Research on Branch Recognition and Pruning Method for Dormant Apple Trees Based on Neural Radiance Fields and PointNeXt

Xinbing Guo
Jianhang Wang
Jinliang Gong
Yanfei Zhang
1392076@sina.com

Research Article

Keywords: Fruit trees, 3D reconstruction, Branch recognition, Branch pruning, Semantic segmentation

Posted Date: May 11th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9576293/v1

Additional Declarations: No competing interests reported.

Research on Branch Recognition and Pruning Method for Dormant Apple Trees Based on Neural Radiance Fields and PointNeXt

Abstract

To address the problem of fine branch identification and pruning decision for dormant apple trees, this study proposes a 3D point cloud branch recognition method integrating Neural Radiance Fields (NeRF) and the PointNeXt network. This method employs the neural radiance field theory to construct a point cloud model of apple trees, achieving fine detail representation and providing a high-precision, high-standard dataset for subsequent branch pruning experiments.

First, a panoramic video is captured by circling the fruit tree, and a multi-view image sequence is obtained through frame sampling. Subsequently, the Structure from Motion (SfM) algorithm is employed for sparse reconstruction to recover the pose information of the images. On this basis, a neural radiance field model is trained. Hierarchical sampling is performed using ray casting, and the sampled points, combined with positional encoding, are fed into a multi-layer perceptron (MLP). The radiance field is then generated via volume rendering, from which a high-fidelity 3D point cloud model of the fruit tree is derived. Finally, the point cloud is processed using the PointNeXt semantic segmentation network to achieve the identification and segmentation of branches to be pruned and branches to be retained.

To verify the effectiveness of the method, this study reconstructed point cloud models of dormant apple trees and selected 10 of them for experimental analysis. The algorithm achieved an average overall recognition accuracy of 75.15% and an average false negative rate (FNR) of 24.85%. The experimental results demonstrate that the proposed method constructs a 3D point cloud model with multi-scale, multi-modal, and high-precision phenotypic information at a relatively low cost. It not only overcomes the limitations of traditional 3D reconstruction methods, such as insufficient point cloud accuracy and difficulty in accurately identifying thin branches, but also effectively mitigates the high misrecognition rate observed in conventional branch recognition approaches. This provides technical support for unmanned agricultural machinery pruning in orchards and holds significant implications for achieving precision agriculture and sustainable development.

Key words： Fruit trees, 3D reconstruction, Branch recognition, Branch pruning, Semantic segmentation

0. Introduction

Pruning is a fundamental horticultural practice in modern apple orchard management, directly influencing tree architecture, light interception, fruit yield, and fruit quality[1]. In intensive orchard systems, particularly during the dormant season, pruning decisions require accurate assessment of branch structure, spatial distribution, and hierarchical relationships within the canopy[2-3]. However, pruning operations still rely heavily on skilled labor, leading to high labor costs, limited scalability, and strong dependence on individual experience. These challenges have motivated increasing research interest in automated and robotic pruning systems[4-6]. Reliable perception of three-dimensional (3D) branch structure is a prerequisite for automated pruning[7]. Over the past decade, various sensing and reconstruction techniques have been explored to capture 3D information of fruit trees. Terrestrial laser scanning and LiDAR-based methods can provide precise geometric measurements[8-9], although they typically lack color information; moreover, they are susceptible to occlusion in dense canopies[10]. Multi-view stereo (MVS) methods based on structure-from-motion (SfM) have been widely adopted due to their low hardware cost[11]; however, their ability to preserve thin branches and complex topological structures remains limited, especially under challenging illumination and texture conditions. Depth sensors such as Kinect have also been applied, but their effective range and reconstruction fidelity are often insufficient for fine-scale branch analysis.

More recently, Neural Radiance Fields (NeRF) have emerged as a powerful representation for novel-view synthesis and high-fidelity 3D reconstruction[12-13]. By modeling scene geometry and appearance through continuous neural functions, NeRF has demonstrated superior capability in capturing high-frequency details and complex structures[14-15]. Several studies have begun to explore NeRF-based reconstruction in agricultural scenarios, including crop and tree modeling. Nevertheless, existing works primarily focus on visual realism or phenotypic measurement[16], while the potential of NeRF for task-oriented perception—particularly pruning-oriented branch analysis—has not been fully investigated. Beyond 3D reconstruction, accurate identification of pruning-required branches is essential for translating perception results into actionable pruning decisions. Point cloud-based learning methods have shown promise in plant structure analysis, with networks such as PointNet and PointNet++ enabling direct processing of unordered 3D data[17-19]. However, these early architectures often struggle with fine-grained branch segmentation and structural continuity in complex tree geometries. Recently, PointNeXt has been proposed as an improved point cloud learning framework, offering enhanced local feature aggregation and improved efficiency[20-21], which makes it particularly suitable for distinguishing thin and overlapping branch structures in orchard environments.

