning branch identification and pruning point localization in walnut trees. By segmenting the 3D point cloud, the method identified target branches and determined pruning points based on specific intelligent pruning rules for walnut trees. The specific research objectives were as follows: (1) constructing a high-quality 3D point cloud dataset of walnut trees and developing the Walnut-PointNet model to segment trunks and one-year-old branches; (2) developing intelligent pruning rules based on tree architecture and branch growth characteristics; (3) extracting branch length parameters; (4) determining pruning points according to pruning rules and extracted parameters.

2. Materials and Methods

2.1. Data Collection and Point Cloud Generation

This study was conducted using leafless Xiangling walnut trees (8 years old on average). On average, the trees in the selected orchard were 3.5 m tall, with approximately 4.5 m between rows and 3 m between trees according to a standardized planting pattern. The experimental site was a modern walnut orchard located in Huanglong County, Shanxi province, China (35°34' N, 109°50' E). Based on the previous experimental method, data were collected under natural light conditions without any manual intervention using a Canon M50 digital camera (Canon, Tokyo, Japan), which has an effective pixel number of 24.1 million and a maximum video resolution of 3840 × 2160. Videos were captured along a circular track from the bottom to the top of each tree to capture images at different levels. In total, 200 multi-view images were extracted from the video for each tree and a total of 100 walnut trees' images were collected.

Point clouds can more accurately capture the complex structure and fine details of trees. In previous experiments, three methods were compared for the 3D reconstruction of walnut trees to obtain the point clouds: First, multi-viewpoint clouds were captured using a Kinect depth camera and registered to form a complete model. Next, the SfM-MVS (Structure from Motion–Multi View Stereo) method and the NeRFs (Neural Radiance Fields) method were applied separately for the 3D reconstruction of walnut trees. The results indicated that NeRF outperformed the other methods in walnut tree 3D reconstruction, providing higher accuracy and more detailed results. Therefore, this study used NeRF to reconstruct high-quality color 3D point clouds of individual walnut trees from multi-view images, as shown in Figure 1, with a total of 100 walnut trees reconstructed. The point clouds were then preprocessed for subsequent analysis, including background

noise removal, downsampling, and normalization. To separate the trees from the complex background, random uniform sampling was applied to reduce the point count in the original point clouds, and normalization was carried out by scaling the 3D coordinates to the range $[-1, 1]$.

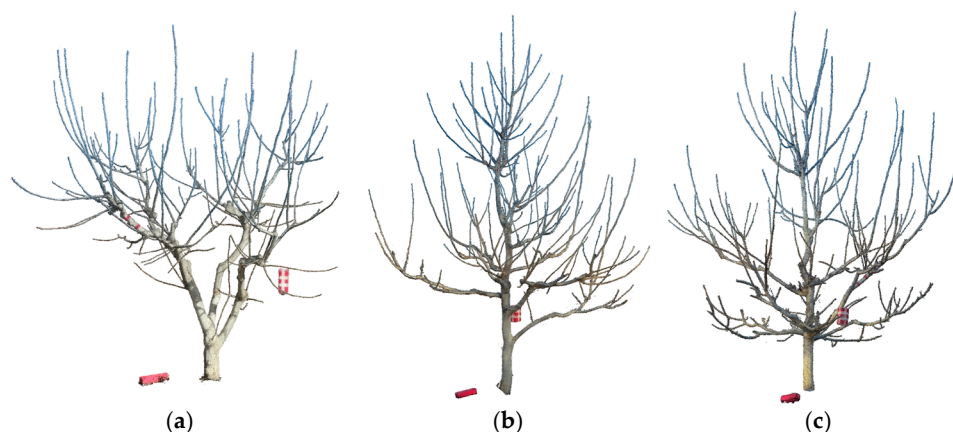


Figure 1. Reconstructed point cloud of walnut tree. (a–c) respectively show tree structures of different complexity.

2.2. Point Cloud Segmentation

2.2.1. Semantic Segmentation Based on Walnut-PointNet

PointNet++ is an extension of PointNet designed for classification and segmentation tasks on irregular and unordered point clouds. By utilizing adaptive sampling and local feature learning, PointNet++ effectively captures the geometric information of point clouds. However, it still has limitations in feature extraction and capturing long-range dependencies in complex structures, especially for unevenly distributed point clouds. To enhance feature extraction and improve the semantic segmentation performance of walnut tree point clouds, this study proposed the Walnut-PointNet model, which optimized PointNet++ by introducing the channel attention and multi-scale aggregation MLP (CAMA-MLP) module and the central point feature (CPF), as illustrated in Figure 2.

The network was primarily composed of the Set Abstraction (SA) module and the Feature Propagation (FP) module. The input to the network was represented as $(N, d + c)$, where N denotes the number of points in the point cloud, d represents the coordinate dimensions (typically (x, y, z) in 3D), and c represents additional features (such as color, normal vectors, etc.). The PointNet layer extracted local features from the sampled points, and this study introduced CAMA-MLP and CPF for optimization. The FP module included interpolation and unit PointNet. The interpolation was used to recover features of the points that were overlooked during downsampling, while the unit PointNet was responsible for updating the feature vector of each point. The SA and FP modules were connected through skip links.

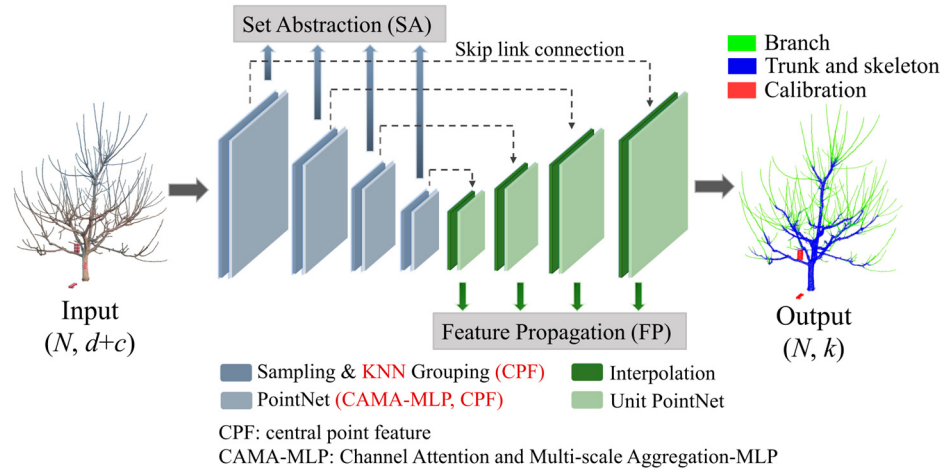


Figure 2. Walnut-PointNet network structure.

(1) Introduction of CAMA-MLP

To address the limitations of PointNet++, Ma et al. [39] proposed that detailed local geometric information may not be the key for point cloud analysis. Consequently, they introduced a pure residual MLP network called PointMLP, which did not rely on complex local geometric extraction modules but achieved superior performance through a lightweight geometric affine module. Inspired by PointMLP, this study introduced CAMA-MLP, which incorporated innovative local feature aggregation and channel attention. The module consisted of three key components: basic feature transformation, local feature aggregation, and the channel attention mechanism, as illustrated in Figure 3.

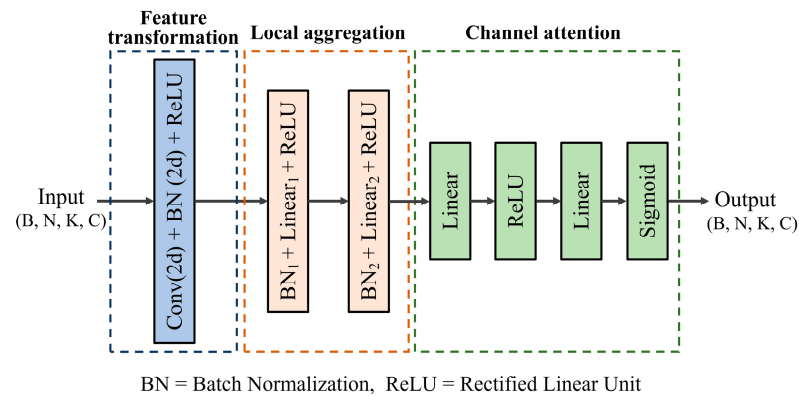


Figure 3. CAMA-MLP structure.

First, a series of 2D convolutional layers and batch normalization layers were applied for basic feature transformation. The input feature tensor x had a shape of (B, N, K, C) , where B was the batch size, N was the number of points, K was the number of neighboring points for each point, and C was the number of channels. After the 2D convolution operation, the dimensions were rearranged to (B, C, N, K) . Batch normalization and the ReLU activation function were applied to perform an initial nonlinear feature transformation, as shown below:

$$\Phi_{pre}(f_{i,j}) = \text{ReLU}(\text{BN}(\text{Conv}_1(f_{i,j}))) \quad (1)$$

where Conv_1 represents the 2D convolution operation, BN denotes the 2D batch normalization operation, $\text{ReLU}(\cdot)$ is the nonlinear activation function, and $f_{i,j}$ represents the features of the i th point and its neighboring point j .

Subsequently, the features were transposed back to the shape $[B, N, K, C]$, followed by the introduction of a local feature aggregation module. The module consisted of two linear transformation layers (Linear_1 and Linear_2), each maintaining the same number of feature channels. These linear layers performed nonlinear mappings to preserve feature space consistency. To retain spatial structural information, 2D batch normalization layers (BN_1 and BN_2) replaced traditional 1D batch normalization, ensuring effective capture of the spatial distribution characteristics of point cloud data during normalization. Additionally, a residual connection was applied after each linear transformation, where the output was added to the input (identity mapping). This helped mitigate the vanishing gradient problem in deep networks and accelerated training. The two linear transformations and residual connections are formulated as follows:

$$o_1 = \text{ReLU}(\text{Linear}_1(\text{BN}_1(x)) + x) \quad (2)$$

$$o_2 = \text{ReLU}(\text{Linear}_2(\text{BN}_2(o_1)) + o_1) \quad (3)$$

where x represents the input features, $\text{BN}_1(\cdot)$ and $\text{BN}_2(\cdot)$ are the first and second batch normalization layers, respectively, and $\text{Linear}_1(\cdot)$ and $\text{Linear}_2(\cdot)$ are the first and second linear transformation layers. o_1 and o_2 represent the output features after the first and second transformations, respectively.

Finally, the module introduced a channel attention mechanism that dynamically optimized feature channels through adaptive weight learning. A max-pooling operation was applied along the K -dimension, producing a feature map of shape (B, N, C) . Next, a two-layer fully connected network, including dimensionality reduction and expansion, was used to learn the channel weights. Specifically, the first fully connected layer reduced the number of channels from C to $C/4$, emphasizing key inter-channel relationships while reducing computational cost. A ReLU activation function was applied in between to introduce nonlinear transformations, as shown in Equation (4). The second fully connected layer restored the channel count back to C and applied a Sigmoid activation function to normalize the channel weights within the range $[0, 1]$, obtaining the importance weights α for each channel, as shown in Equation (5). Finally, the attention weights were applied to the original features through broadcasting, enabling adaptive weighting of feature channels, as shown in Equation (6). In this way, the model can dynamically adjust the importance of different channels based on the input data.

$$h_1 = \text{ReLU}(W_1 x_{\text{pool}} + b_1) \quad (4)$$

$$\alpha = \sigma(W_2 h_1 + b_2) \quad (5)$$

$$x_{\text{out}} = x \odot \alpha \quad (6)$$

where h_1 represents the output from the first fully connected network, α denotes the channel importance weights, W_1 and W_2 are the weight matrices of the two fully connected layers, b_1 and b_2 are the bias terms, σ represents the Sigmoid activation function, $\odot$ denotes element-wise multiplication, and x_{out} is the weighted output feature.

In recent years, several attention mechanisms have been proposed for point cloud analysis. PointNet++ employs a hierarchical structure to capture local features at multiple scales, but it relies on neighborhood aggregation, which struggles with long-range dependencies. Self-attention mechanisms in Point Transformer models capture global dependencies through pairwise interactions, but their quadratic complexity incurs significant computational overhead [40]. Most of them primarily focus on capturing global relationships among all points, which increases computational complexity and memory

usage. In contrast, CAMA-MLP offers a more efficient solution by combining local feature aggregation with channel attention. This design allows for dynamic adjustment of feature channel importance, addressing the challenge of feature weighting while avoiding the high computational cost associated with full self-attention mechanisms. As a result, CAMA-MLP achieves both high performance and enhanced computational efficiency.

(2) Introduction of central point feature

In feature extraction, the original PointNet++ primarily encodes local geometric information using the coordinate offsets of sampled points relative to the central point. While this method captures basic geometric structures, it has limitations in capturing the local semantic information. In this study, while retaining the original relative coordinate features, we introduced the central point feature (CPF), allowing the model to better capture both local and global information of point clouds.

Farthest Point Sampling (FPS) was used to select central points from the original point cloud. In the original model, only the coordinates of the central points were extracted. Building on this, the study further extracts the features of the central points, preserving more detailed information on key points. According to the uneven distribution of branch point clouds, the study replaced the ball query in the original model with k -nearest neighbors (KNN) to identify neighboring points around each central point, as shown below:

$$\mathcal{N}(q_i) = \text{top-}k \left(\left\{ p_j \in P : \sqrt{\sum_{d=1}^D (q_i^d - p_j^d)^2} \right\}, k \right), i \in \{1, 2, \dots, M\}, j \in \{1, 2, \dots, N\} \quad (7)$$

where $\mathcal{N}(q_i)$ represents the index set of the k -nearest neighbors of the query point q_i , P denotes all points in the point cloud, and D is the feature dimension of the points (for 3D point cloud, $D = 3$). M represents the number of query points, and N is the total number of points in the point cloud.

After KNN grouping, the coordinates and features of the neighboring points relative to the central point were obtained to form the neighborhood features. These features were then concatenated with CPF along the channel dimension. The neighborhood features continued to encode the local spatial structure, while CPF retained the original attributes, such as color and normal vectors. This fusion enhanced the representation of both point cloud semantics and global information.

2.2.2. Point Cloud Annotation and Network Training

In this study, CloudCompare v2.6.3 software was used to annotate walnut tree point clouds, creating a 3D semantic segmentation dataset. The walnut tree was classified into three categories: Trunk, Branch, and Calibration, assigned scalar values of 0, 1, and 2, respectively. As shown in Figure 4, the Trunk category includes the central main trunk and skeleton branches, while the Branch category represents outer branches—annual shoots connected to the trunk or skeleton branches at only one end, which are typically the primary targets for pruning. Since multi-view image-based reconstruction methods struggle to directly capture the object's scale, and to facilitate branch orientation determination, this study placed calibration objects near the tree as scale and orientation references, categorized as Calibration. The annotating results were saved in TXT format, consisting of seven columns: x , y , z , N_x , N_y , N_z and semantic label values. x , y , and z represent the 3D coordinates, and N_x , N_y , and N_z represent the normal vector coordinates, creating a semantic segmentation dataset consisting of 100 walnut tree point clouds. In total, 70% was randomly allocated for training, while the remaining 30% was allocated for testing. All experiments were conducted on a computer equipped with an Intel i7-12700KF CPU, an NVIDIA GeForce RTX 3060 GPU, and 32GB of RAM running on Windows 10. The

network model was developed using PyTorch 1.13.1. The model was trained for 80 epochs, with a batch size of 64 and an initial learning rate of 0.001.

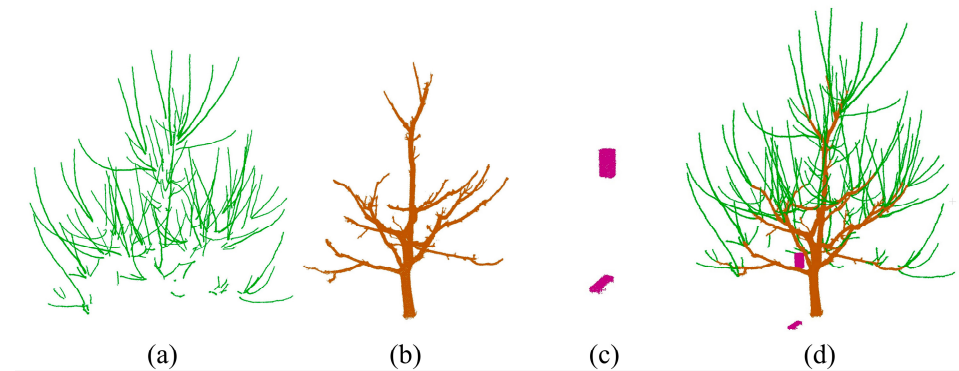


Figure 4. Annotated examples. (a) Branch. (b) Trunk. (c) Calibration. (d) Segmentation.