Despite these advances, two key challenges remain for automated pruning applications. First, conventional 3D reconstruction methods often fail to generate point clouds with sufficient structural fidelity to support reliable branch-level pruning analysis[22]. Second, most existing studies evaluate segmentation performance using generic point-wise metrics such as overall accuracy or mean intersection over union, which do not adequately reflect pruning decision reliability[23], where missed or falsely identified branches can lead to significant operational errors. To address these challenges, this study proposes a pruning-oriented framework that integrates NeRF-based high-fidelity 3D reconstruction with PointNeXt-based point cloud segmentation for dormant apple trees. The reconstructed NeRF-derived point clouds preserve fine branch geometry, color consistency, and topological continuity, providing a robust foundation for downstream pruning analysis. Furthermore, in addition to conventional segmentation metrics, this study introduces branch-level pruning evaluation indicators to assess false positives, false negatives, and omission rates, thereby aligning model evaluation more closely with practical pruning requirements. The proposed method is validated on dormant apple trees under real orchard conditions. Experimental results demonstrate that the proposed framework outperforms traditional MVS-based reconstruction and classical point cloud segmentation networks in both geometric fidelity and pruning applicability, highlighting its potential for robotic pruning and intelligent orchard management.

1. Materials and methods

The study was conducted at Hongji Ecological Farm in Zibo City, Shandong Province (36°56′N, 117°58′E). Ten five-year-old “Fuji” apple trees with a spindle-shaped canopy and good growing conditions were selected. The trees were in the dormant stage, during which the main task is branch pruning. The experiment was conducted in December 2025, with data acquisition from 15:30 to 18:00. During this period, the weather was predominantly cloudy with light wind, ambient light intensity was moderate, and there was no direct sunlight, providing uniform and soft illumination.

The specific procedure of the proposed method is as follows: First, a self-developed information acquisition platform is used to capture multi-view videos by circling the fruit tree. Secondly, Adobe Premiere Pro (Pr) is employed to enhance and subsample the video frames. Subsequently, the Structure from Motion (SfM) method is applied to perform sparse reconstruction on the images, producing a multi-view image sequence with associated camera poses. Finally, this sequence is used to train a Neural Radiance Field (NeRF) model, from which a point cloud is derived to accomplish dense 3D reconstruction of the fruit tree. The computer hardware used in the experiment was configured with an Intel i9-11980HK processor, an NVIDIA GeForce RTX 3090 graphics card, and 64 GB of RAM.

1.1 Information Acquisition

The image acquisition site is shown in Figure 1a, where the orchard fruit trees are in the dormant period. The image acquisition equipment is shown in Figure 1b. The equipment mainly consists of four components: a three-axis gimbal camera, a sky-end signal receiver, a Wi-Fi image transmission module, and an antenna. The three-axis self-stabilizing gimbal maintains the stability of the camera's horizontal and pitch angles. Mounted on the gimbal is a zoom camera equipped with Gyroflow digital stabilization technology, featuring a field of view of 170°, an equivalent focal length of 24–48 mm, an image resolution of 3840 × 2160 pixels, a shutter speed ranging from 1/8 to 1/8000 s, an effective pixel count of 8 million, and support for 6× electronic zoom. The acquisition platform can be mounted on a telescopic pole to increase the shooting height, with the pole adjustable from 1.5 m to 7 m.

To achieve efficient multi-angle image acquisition of fruit trees, this study proposes a spiral trajectory shooting method based on a handheld mobile platform. Using a handheld telescopic pole with a gimbal-mounted camera, the collector traversed a spiral path around the fruit tree, recording three loops of video at the upper, middle, and lower canopy levels, while ensuring that the start and end points of the spiral remained vertically aligned. During shooting, the pan-tilt direction is controlled in real time to keep the camera frame center constantly aimed at a fixed point on the tree trunk, thereby synchronously acquiring multi-angle continuous video images including upward, horizontal, and downward views. Using video recording instead of single-frame photography avoids frequent shutter operation and improves acquisition efficiency. Subsequent frame extraction with Premiere Pro software enables the extraction of a large number of high-quality still frames from a short video. In the above acquisition procedure, the total video duration is approximately 150 s, which significantly reduces field operation time while ensuring image coverage.

a. Image acquisition site b. Image acquisition equipment

Fig.1 Image acquisition scene and equipment

1.2 Fruit Tree 3D Reconstruction Method Based on Neural Radiance Field

This study employs Neural Radiance Fields (NeRF) for three-dimensional reconstruction. Neural Radiance Fields (NeRF) are fundamentally a method for representing 3D scenes based on an implicit function. Their core component is a continuous function approximated by a multilayer perceptron (MLP), which encodes the color information and volume density of any spatial point within the scene. This function can be expressed as:

$$\boldsymbol{F}(x, y, z, \theta, \varphi) \to (R, G, B, w) \tag{1}$$

Where *x, y, z, θ, φ* are the input quantities to the MLP, constituting a 5-dimensional signal formed by the spatial point coordinates (x, y, z) and the camera ray direction (θ, φ); while (R, G, B, w) represents the output vector of the MLP.