2.2.3. Performance Evaluation

Overall accuracy (OA), class average accuracy (ACC), and mean IoU (mIoU) were adopted to evaluate the performance of the trained Walnut-PointNet for segmenting the Trunk, Branch, and Calibration of an individual walnut tree. OA indicates the proportion of correctly predicted points, as shown in Equation (8). ACC is the average of the accuracies for all categories, assessing the model's classification precision for each class, as shown in Equation (9). mIoU measures the overlap between the predicted and ground truth classes. A higher IoU indicates that the model's prediction is closer to the actual target, with the calculation as shown in Equation (10).

$$OA = \frac{\sum_{i=1}^m TP_i}{\sum_{i=1}^m N_i} \quad (8)$$

$$ACC = \frac{1}{m} \sum_{i=1}^m \frac{TP_i}{N_i} \quad (9)$$

$$mIoU = \frac{1}{m} \sum_{i=1}^m \frac{TP_i}{TP_i + FP_i + FN_i} \quad (10)$$

where m is the number of classes and N_i is the number of all points of this class. The TP , TN , FP and FN denote true positives, true negatives, false positives, and false negatives, respectively.

2.3. Branch Clustering

To further identify and separate individual pruned branches, the DBSCAN algorithm was employed for clustering analysis of the segmented branch point cloud data. By setting a neighborhood radius (ϵ) and a minimum number of points (MinPts), the DBSCAN algorithm identifies regions with high local density and groups neighboring points within these regions into the same cluster. Specifically, when a point's neighborhood contains a sufficient number of points, it is considered a core point, and its neighboring points are assigned to the same cluster. Points that are not core points but are adjacent to a core point are also included in the cluster. By segmenting the point cloud of different branches into distinct clusters based on local density, each branch to be pruned was effectively separated. In total, 20 walnut trees were randomly selected, and the clustering results of the

pruned branches were compared with the actual number of branches to assess the accuracy of the clustering analysis.

2.4. Pruning Method

The walnut tree has strong branching ability, easily forming a distinct main trunk and canopy structure. The branches exhibit significant apical dominance, leading to the formation of upright and vigorous one-year-old branches that grow rapidly. These branches are typically long and thin, which may impact the canopy's ventilation, light transmission, and reproductive growth. For dormant walnut trees without leaves, the primary goal of pruning is to alter the apical dominance, encouraging outward expansion. This helps maintain the balance between vegetative and reproductive growth, thereby improving the overall health and vigor of the tree [41]. Manual pruning typically relies on the subjective experience of fruit farmers, lacking clear and unified pruning standards. Through consulting experts and reviewing relevant pruning agricultural guidelines, the following pruning rules have been summarized: (1) Retain the smaller, fruit-bearing branches from the previous year as much as possible. These branches typically yield more fruit and possess higher fruiting potential. This approach supports the optimization of the tree's reproductive growth. By ensuring that the tree continues to fruit from branches with the greatest potential, overall yield can be enhanced. (2) A short-cutting approach is usually applied, focusing on pruning the upright, vigorous branches that grew the previous year. These branches often exhibit strong apical dominance, leading to rapid growth and suppression of lateral branch development. By shortening these branches, apical dominance is effectively reduced, encouraging the growth of lateral branches and enhancing the overall canopy structure. This approach helps maintain a balanced and open tree crown. (3) Due to variations in tree vigor, the amount of pruning should be adjusted according to the tree's condition. When the tree is vigorous, excessively strong branches drain too many nutrients and energy, hindering the tree's ability to allocate sufficient resources to the main growth points and fruit development. In such cases, pruning should be increased to redirect energy toward the most productive areas for growth and fruiting, thus enhancing both yield and fruit quality. Conversely, when the tree is weaker, pruning should be reduced to prevent the loss of too many nutrient sources, ensuring sufficient support for the next growth cycle.

2.5. Pruning Points Positioning

2.5.1. Extraction of Branch Parameter

After identifying individual pruning branches, key parameters were automatically extracted to determine the pruning position. Walnut tree pruning branches are typically slender and elongated, with measurements showing that their diameters generally range between 9 and 12 cm, exhibiting minimal variation from base to tip. As a result, diameter differences were ignored, and branch length was primarily used to determine the pruning location. Since pruning branches exhibit an upright and slender morphology, their point cloud clusters display distinct characteristics along the main axis. Therefore, Principal Component Analysis (PCA) was employed to extract the principal axis direction of the point cloud, representing the primary growth direction of the branch by projecting the original high-dimensional data into a lower-dimensional space.

First, the point cloud data were centralized, as shown in Equation (11), by subtracting the mean of each dimension from the corresponding feature values to ensure a zero-mean distribution. Next, the covariance matrix of the data was constructed and subjected to eigenvalue decomposition. The eigenvalues represented the variance of the data along their corresponding eigenvectors, with the eigenvector associated with the largest eigenvalue

indicating the principal axis direction of the branch point cloud. The calculations are shown in Equations (12) and (13).

$$X' = X - \mu \quad (11)$$

$$H = \frac{1}{n-1} X'^T X' \quad (12)$$

$$Hv = \lambda v \quad (13)$$

where the branch point cloud matrix is denoted as $X_{n \times 3}$, and $\mu = \frac{1}{n} \sum_{i=1}^n x_i$ represents the mean of each column feature. H is the covariance matrix, v denotes the eigenvectors, and λ represents the eigenvalues.

Since branches typically grow obliquely upward, the branch base point, p_{root} , was identified as the point with the smallest Z-coordinate. From this base, the point cloud data were projected onto the principal axis direction, as shown in Equation (14), where v_1 is the unit vector of the principal axis. The branch length, l , was then calculated as the difference between the maximum and minimum projection coordinates, as shown in Equation (15).

$$t = (X' - p_{\text{root}}) \cdot v_1 \quad (14)$$

$$l = \max(t) - \min(t) \quad (15)$$

Since the walnut tree point cloud was obtained through 3D reconstruction from multi-view images, the extracted dimensions of the point cloud did not directly correspond to real-world measurements. To establish the relationship between the real-world scene and the reconstructed point cloud, a scale factor must be calculated based on the true size of the calibration. This allowed for the extraction of the actual branch parameters in the real world. The calculations are shown in Equation (16), where λ represents the scale factor, l_{m0} and l_{T0} are the extracted and true values of the calibration, and l_m and l_T represent the extracted and actual lengths of the branches.

$$\begin{cases} \lambda = \frac{l_{m0}}{l_{T0}} \\ l_T = \lambda \times l_m \end{cases} \quad (16)$$

The lengths of 56 branches were randomly selected and manually measured using a caliper and tape. The automatically extracted values were then compared with the manual measurements, and evaluated using mean absolute error (MAE), as defined in Equation (17).

$$MAE = \frac{1}{n} \sum_{i=1}^n |e_i - m_i| \quad (17)$$

where n represents the sample size, e_i is the extracted parameter value for the i th sample, and m_i is the manually measured value for the i th sample.

2.5.2. Pruning Points Determination

To meet the demands of intelligent pruning operations, a quantifiable intelligent pruning method needs to be developed, in conjunction with traditional manual pruning rules. The one-year-old branches of walnut trees are typically thin and elongated, with diameters mainly ranging from 9 to 12 mm, exhibiting minimal variation along the length. As a result, pruning positions and amounts were primarily determined by branch length.

Walnut trees were classified into the initial fruiting stage and peak fruiting stage according to the complexity of their tree structure. Based on experimental findings, branches with fewer than 80 were categorized as the initial fruiting stage, whereas those with more than 80 were categorized as the peak fruiting stage. The pruning targets were one-year-old, upright, vigorous branches, and the method involved short-cutting the tips of these branches. By reviewing branch characteristics, pruning guidelines, and considering actual conditions, the pruning amounts were classified according to branch length, as shown in Tables 1 and 2.

Table 1. Classification of pruned branches for walnut trees in the peak fruiting stage.

Classification	Length	Short-Cutting Type	Retention Length
Water sprout	Above 70 cm	Heavy	Retain 40 cm
Long branch	30–70 cm	Medium	Retain 30 cm
Medium branch	15–30 cm	Light	Retain 15 cm
Short branch	5–15 cm	No	-

Table 2. Classification of pruned branches for walnut trees in the initial fruiting stage.

Classification	Length	Short-Cutting Type	Retention Length
Water sprout	Above 60 cm	Heavy	Retain 50 cm
Long branch	40–60 cm	Medium	Retain 40 cm
Medium branch	15–40 cm	No	-
Short branch	5–15 cm	No	-

Based on the intelligent pruning rules above, the branches to be pruned were classified into four types—water sprout, long branch, medium branch, and short branch. Each classification was assigned a specific short-cutting type and pruning length for precise localization of pruning points. Since trees in the initial fruiting stage have just started to bear fruit, they require lighter pruning. The medium branch was preserved to promote balanced growth and facilitate the development of future fruit-bearing branches, thereby ensuring a healthy structure that enhances fruit production. Additionally, the system outputs the branch parameters and pruning types for each point cloud cluster, offering detailed information for subsequent analysis.

3. Results

3.1. Semantic Segmentation Result

3.1.1. Ablation Experiments for Method Effectiveness Verification

As an optimized version of the original PointNet++, an ablation study was conducted to validate the effectiveness of each component in the proposed optimization method. The study compared the performance of models with different components under the same experimental environment, including: (1) PointNet++ (baseline); (2) model with only CAMA-MLP; (3) model with only CPF; (4) Walnut-PointNet (ours). Figure 5a shows the training curve of PointNet++. The loss curve increased with the number of epochs, indicating overfitting during the training process. Additionally, its OA, ACC, and mIoU metrics fluctuated around lower values without significant improvement, suggesting that the model struggled to learn and generalize effectively to the walnut tree point cloud dataset, exhibiting poor adaptability. Figure 5b–d further compare the OA, ACC, and mIoU curves of different models during training and evaluation. Each component of CAMA-MLP and CPF individually improved the model's performance. With the combined effect of CAMA-MLP and CPF, Walnut-PointNet significantly outperformed other models across

all metrics, converging around 50 epochs, demonstrating faster convergence and greater stability.

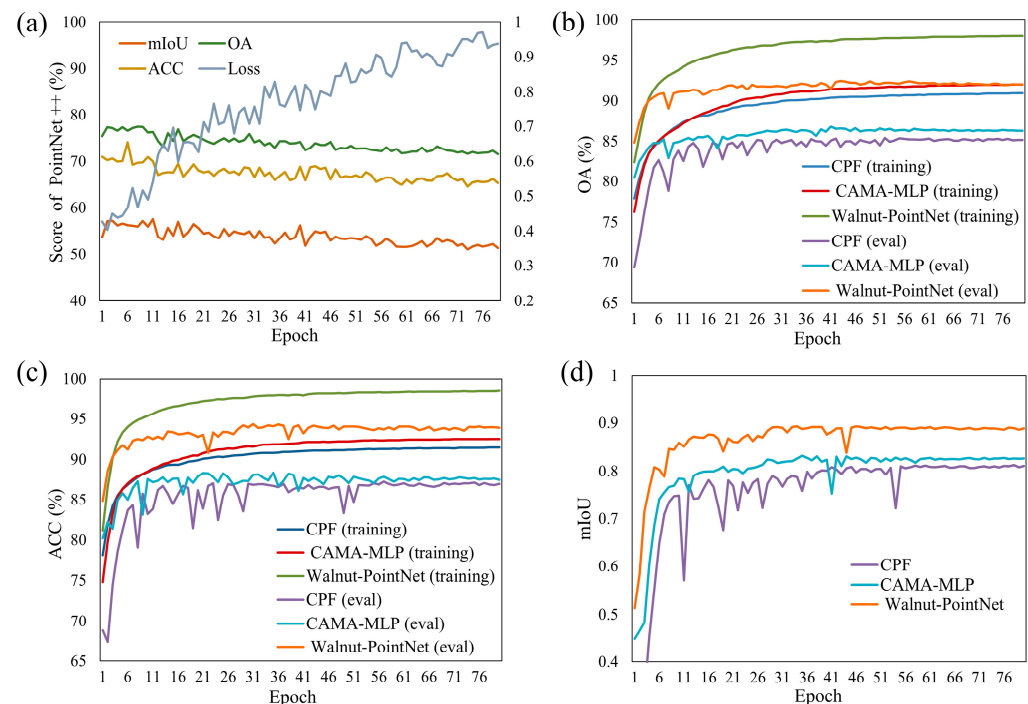


Figure 5. Comparison of different models during training in ablation study. Training and eval represent the training curve and evaluation curve, respectively. (a) shows the training curve of PointNet++; (b–d) represent the OA, ACC, and mIoU curves of different models, respectively.

To further evaluate the performance of different models in the ablation study, a comparison was conducted on four models using the test dataset. Table 3 presents the OA, ACC, IoU for different classes, and mIoU for each model on the test set. The results showed that the baseline model, PointNet++, performed poorly across all metrics, indicating its limited feature extraction capability on the complex walnut tree point clouds. In contrast, incorporating CPF and CAMA-MLP components significantly improved the model's segmentation accuracy. Walnut-PointNet achieved the best performance across all evaluation metrics, with an OA of 93.39%, an ACC of 95.29%, and an mIoU of 0.912, representing improvements of 21.1%, 34.95%, and 53.79%, respectively, compared to the original PointNet++ model. These results demonstrated that the optimization strategy effectively enhanced segmentation performance on walnut tree point clouds.

Table 3. Comparison of different models in tests in ablation study.

Model	OA	ACC	IoU ₁	IoU ₂	IoU ₃	mIoU	Epoch
PointNet++	77.12	70.61	0.650	0.608	0.520	0.593	80
with CPF	86.41	88.26	0.819	0.764	0.916	0.833	80
with CAMA-MLP	87.54	88.79	0.831	0.804	0.909	0.848	80
Walnut-PointNet (ours)	93.39	95.29	0.887	0.853	0.995	0.912	80

Note: Bold indicates the best performance. IoU₁, IoU₂, and IoU₃ represent the IoU values for the Trunk, Branch, and Calibration categories, respectively.

3.1.2. Branch Segmentation Evaluation and Visualization

Figures 6 and 7 present the segmentation results for the entire walnut tree and for specific branch areas. From the results in Figures 6b and 7b, it was evident that PointNet++

suffered from significant errors in branch segmentation, leading to noticeable misclassifications in certain areas. The model struggled with dense branches, particularly at the top of the canopy, where branches and the trunk were not correctly distinguished. This caused entire branch clusters and their connecting skeleton branches and trunks to be incorrectly classified as the same category. In contrast, Figures 6c and 7c showed that Walnut-PointNet outperformed PointNet++ in segmenting the trunk and branches, with its segmentation results closely matching the manually labeled dataset. Walnut-PointNet achieved more accurate segmentation, especially at the junctions between branches and the trunk, as well as in dense canopy areas, effectively reducing misclassifications between the two. Moreover, Walnut-PointNet captured more intricate branch details, ensuring accurate identification of even smaller branches and preventing errors caused by branch merging.

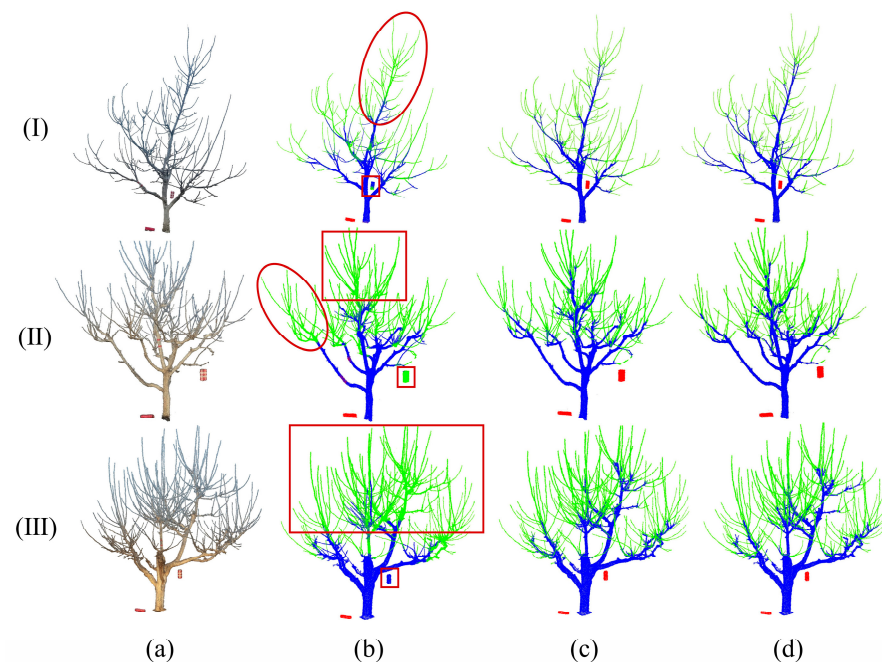


Figure 6. Segmentation visualization of the entire walnut tree. (a) Original point cloud, (b) PointNet++, (c) Walnut-PointNet (ours), and (d) the ground truth (labeled). Green, blue and red represent the classes of Trunk, Branch and Calibration, respectively. In (b), the areas outlined in red indicate significant segmentation failures.