The initially generated radiance field exhibits a discrete, cloud-like appearance and cannot accurately represent the morphology of the fruit trees. Therefore, volume rendering is applied to the volume density and color of the sampled points output by the MLP. By using the volume density as weights, a weighted summation of the colors of all sampled points along the camera ray is performed, and the color of a pixel on the imaging plane traversed by the ray is computed via integration. The formula is as follows:

$$\begin{cases} \boldsymbol{C}_c(\boldsymbol{r}) = \pi \int_{t_n}^{t_f} T(t) w(r(t)) \boldsymbol{C}(\boldsymbol{r}(\boldsymbol{t})) dt \\ T(t) = exp(-\int_{t_n}^{t} w(r(s)) ds) \end{cases} \tag{2}$$

Where $\boldsymbol{C_c(r)}$ is the pixel color, t_n denotes the near bound of the viewing frustum, and t_f denotes the far bound. $T(t)$ represents the accumulated transmittance coefficient, $w(r(r(t))$ is the volume density of the sampled point at position $r(t)$, $w(r(r(s))$ is the volume density of the sampled point at position $r(s)$, and $\boldsymbol{C(r(t))}$ is the color of the sampled point at position $r(t)$.

To achieve high-quality rendering results with fewer sampled points while effectively reducing computational load, hierarchical sampling and rendering with two levels of granularity—coarse and fine—are performed along each ray. Specifically, using the volume density distribution of coarse-grained sampled points as a reference, more fine-grained sampled points are concentrated in regions with higher volume density along the ray. First, coarse-grained sampling is applied to uniformly sample N_c points along the camera ray, which are fed into the MLP for volume density rendering to obtain a preliminary volume density distribution, along with the coarse-grained rendered pixel color $\boldsymbol{\hat{C}_c(r)}$. Then, N_f fine-grained sampled points are drawn from regions with higher volume density and input again into the MLP for volume rendering, yielding the fine-grained rendered volume density and pixel color $\hat{\boldsymbol{C}}_f(\boldsymbol{r})$. The formulas are as follows:

$$\begin{cases} \hat{\boldsymbol{C}}_c(\boldsymbol{r}) = \sum_{g=1}^{N_c} w_g \boldsymbol{C}_g \\ \hat{\boldsymbol{C}}_f(\boldsymbol{r}) = \sum_{l=1}^{N_c+N_f} w_l \boldsymbol{C}_l \end{cases} \tag{3}$$

Where w_g and w_l denote the cumulative volume densities of the coarse-grained and fine-grained sampled points, respectively, $\boldsymbol{C}_g$ and $\boldsymbol{C}_l$ represent the colors of the coarse-grained and fine-grained sampled points, respectively, and N_c and N_f are the numbers of coarse-grained and fine-grained sampled points.

After computing the coarse-grained rendered pixel color $\hat{\boldsymbol{C}}_c(\boldsymbol{r})$ and the fine-grained rendered pixel color $\hat{\boldsymbol{C}}_f(\boldsymbol{r})$, a loss function is employed to update the weights of the MLP through backpropagation, progressively driving the scene to converge toward the final 3D representation of the fruit tree. Specifically, $\hat{\boldsymbol{C}}_c(\boldsymbol{r})$ and $\hat{\boldsymbol{C}}_f(\boldsymbol{r})$ are compared with the actual pixel color on the imaging plane traversed by the camera ray, and the training loss is calculated. By iterating over all pixels in the same view, the loss function L for that view is obtained as follows:

$$L = \sum_{r \in \boldsymbol{O}} [\left\| \hat{\boldsymbol{C}}_c(\boldsymbol{r}) - \boldsymbol{C}(\boldsymbol{r}) \right\|_2^2 + \left\| \hat{\boldsymbol{C}}_f(\boldsymbol{r}) - \boldsymbol{C}(\boldsymbol{r}) \right\|_2^2] \tag{4}$$

Where $\hat{\boldsymbol{C}}_c(\boldsymbol{r})$ and $\hat{\boldsymbol{C}}_f(\boldsymbol{r})$ represent the pixel colors obtained from coarse-grained and fine-grained rendering calculations, respectively, $\boldsymbol{C}(\boldsymbol{r})$ denotes the actual color of the pixel along the ray direction; and ***O*** refers to the set of all pixels in a single view.