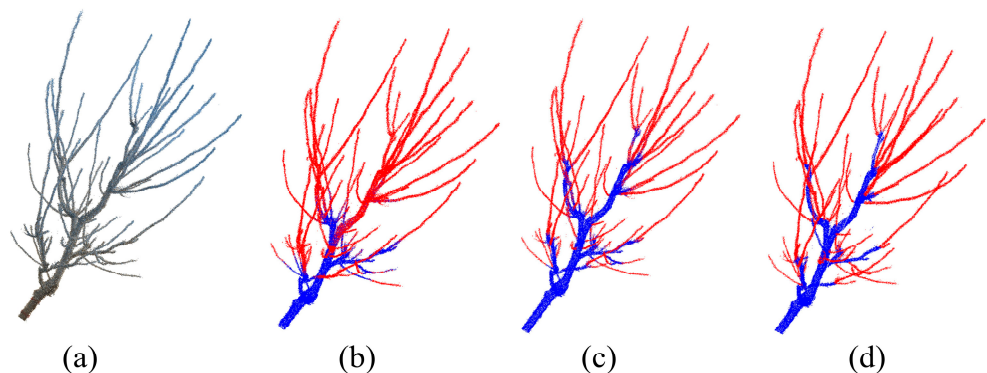


Figure 7. Segmentation visualization of local branches. (a) Original point cloud, (b) PointNet++, (c) Walnut-PointNet (ours), and (d) the ground truth (labeled). In (b–d), blue and red represent the classes of Trunk and Branch, respectively.

3.2. Analysis of the Clustering Result

DBSCAN algorithm was used to cluster the segmented pruning branches in order to extract individual branches. In this experiment, the neighborhood radius was set to 0.01 m, and the minimum points threshold was set to 10. In total, 30 trees were randomly selected for the clustering analysis. The final clustering output consisted of two categories: branch clusters and noise clusters. The noise clusters were treated as invalid data. For the branch clusters, each cluster was considered an individual pruning branch and assigned a random color for distinction. The clustering results, shown in Figure 8, demonstrated that most branches were successfully clustered and clearly distinguished.

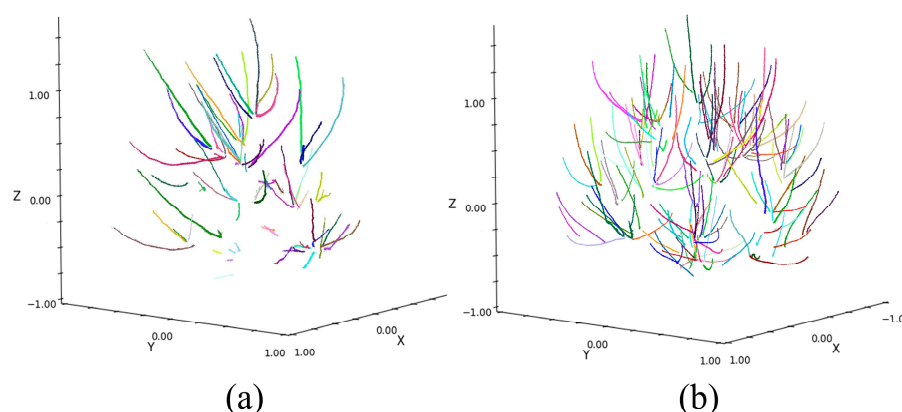


Figure 8. Clustering of walnut tree pruned branches. (a) Walnut tree with 65 branches (initial fruiting stage); (b) walnut tree with 158 branches (peak fruiting stage).

However, the complex structure of walnut trees, with densely arranged and interwoven branches, presents significant challenges in clustering individual branches. Figure 9 illustrates typical examples of erroneous clustering. The intersection of branches may cause adjacent branches to be incorrectly assigned to the same cluster (Figure 9a–c). Similarly, for branches growing close together on the same primary branch, DBSCAN might misclassify them as one cluster due to their small basal spacing, incorrectly grouping multiple branches (Figure 9d–f). Additionally, gaps in the branch point cloud may cause points from the same branch to be split into separate clusters (Figure 9g,h). These factors affected the accuracy and reliability of the clustering process. Despite these errors, the method still effectively identified and separated the most branches. Figure 10 demonstrates the relationship between the clustered pruning branches and the actual number of branches. Most points fell within the 95% confidence interval, and the coefficient of determination (R^2) was 0.936, indicating strong consistency between the clustering results and the actual values. Overall, this study successfully detected and clustered most pruning branches, laying a foundation for subsequent pruning point localization.

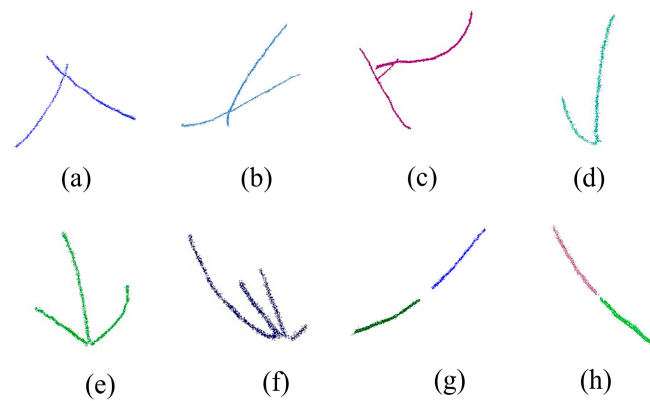


Figure 9. Examples of clustering errors. (a–c) Intersecting branches. (d–f) Branches connected at the base. In (g) and (h), branches are mistakenly split into two clusters.

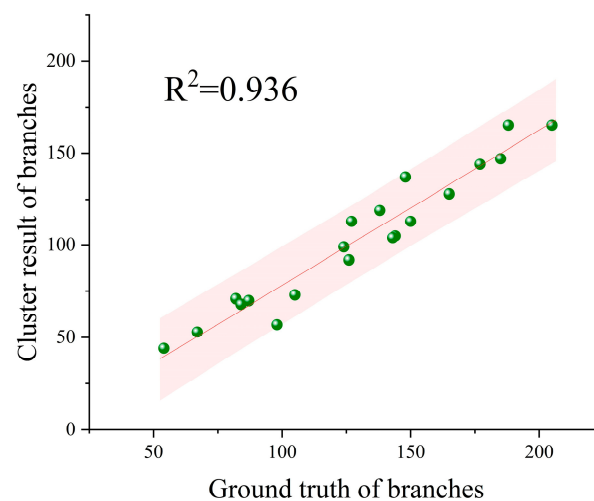


Figure 10. Comparison between the clustered branch count and the actual branch count.

3.3. Results of Branch Parameter Extraction

The comparison between the automatic extraction values and manual measurements for branch lengths in the short-growth branch group (long, medium, and short branches) and water sprout is shown in Figure 11. The branch lengths in the short-growth group ranged from 100 to 700 mm, with an MAE of 28.04 mm and an R^2 value of 0.971, indicating high accuracy in the automatic extraction method. In contrast, water sprouts had a broader length range of 700 to 1500 mm, with an MAE of 50.11 mm and an R^2 value of 0.959. Although the MAE for water sprouts was slightly higher than that of the short-growth branch group, the relative error remained reasonable due to the longer length of the water sprouts, still meeting the requirements for pruning operations. Compared to traditional manual pruning, which relied on visual estimation and experience, this method offered smaller errors and accurately extracted branch lengths across a broader range, enhancing the precision of intelligent pruning operations.

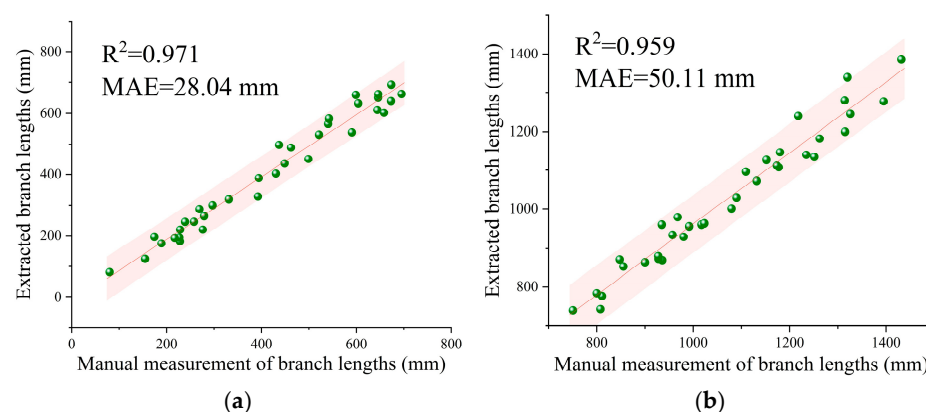


Figure 11. Comparison of automatically extracted and manually measured values of pruning branch length. (a) Short-growth branch group (long, medium, and short branches). (b) Water sprout.

3.4. Results of Pruning Points Positioning

Based on the pruning rules, branches were classified by length, and the corresponding pruning point positions were determined for each type. The pruning points were assigned different colors based on the pruning type and added to the output point cloud for visual display and quantitative assessment, as shown in Figure 12.