The network weights of the MLP are updated via back-propagation of the loss function, and the scene is iteratively adjusted to gradually approximate the true morphology of the fruit tree until convergence conditions are met. In this way, the training of the neural radiance field for the fruit tree is completed, and the NeRF scene representation of the fruit tree is obtained.

To convert the converged neural radiance field into a 3D geometric model for quantitative analysis, the density threshold method is adopted for point cloud extraction in this study. The principle of this method is based on the physical interpretation of the NeRF output: the predicted volume density σ represents the probability that a given point in space lies on an object surface. By setting an appropriate global threshold τ, the reconstructed 3D space is sampled uniformly or adaptively. All sampled points satisfying $\sigma\left(x,y,z\right)>\tau$ are retained, and their spatial coordinates along with the associated color information are exported. This process yields an initial point cloud model representing the geometric surface of the fruit tree. This strategy directly extracts explicit geometry from the implicit field, which is efficient and requires no additional training. The reconstruction is optimized for branch continuity rather than surface completeness. This method is suitable for geometric reconstruction of slender branches of dormant fruit trees, directly serving pruning decision support. The density thresholding method, by directly thresholding the continuous density field, can effectively capture the thin branch structures of dormant fruit trees, providing a reliable 3D data basis for branch identification, tree architecture analysis, and pruning decision-making.

1.3 Point Cloud Preprocessing Methods

To eliminate ground interference, ground points are first removed from the initial point cloud model. Based on the characteristic that ground points have low Z-coordinate values, a threshold interval along the Z-axis is defined. Points within this interval are retained as the tree point cloud, while points outside the interval are removed as ground and weeds. This method enables rapid filtering of ground regions through coordinate-based screening.

$$Z_{\mathrm{min}} \le Z_{\mathrm{i}} \le Z_{\mathrm{max}} \tag{5}$$

Where Z_i is the Z-axis coordinate value of any point in the target point cloud, $Zmin$ is the minimum Z-axis coordinate value of the retained point cloud, and also the maximum Z-axis coordinate value of the ground point cloud, $Zmax$ is the maximum Z-axis coordinate value of the retained point cloud.

To address the presence of outlier noise and unconverged white discrete points in the point cloud, a combined method of RGB color filtering and statistical filtering is employed for denoising. To reduce the computational load of statistical filtering, a color threshold based on the RGB value (255, 255, 255) is first applied to filter out approximately 50% of the white discrete points, resulting in a preliminarily cleaned point cloud model. Given the advantages of statistical filtering—simple parameter setting, independence from specific prior models, strong adaptability, and high computational efficiency—this study selects this method for outlier noise filtering of the fruit tree point cloud. The process is as follows:

Let the point cloud set be:

$$P=\left\{p_i(x_i,y_i,z_i)\,|\,i=1,2,\ldots,N\right\} \tag{6}$$

Where P is the point cloud dataset, P_i is an individual point within the dataset, and N is the number of points in the point cloud. First, the point cloud data is organized into a k-d tree data structure. Then, a statistical analysis is performed on all points in the dataset by calculating the average distance from each point to its k nearest neighbors, which can be expressed as follows:

$$\left\{\begin{array}{l} d_i=\sqrt{(x_i-x_i')^2+(y_i-y_i')+(z_i-z_i')^2} \\ \overline{d_i}=\dfrac{1}{k}\sum_{i=1}^{k}d_i \end{array}\right\} \tag{7}$$

Where d_i is the distance between point P_i and one of its neighboring points, $\overline{d_i}$ is the average distance between point P_i and its neighboring points, and (x_i' , y_i' , z_i') denotes the coordinates of a neighboring point of P_i.

Subsequently, the global average point distance and standard deviation across all points in the point cloud dataset are calculated, and a threshold L is defined, as expressed by the following formulas:

$$\mu = \frac{1}{N}\sum_{i=1}^{N} d_i \tag{8}$$

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(d_i - \mu)^2} \tag{9}$$

$$L = \mu + \lambda \cdot \sigma \tag{10}$$

Finally, the threshold L is compared with $\overline{d_i}$. If $\overline{d_i} > L$, the point P_i is identified as an outlier noise point and removed; otherwise, it is retained. By iterating through all points in the point cloud dataset P according to the above procedure, point cloud filtering is completed. After RGB filtering and statistical filtering, the point cloud model becomes cleaner. This process effectively removes millions of noise points, reduces the data scale and the computational burden for subsequent segmentation training, thereby improving both training speed and accuracy.

1.4 Branch Recognition Method Based on PointNeXt

The semantic segmentation module of the PointNeXt network is used to segment the fruit tree point cloud. PointNeXt, an enhancement of PointNet++ for point cloud semantic segmentation, retains the pyramidal architecture of hierarchical feature learning and multi-stage downsampling while incorporating multiple optimizations in its core units. Among them, the lightweight MLP encoder adopts a decoupled design to reduce computational redundancy: it first processes the features of each point independently via depthwise convolution with a kernel size of 1×1×K, where K is the number of input channels, and then performs cross-channel fusion through pointwise convolution, thereby enhancing the accuracy of local feature perception and extraction while significantly improving training efficiency. The formula is as follows:

$$F_{depth}\left(\varpi_i\right) = \sum_{k=1}^{K} \eta_k^{(d)} \cdot \varpi_i^{(k)} \tag{11}$$

$$F_{point} = \eta_p \cdot F_{depth} + b \tag{12}$$

Where ϖ_i represents the feature vector of the i-th point in the input point cloud, $\eta_k^{(d)}$ denotes the weight parameter of the k-th channel in the depthwise convolution kernel, and $\varpi_i^{(k)}$ indicates the feature value of the k-th channel for the i-th point.

The design adopts a decoupled two-step structure: the first step performs depthwise convolution only on spatial information, and the second step achieves cross-channel fusion via pointwise convolution. This approach reduces the parameter count from input channels × output channels to only output channels. The independent convolution per channel preserves channel-wise independence and enhances boundary sensitivity, thereby improving local perception capability while reducing computational cost.

The Inverted Bottleneck structure is designed to address the issues of information loss and impaired nonlinearity caused by premature channel compression in traditional bottleneck structures. This structure processes features in three stages—expansion, transformation, and compression—with the channel count following an hourglassshaped variation: first, the input channels are expanded to a higher dimension to enhance feature decoupling capability; then, depthwise convolution is applied to maintain the channel count while extracting local geometric features; finally, the channels are compressed and a linear transformation is used to avoid nonlinear suppression, thereby achieving feature distillation. Compared with conventional architectures, this design reduces computational cost while enhancing operational efficiency. The Dynamic Spatial Local Encoder (DSLE) is designed to overcome the insufficient adaptability of traditional fixed-weight convolutions to varying geometric structures. It dynamically generates convolution kernel weights based on local geometric features, thereby improving sensitivity to spatial structure perception and reducing redundant computations.

The three aforementioned innovative modules of PointNeXt are processed in a cascaded manner, forming a complete processing chain: first, a lightweight MLP performs initial feature embedding; then, an Inverted Bottleneck structure enhances nonlinear representation; finally, dynamic convolution captures fine details. This approach successfully improves the computational efficiency and target recognition accuracy of the network. A schematic diagram of the workflow is shown in Figure 2.

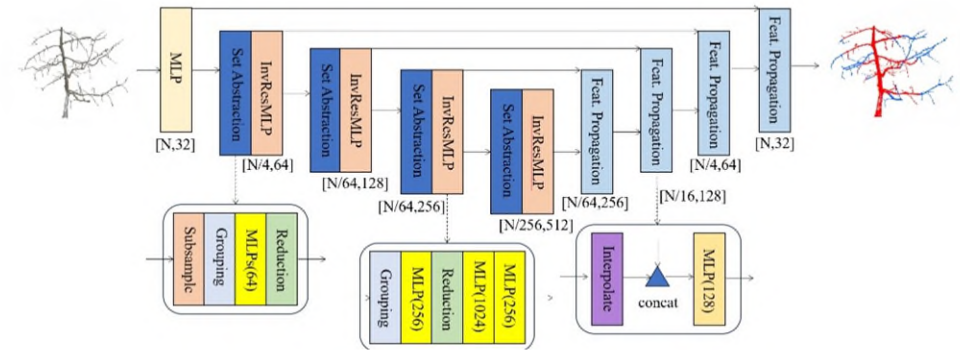


Fig.2 PointNeXt Network Architecture

2. Results and analysis

2.1 Experimental Evaluation Metrics and Pruning Criteria

In deep learning research, evaluation metrics serve as a critical basis for measuring model performance and effectiveness, as well as for selecting the optimal model. This paper selects commonly used performance evaluation metrics in the field of point cloud semantic segmentation as the benchmark for assessing the semantic segmentation performance of the network model. These include the mean Intersection over Union (mIoU), Overall Accuracy (OA), and mean Accuracy (mAcc).

$$OA = \frac{\sum_{i=1}^{n} TP_i}{\sum_{i=1}^{n} W_i} \times 100\% \tag{13}$$

$$mACC = \frac{1}{n}\sum_{i=1}^{n}\frac{TP_i}{W_i} \times 100\% \tag{14}$$

$$mIoU = \frac{1}{n}\sum_{i=1}^{n}\frac{|P_i \cap L_i|}{|P_i \cup L_i|} \tag{15}$$

Where m is the number of classes, TP_i denotes the number of correctly predicted points in class i, and W_i represents the total number of points in class i, P_i refers to the set of predicted points for class i, and L_i indicates the set of ground truth points for class i.