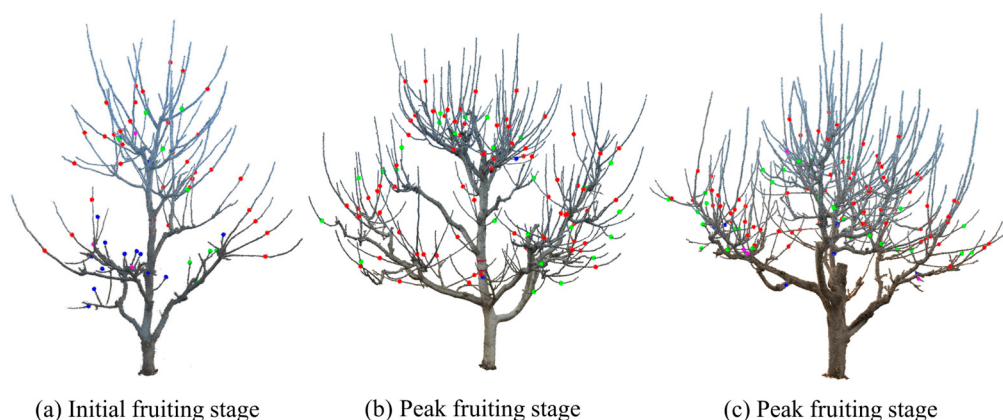


Figure 12. Identification of pruning points in walnut trees with different levels of complexity. Note: Red points indicate pruning points for water sprouts, green points for long branches, and blue points for medium branches. Purple points indicate short branches, which are not pruned. (a) represents the initial fruiting stage, (b,c) represent the peak fruiting stage.

In total, 10 walnut trees were randomly selected to compare the predicted result with the actual result (based on manual experience and annotations from agricultural experts), as shown in Figure 13. The average pruning point identification success rate was 89.33%. For walnut trees with complex structures, there was a certain discrepancy between the predicted and actual pruning point numbers. However, for most walnut trees, the prediction success rate for pruning points was high, especially for Tree 1 and Tree 4, where the success rates reached 94.83% and 92.75%, respectively, indicating good accuracy.

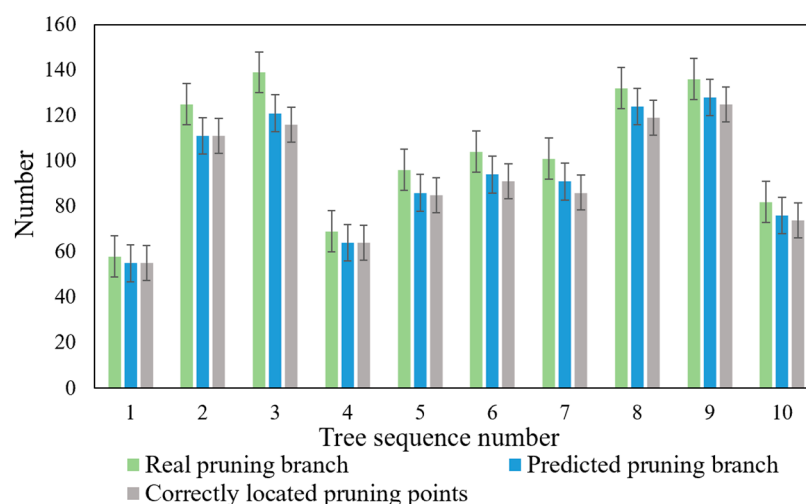


Figure 13. Pruning points identification results.

To further evaluate the accuracy of pruning point localization, the predicted pruning point positions were compared with the actual manual pruning results, as shown in Figure 14. The results indicated that the predicted pruning points generally aligned with the actual manual pruning locations, and the post-pruning tree structure was similar, indicating the method's effectiveness in facilitating pruning decisions. Although tree pruning lacks strictly fixed rules, and differences in experience and judgment may lead to variations among different pruners, the pruning point localization method proposed in this study still demonstrated high accuracy and consistency, further validating its feasibility.

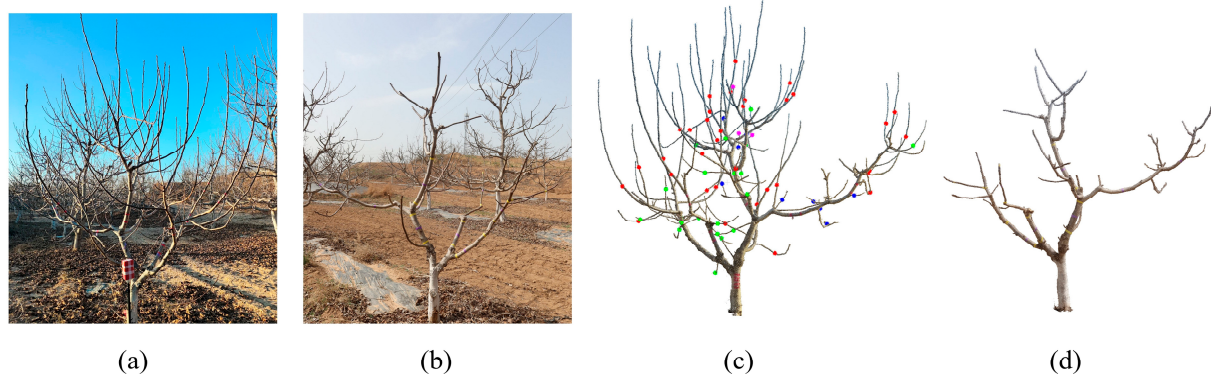


Figure 14. Comparison of predicted pruning points with actual manual pruning. (a) Photo of the walnut tree before pruning. (b) Photo of the walnut tree after actual manual pruning. (c) Predicted pruning points. (d) Walnut tree after actual manual pruning.

4. Discussion

The study results revealed that the proposed method performed excellently in pruning branch identification and pruning point localization, demonstrating high accuracy. The following sections describe the effects of various factors on the results and the advantages and limitations of the method for practical applications.

4.1. Accuracy and Effects of Branch Semantic Segmentation

Our proposed Walnut-PointNet outperformed the baseline model PointNet++ and models with individual optimization components on the walnut tree point cloud dataset, demonstrating the effectiveness of the proposed optimization strategies. The experimental results showed that PointNet++ suffered from overfitting during training due to

its limited feature extraction capability, resulting in low segmentation accuracy on the test data, with an mIoU of only 0.593. The possible reasons were as follows: The walnut tree point cloud was divided into three categories—Trunk, Branch, and Calibration. The trunk and skeleton branches usually occupied larger spatial regions with higher point cloud density, while pruning branches had fewer points that were more sparsely distributed. Calibration objects had even fewer points. This imbalance in distribution affected the model's feature extraction, causing it to lean toward the Trunk categories during training. Additionally, PointNet++ employed a random sampling strategy for point cloud data processing, which can lead to under-sampling of minority categories. This weakened the model's ability to learn features from underrepresented categories, making it harder to generalize on imbalanced datasets. Moreover, the high geometric similarity between the trunk and branches made it more difficult to distinguish between categories.

In the optimization strategy, we utilized the KNN grouping method, which enabled the model to adaptively select neighboring points during feature extraction. This allowed the model to better capture the local structural features of the sparsely distributed pruning branch category, improving its ability to handle imbalanced data. Additionally, we incorporated CAMA-MLP, which leveraged a channel attention mechanism to strengthen the extraction of key features. The model can adaptively adjust the weights of each channel, enabling it to focus more precisely on the differences between categories, thereby enhancing the accuracy of feature extraction. Compared to the original PointNet++ model, which relied solely on relative coordinates for feature representation, we introduced the central point feature during the feature extraction process. This integration of geometric and semantic information allowed for a more comprehensive representation of both local and global semantic features.

Branch segmentation in individual trees was more challenging due to the morphological similarity of branches, requiring higher accuracy compared to semantic segmentation of individual trees in forest scenes [42,43]. Overall, our optimized model achieved significant improvements in segmentation accuracy and generalization on the walnut tree dataset, particularly in handling unevenly distributed and complexly shaped pruning branches. It can more precisely capture the local structural features of such branches, demonstrating superior robustness and adaptability.

4.2. Reliability and Effects of Pruning Points Positioning

The pruning point localization method proposed in this study is closely aligned with actual manual pruning results. While manual pruning lacks a unified quantitative standard and relies more on expert judgment, we simplified and adjusted the traditional rules to make them compatible with intelligent algorithms. The overall goal is to control the pruning amount based on branch length, ensuring that the pruning position falls between one-half and two-thirds of the length of vigorous branches, which serves as a key factor in expert judgment. The identification of all pruning branches and localization of pruning points for the entire walnut tree can be completed in about 16 s. Compared to methods based on local depth images [29], this approach offers a wider recognition range and greater overall efficiency.