In the task of fruit tree branch recognition and pruning, the reliability of an algorithm depends not only on its ability to correctly identify targets, but also on a systematic evaluation of its false positives and false negatives. Therefore, this study introduces the following key metrics: False Positive Branch (FPB) refers to the number of non-pruning branches that the algorithm mistakenly identifies as pruning branches, reflecting its misclassification capability. False Negative Branch (FNB) refers to the number of actual pruning branches that the algorithm fails to correctly identify and locate, indicating its omission. False Negative Rate (FNR) is the percentage of false negative branches relative to the total number of pruning branches, measuring the extent to which the algorithm misses pruning branches.

The pruning criteria for apple trees are to optimize canopy structure and function, with a focus on improving ventilation and light transmission, thereby enhancing photosynthetic efficiency and fruit quality. During pruning, excessively dense branches should be properly removed to enhance light uniformity and air circulation within the canopy, thereby promoting sufficient leaf photosynthesis, providing adequate nutrition for fruit development, while reducing canopy humidity and suppressing disease occurrence. To achieve the above objectives, pruning must be guided by clear quantitative indicators to ensure a rational tree structure, adequate ventilation and light penetration, and a balance between vegetative and reproductive growth. The specific pruning criteria are as follows: (1) Spacing control: When the distance between the starting points of adjacent branches on the same side is less than 30 cm, or the distance between the endpoints is less than 20 cm, remove the branch that is positioned higher or exhibits excessive growth vigor, so as to optimize branch distribution and avoid competition and overlap. (2) Length control: For branches exceeding 80 cm in length, the excess part shall be pruned to prevent excessive outward expansion of the canopy, promote the development of medium and short branches, and improve the stability of fruiting sites. (3) Branch to trunk ratio control: when the diameter at the base of a branch exceeds half of the trunk diameter, the balance between the branch and the trunk is disrupted. In this case, severe retraction should be applied, leaving only a short stub of 20 to 30 cm to stimulate new bud germination at the base, thereby maintaining the dominance of the central trunk. The above three criteria need to be comprehensively evaluated and flexibly applied to achieve three-dimensional fruiting and long-term tree health.

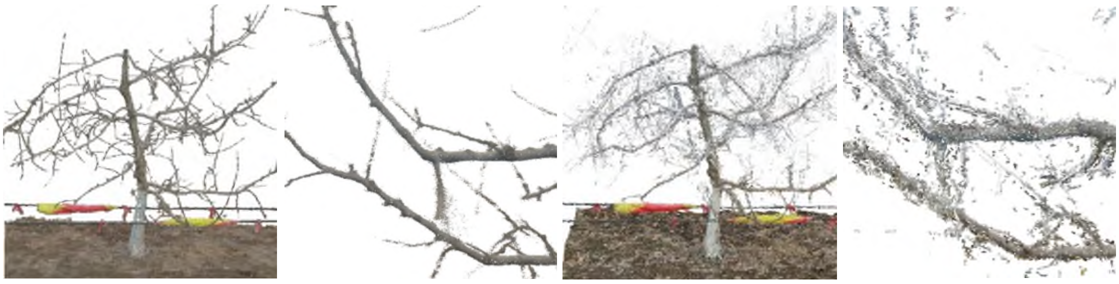
2.2 3D Reconstruction Results

Using the above method, the NeRF scene, point cloud, and depth scene model of dormant apple trees were successfully reconstructed. To evaluate the representation quality, one real image was randomly selected from the image sequence and compared with the NeRF scene, point cloud model, and depth map, as shown in Figure 3. Both the NeRF scene and the point cloud model clearly depict the branch architecture of apple trees, with distinct canopy contours and well-defined branch layering in the depth map, thereby precisely representing the spatial distribution of the canopy. This indicates that the method exhibits strong representational capacity and adaptability for fruit trees.

a.realistic picture b.NeRF scene c.NeRF point cloud d.NeRF depth scene

Fig.3 Comparison between fruit tree image and NeRF scene、point cloud and depth scene

The traditional MVS method was used to perform three-dimensional reconstruction of the same fruit tree, yielding an MVS point cloud model as the control group. A comparison of the reconstruction results between NeRF and MVS is shown in Figure 4. At the macroscopic scale, the fruit tree point cloud model reconstructed by NeRF exhibits few discrete noise points and almost no white noise, whereas the point cloud model reconstructed by MVS contains a substantial number of discrete noise points, with white noise enveloping the entire tree and hindering the acquisition of its morphological data. At the local scale, the point cloud model reconstructed by NeRF exhibits distinct branch shapes, whereas that reconstructed by MVS suffers from branch adhesion, failing to accurately capture the branch structure of fruit trees. The branch edges are surrounded by white noise, resulting in blurred and indistinguishable boundaries, and the point cloud model is severely distorted.



a.NeRF Macroscopic point cloud　　b.NeRF Local point cloud　　c.MVS Macroscopic point cloud　　d.MVS Local point cloud

Fig.4　Comparison of fruit tree point clouds generated by NeRF and MVS methods

To compare the accuracy of the NeRF and MVS point cloud models, the NeRF scene was converted and exported in point cloud format. The dense reconstruction time and the number of points required by the two reconstruction methods are presented in Table 1. The number of points in the NeRF point cloud during the dormant period increased by an order of magnitude compared to that of the MVS point cloud, while the dense reconstruction time of NeRF was reduced by an average of 1.27% relative to MVS, indicating improved efficiency.

Table 1　Point cloud quantity and reconstruction time

Method	Number of point clouds	Reconstruction time/(s)
NeRF	21463593	3204
MVS	2766091	2933

To evaluate the data consistency of point cloud models, this study selected tree height, diameter at breast height, and ground diameter as tree structure indicators, and selected HSB color as an organ phenotypic indicator. In the measurements, the diameter at breast height and the basal diameter were measured at their widest parts using a vernier caliper, while the tree height was measured using the projection method with a tape measure. HSB color data were obtained from three selected locations on the real-scene images. Each indicator was measured three times, and the arithmetic mean of the three measurements was taken as the measured value. In the model measurement stage, markers wrapped around a 68.33 cm long drip irrigation pipe in the orchard scene are used to recover the scale of the initial point cloud lacking true scale, thereby scaling the model to its actual dimensions. Then, measurements were performed at the same positions within the scaled point cloud, and the average of three repeated measurements was taken as the experimental value.

The results show that tree structure data consistency is presented in Table 2. NeRF achieves an average error of 3.38% across the three metrics, whereas MVS yields an average error of 14.73%, representing an 11.35% reduction compared to MVS. Moreover, NeRF outperforms MVS in all three metrics, with values closer to the ground truth. Notably, for thick structures such as the ground diameter, the MVS error reaches 24.69%, while NeRF achieves only 4.13%, indicating that NeRF is more robust in reconstructing large structures. The color consistency is shown in Table 3. The NeRF model achieves color data that are closer to the real-scene images with smaller deviations, demonstrating superior color consistency over MVS, especially in terms of hue and brightness, where it exhibits more stable performance. In contrast, MVS exhibits obvious systematic bias, characterized by large hue deviations and significantly underestimated brightness, failing to accurately reconstruct the true color information. Overall, the NeRF model achieves significant improvements over the MVS model in both tree architecture and color data consistency, making it more suitable for subsequent branch pruning requirements of fruit trees.

Table 2 Tree structured data consistency statistics

	Measured value	NeRF Test value	MVS Test value	NeRF Deviation/%	MVS Deviation/%
Breast diameter /mm	78.39	76.69	86.88	1.70	8.49
Ground diameter/mm	99.75	103.88	124.44	4.13	24.69
Tree height/cm	208.71	200.38	190.73	8.33	17.98

Table 3 Color Consistency statistics

		Branches1	Branches2	Branches3
Measured value	H	26°	60°	35°
	S	4%	5%	6%
	B	63%	62%	65%
NeRF Test value	H	43°	51°	38°
	S	9%	10%	8%
	B	61%	52%	63%
MVS Test value	H	180°	27°	100°
	S	9%	9%	10%
	B	51%	46%	55%
NeRF Deviation	H	17°	-9°	3°
	S	5%	5%	2%
	B	-2%	-10%	-2%
MVS Deviation	H	154°	-33°	65°
	S	5%	4%	4%
	B	-12%	-16%	-10%

2.3 Point Cloud Preprocessing Results

To improve the accuracy of subsequent point cloud segmentation experiments, reduce computational load, and enhance data accuracy and reliability, this study preprocessed the reconstructed fruit tree point cloud by removing ground points and applying noise filtering operations. This process effectively preserves the main structure of the fruit tree while significantly reducing irrelevant points and noise points, ultimately decreasing the point cloud count by approximately 44.4%, see Table 4.