The final pruning point localization success rate reached 89.33%. The unsuccessful cases were primarily influenced by the following factors: (1) Due to the morphological similarity of branches, some pruning branches may have been misclassified as the main trunk during the semantic segmentation. (2) The densely arranged and interwoven branches may lead to incorrect branch association during the clustering process. Specifically, when adjacent branches cross or grow close together, their intersecting points or small basal spacing may cause them to be incorrectly grouped as a single branch. Future work could explore more refined clustering techniques, such as considering the angle of

inclination or spatial distance between branches to better differentiate adjacent branches. Additionally, integrating spatial topology information or skeleton extraction could enhance the understanding of the spatial relationships and positions of branches. (3) There were some errors in branch length extraction. This study estimated length based on the primary axis direction of the branches; however, the dense distribution and complex structure of walnut tree branches, along with some branches growing in a curved pattern, may have caused extraction inaccuracies. Despite these challenges, the proposed method achieved a higher accuracy compared to the skeleton extraction-based pruning point localization method for apple trees [17]. This study determined pruning points through global 3D point cloud analysis, offering enhanced comprehensiveness and spatial consistency compared to methods that depend on local RGB and depth images of fruit trees [29,37]. The global 3D point cloud can fully capture the tree structure, mitigating recognition biases caused by limited local information.

4.3. Summary and the Future Work

In summary, the study offers the following advantages in intelligent pruning technology for walnut trees: In terms of point cloud semantic segmentation, we proposed the WalnutPointNet model, specifically designed to address the unique characteristics of walnut trees for precise pruning branch identification. Additionally, we developed a pruning point localization algorithm based on walnut tree pruning guidelines. Compared to existing methods based on 2D images, our approach captured richer 3D information. Furthermore, compared to methods using depth cameras to acquire depth data, it provided more accurate and comprehensive spatial structural information, without being limited by distance thresholds.

Of course, this study also has several limitations, as mentioned above. Firstly, future research will further optimize parameter extraction methods by incorporating branch characteristics, such as curvature and diameter, to enhance recognition and localization accuracy. In addition, the pruning rules can be further enhanced by incorporating additional factors, such as detecting dead branches, thereby broadening the scope of pruning targets. This will necessitate the construction of more diverse datasets. Moreover, due to seasonal factors, we have not been able to conduct a more accurate quantitative evaluation of actual pruning, as pruning only occurs once a year. In future research, we plan to expand the ground truth data provided by experts to more effectively evaluate the performance of the algorithm and ensure a more comprehensive assessment. Furthermore, the system has been successfully implemented for identification purposes currently. To conduct field testing for practical applications, further research is needed to integrate the algorithms with robotic arms, end-effectors and path-planning components to perform actual pruning operations. Finally, future research could apply the approach to other fruit tree varieties, thereby broadening the scope of the study and increasing its applicability in agriculture.

5. Conclusions

This study proposed a method for identifying pruning branches and locating pruning points in walnut trees using 3D point cloud semantic segmentation. The deep learning method was employed to segment the tree into three categories—Trunk, Branch, and Calibration—enabling pruning branch identification. Additionally, pruning branch lengths were extracted, and pruning point locations were determined according to established pruning rules. The specific conclusions of this study are as follows:

(1) By optimizing the original PointNet++ model, incorporating CAMA-MLP with a channel attention mechanism, and adding central point features, we proposed the Walnut-PointNet model. On the walnut tree point cloud dataset, the OA, ACC, and mIoU

reached 93.39%, 95.29%, and 0.912, respectively. Compared to PointNet++, these values improved by 21.1%, 34.95%, and 53.79%, significantly enhancing the accuracy of walnut tree point cloud segmentation and pruning branch identification.

(2) Pruning rules were established based on the characteristics of walnut tree branches. Ignoring variations in branch diameter, pruning branches were categorized into long, medium, and short branches and water sprouts based on their lengths, with specific pruning quantities assigned to each category.

(3) Clustering analysis was applied to extract individual pruning branches, and PCA was used to extract branch length parameters. Compared to manual measurements, the length extraction values for the short-growth branch group (long, medium, and short branches) with a length range of 100 mm–700 mm, and the water sprouts group (length range 700 mm–1500 mm), achieved R^2 values of 0.971 and 0.959, with MAE of 28.04 mm and 50.11 mm, respectively. This demonstrated that the proposed method provided high accuracy in extracting length parameters.

(4) The pruning branch type was determined based on the extracted branch length, and the pruning point location was identified accordingly. The average success rate of pruning point recognition reached 89.33%, and the post-pruning tree shape was consistent with the manual pruning results. The total time for pruning branch recognition and pruning point localization for the entire tree was approximately 16 s.

(5) This approach successfully enabled accurate identification of pruning branches and localization of pruning points, providing support for intelligent pruning decisions in walnut trees. Moreover, it offered a novel solution for smart pruning techniques in fruit trees.

Author Contributions: Conceptualization, W.Z., X.B. and W.L.; Funding acquisition, W.L.; Investigation, W.Z., X.B. and D.X.; Methodology, W.Z., X.B. and D.X.; Software, W.Z.; Supervision, W.L.; Writing—original draft, W.Z.; Writing—review and editing, W.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the National Innovation Park for Forestry and Grass Equipments R&D Key Project (No. 2024YG06), and the National Key Research and Development Program of China (No. 2018YFD0700601).

Institutional Review Board Statement: Not applicable.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

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

References

1. Zahid, A.; Mahmud, M.S.; He, L.; Heinemann, P.; Choi, D.; Schupp, J. Technological advancements towards developing a robotic pruner for apple trees: A review. *Comput. Electron. Agric.* **2021**, *189*, 106383. <https://doi.org/10.1016/j.compag.2021.106383>.
2. Amatya, S.; Karkee, M.; Gongal, A.; Zhang, Q.; Whiting, M.D. Detection of cherry tree branches with full foliage in planar architecture for automated sweet-cherry harvesting. *Biosyst. Eng.* **2016**, *146*, 3–15. <https://doi.org/10.1016/j.biosystemseng.2015.10.003>.
3. Tabb, A.; Medeiros, H. Automatic segmentation of trees in dynamic outdoor environments. *Comput. Ind.* **2018**, *98*, 90–99. <https://doi.org/10.1016/j.compind.2018.03.002>.
4. Chen, Z.; Ting, D.; Newbury, R.; Chen, C. Semantic segmentation for partially occluded apple trees based on deep learning. *Comput. Electron. Agric.* **2021**, *181*, 105952. <https://doi.org/10.1016/j.compag.2020.105952>.
5. Zhang, X.; Karkee, M.; Zhang, Q.; Whiting, M.D. Computer vision-based tree trunk and branch identification and shaking points detection in Dense-Foliage canopy for automated harvesting of apples. *J. Field Robot.* **2021**, *38*, 476–493. <https://doi.org/10.1002/rob.21998>.

6. Sapkota, R.; Ahmed, D.; Karkee, M. Comparing YOLOv8 and Mask R-CNN for instance segmentation in complex orchard environments. *Artif. Intell. Agric.* **2024**, *13*, 84–99. <https://doi.org/10.1016/j.aiia.2024.07.001>.
7. Sapkota, R.; Karkee, M. Integrating YOLO11 and Convolution Block Attention Module for Multi-Season Segmentation of Tree Trunks and Branches in Commercial Apple Orchards. *arXiv* **2024**, arXiv:2412.05728.
8. Tong, S.; Yue, Y.; Li, W.; Wang, Y.; Kang, F.; Feng, C. Branch Identification and Junction Points Location for Apple Trees Based on Deep Learning. *Remote Sens.* **2022**, *14*, 4495. <https://doi.org/10.3390/rs14184495>.
9. Zhang, C.; Zhang, Y.; Liang, S.; Liu, P. Research on Key Algorithm for Sichuan Pepper Pruning Based on Improved Mask R-CNN. *Sustainability* **2024**, *16*, 3416. <https://doi.org/10.3390/su16083416>.
10. Zheng, Z.; Liu, Y.; Dong, J.; Zhao, P.; Qiao, Y.; Sun, S.; Huang, Y. A novel jujube tree trunk and branch salient object detection method for catch-and-shake robotic visual perception. *Expert Syst. Appl.* **2024**, *251*, 124022. <https://doi.org/10.1016/j.eswa.2024.124022>.
11. Majeed, Y.; Zhang, J.; Zhang, X.; Fu, L.; Karkee, M.; Zhang, Q.; Whiting, M.D. Deep learning based segmentation for automated training of apple trees on trellis wires. *Comput. Electron. Agric.* **2020**, *170*, 105277. <https://doi.org/10.1016/j.compag.2020.105277>.
12. Borrenpohl, D.; Karkee, M. Automated pruning decisions in dormant sweet cherry canopies using instance segmentation. *Comput. Electron. Agric.* **2023**, *207*, 107716. <https://doi.org/10.1016/j.compag.2023.107716>.
13. Li, C.; Pan, Y.; Li, D.; Fan, J.; Li, B.; Zhao, Y.; Wang, J. A curved path extraction method using RGB-D multimodal data for single-edge guided navigation in irregularly shaped fields. *Expert Syst. Appl.* **2024**, *255*, 124586. <https://doi.org/10.1016/j.eswa.2024.124586>.
14. Chen, L.Z.; Lin, Z.; Wang, Z.; Yang, Y.L.; Cheng, M.M. Spatial Information Guided Convolution for Real-Time RGBD Semantic Segmentation. *IEEE Trans. Image Process.* **2021**, *30*, 2313–2324. <https://doi.org/10.1109/TIP.2021.3049332>.
15. Kang, S.; Li, D.; Li, B.; Zhu, J.; Long, S.; Wang, J. Maturity identification and category determination method of broccoli based on semantic segmentation models. *Comput. Electron. Agric.* **2024**, *217*, 108633. <https://doi.org/10.1016/j.compag.2024.108633>.
16. Tan, C.; Sun, J.; Paterson, A.H.; Song, H.; Li, C. Three-view cotton flower counting through multi-object tracking and RGB-D imagery. *Biosyst. Eng.* **2024**, *246*, 233–247. <https://doi.org/10.1016/j.biosystemseng.2024.08.010>.
17. 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. <https://doi.org/10.1016/j.compag.2014.02.013>.
18. Li, L.; Fu, W.; Zhang, B.; Yang, Y.; Ge, Y.; Shen, C. Branch segmentation and phenotype extraction of apple trees based on improved Laplace algorithm. *Comput. Electron. Agric.* **2025**, *232*, 109998. <https://doi.org/10.1016/j.compag.2025.109998>.
19. Sarker, S.; Sarker, P.; Stone, G.; Gorman, R.; Tavakkoli, A.; Bebis, G.; Sattarvand, J. A comprehensive overview of deep learning techniques for 3D point cloud classification and semantic segmentation. *Mach. Vis. Appl.* **2024**, *35*, 67. <https://doi.org/10.1007/s00138-024-01543-1>.
20. Windrim, L.; Bryson, M. Detection, Segmentation, and Model Fitting of Individual Tree Stems from Airborne Laser Scanning of Forests Using Deep Learning. *Remote Sens.* **2020**, *12*, 1469. <https://doi.org/10.3390/rs12091469>.
21. Chen, X.; Jiang, K.; Zhu, Y.; Wang, X.; Yun, T. Individual Tree Crown Segmentation Directly from UAV-Borne LiDAR Data Using the PointNet of Deep Learning. *Forests* **2021**, *12*, 131. <https://doi.org/10.3390/f12020131>.
22. Henrich, J.; van Delden, J. Towards general deep-learning-based tree instance segmentation models. *arXiv* **2024**, arXiv:2405.02061.
23. 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. <https://doi.org/10.1016/j.compag.2021.106484>.
24. 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.
25. Sun, X.; He, L.; Jiang, H.; Li, R.; Mao, W.; Zhang, D.; Majeed, Y.; Andriyanov, N.; Soloviev, V.; Fu, L. Morphological estimation of primary branch length of individual apple trees during the deciduous period in modern orchard based on PointNet++. *Comput. Electron. Agric.* **2024**, *220*, 108873. <https://doi.org/10.1016/j.compag.2024.108873>.
26. Jiang, L.; Li, C.; Fu, L. Apple tree architectural trait phenotyping with organ-level instance segmentation from point cloud. *Comput. Electron. Agric.* **2025**, *229*, 109708. <https://doi.org/10.1016/j.compag.2024.109708>.
27. Guan, H.; Zhang, R.; Lu, T.; Lv, Q.; Ge, B.; Huang, F. PDEC-Net: An Improved Single Tree Segmentation Method for Urban Mobile Laser Scanning Point Clouds Based on PDE-Net. In Proceedings of the IGARSS 2024–2024 IEEE International Geoscience and Remote Sensing Symposium, Athens, Greece, 7–12 July 2024; pp. 8551–8554.

28. Mu, S.; Dai, N.; Yuan, J.; Liu, X.; Xin, Z.; Meng, X. S2CPL: A novel method of the harvest evaluation and subsoil 3D cutting-Point location for selective harvesting of green asparagus. *Comput. Electron. Agric.* **2024**, *225*, 24. <https://doi.org/10.1016/j.compag.2024.109316>.
29. Tong, S.; Zhang, J.; Li, W.; Wang, Y.; Kang, F. An image-based system for locating pruning points in apple trees using instance segmentation and RGB-D images. *Biosyst. Eng.* **2023**, *236*, 277–286. <https://doi.org/10.1016/j.biosystemseng.2023.11.006>.
30. Chen, Z.; Wang, Y.; Tong, S.; Chen, C.; Kang, F. Grapevine Branch Recognition and Pruning Point Localization Technology Based on Image Processing. *Appl. Sci.* **2024**, *14*, 3327. <https://doi.org/10.3390/app14083327>.
31. Li, L.; Ma, S.; Li, Y.; Huo, P.; Peng, C.; Li, W. Tree-PointNet: A Novel Neural Network for Dormant Apple Trees Using Semi-Circle-Based Point Cloud for Robotic Pruning. *J. ASABE* **2024**, *67*, 1405–1413. <https://doi.org/10.13031/ja.15871>.
32. Fu, Y.; Xia, Y.; Zhang, H.; Fu, M.; Wang, Y.; Fu, W.; Shen, C. Skeleton extraction and pruning point identification of jujube tree for dormant pruning using space colonization algorithm. *Front. Plant Sci.* **2023**, *13*, 1103794. <https://doi.org/10.3389/fpls.2022.1103794>.
33. Li, Y.; Zhang, Z.; Wang, X.; Fu, W.; Li, J. Automatic reconstruction and modeling of dormant jujube trees using three-view image constraints for intelligent pruning applications. *Comput. Electron. Agric.* **2023**, *212*, 108149. <https://doi.org/10.1016/j.compag.2023.108149>.
34. Giang, T.T.H.; Ryoo, Y.-J. Pruning Points Detection of Sweet Pepper Plants Using 3D Point Clouds and Semantic Segmentation Neural Network. *Sensors* **2023**, *23*, 4040. <https://doi.org/10.3390/s23084040>.
35. Gebrayel, F.; Mujica, M.; Danès, P. Visual Servoing for Vine Pruning based on Point Cloud Alignment. In Proceedings of the Informatics in Control, Automation and Robotics—21st ICINCO, Porto, Portugal, 18–20 November 2024.
36. You, A.; Grimm, C.; Silwal, A.; Davidson, J.R. Semantics-guided Skeletonization of Sweet Cherry Trees for Robotic Pruning. *arXiv* **2021**, arXiv:2103.02833.
37. Silwal, A.; Yandun, F.; Nellithimaru, A.; Bates, T.; Kantor, G. Bumblebee: A Path Towards Fully Autonomous Robotic. Vine Pruning. *arXiv* **2021**, arXiv:2112.00291.
38. You, A.; Parayil, N.; Krishna, J.G.; Bhattarai, U.; Sapkota, R.; Ahmed, D.; Whiting, M.; Karkee, M.; Grimm, C.M.; Davidson, J.R. Semiautonomous Precision Pruning of Upright Fruiting Offshoot Orchard Systems: An Integrated Approach. *IEEE Robot. Autom. Mag.* **2023**, *30*, 10–19. <https://doi.org/10.1109/MRA.2023.3309098>.
39. Ma, X.; Qin, C.; You, H.; Ran, H.; Fu, Y. Rethinking Network Design and Local Geometry in Point Cloud: A Simple Residual MLP Framework. *arXiv* **2022**, arXiv:2202.07123.
40. Zhao, H.; Jiang, L.; Jia, J.; Torr, P.H.S.; Koltun, V. Point Transformer. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 16239–16248.
41. Narbayeva, A.; Akça, Y. Comparison of Different Pruning Methods for Training Young Fernor Walnut Trees. *J. Agric. Fac. Gaziosmanpasa Univ.* **2024**, *41*, 25–32. <https://doi.org/10.55507/gopzfd.1455043>.
42. Lu, H.; Li, B.; Yang, G.; Fan, G.; Wang, H.; Pang, Y.; Wang, Z.; Lian, Y.; Xu, H.; Huang, H. Towards a point cloud understanding framework for forest scene semantic segmentation across forest types and sensor platforms. *Remote Sens. Environ.* **2025**, *318*, 114591. <https://doi.org/10.1016/j.rse.2024.114591>.
43. Chen, Q.; Luo, H.; Cheng, Y.; Xie, M.; Nan, D. An Individual Tree Detection and Segmentation Method from TLS and MLS Point Clouds Based on Improved Seed Points. *Forests* **2024**, *15*, 1083. <https://doi.org/10.3390/f15071083>.

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.