Table 4 Number of point clouds on fruit trees

Point cloud type	Unprocessed	Remove ground point clouds	Remove ground point clouds and noise
Number of point clouds	21463593	17064306	11929177

The reconstructed point cloud model of the fruit tree and the point cloud after ground removal are shown in Figure 5a and Figure 5b, respectively. However, numerous outliers and white noise points still exist, which affect subsequent dataset construction and the accuracy of semantic segmentation experiments. Therefore, this study sequentially applied statistical outlier filtering to remove discrete outliers, followed by RGB filtering to eliminate white noise points. After processing, as shown in Figure 5c, outliers and white noise points are effectively removed. The main structure of the fruit tree is fully preserved, and the branch contours are clearly distinguishable, providing high-quality point cloud data for subsequent analysis.

a.Reconstructed point cloud model of fruit tree　　b.Remove ground point cloud model of fruit tree　　c.Tree point cloud model after removing noise

Fig 5　pretreatment

Datasets are a critical component of deep learning training. In this study, PointNeXt is selected as the deep learning framework for point cloud segmentation. The PointNeXt segmentation model requires appropriate three-dimensional point cloud datasets of fruit trees. The dataset should contain rich and accurate data information so that the fruit tree model can be well trained to ensure the performance of subsequent segmentation experiments. After establishing a 3D fruit tree model based on NeRF, preprocessing steps such as ground point removal and filtering of outliers and white noise points are first performed. Subsequently, fine-grained annotation of branch parts is carried out in CloudCompare according to fruit tree pruning principles: all branch point clouds are initially uniformly labeled with class label "1", and then the branches to be pruned are manually and precisely segmented based on quantitative pruning metrics and labeled with class label "0", thereby distinguishing retained branches from those to be pruned. The entire annotation process was performed using CloudCompare and custom annotation tools to construct a high-quality dataset for subsequent model training.

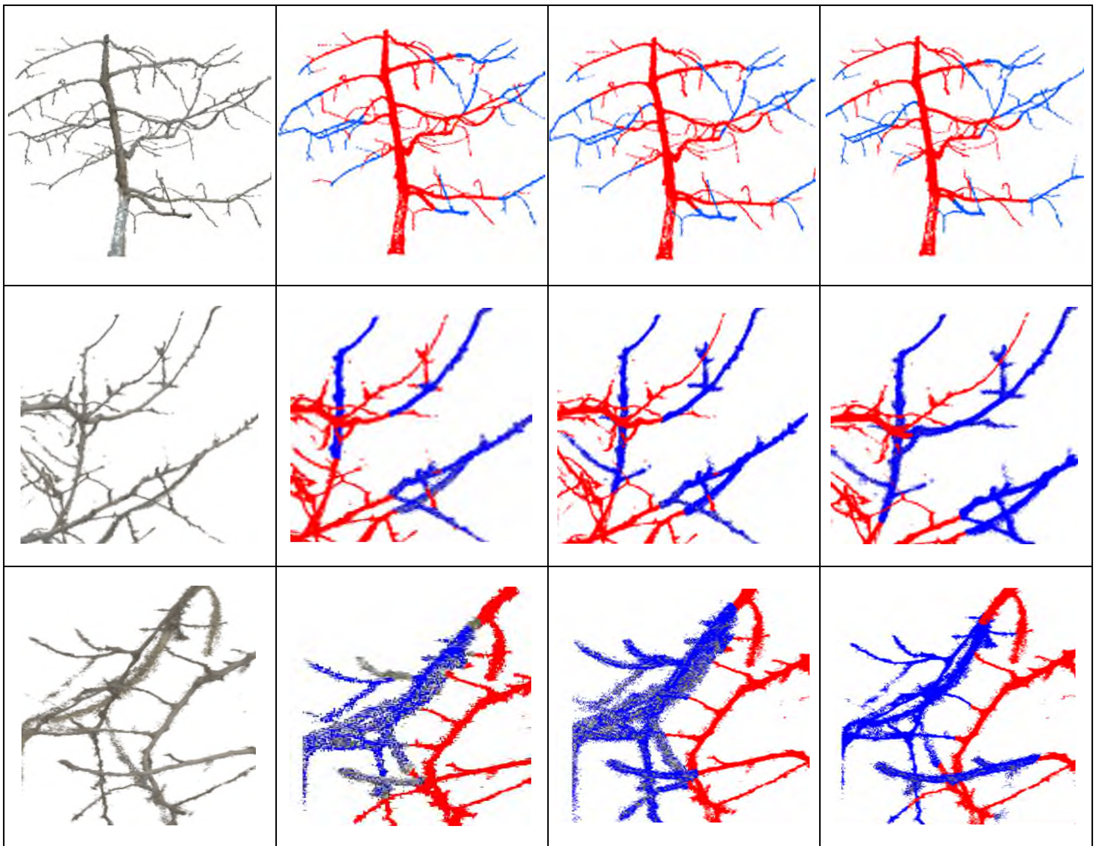
2.4 Branch Recognition Results

All processed apple trees were used as the training set for branch recognition and segmentation using the PointNeXt network. Ultimately, a network model capable of recognizing and segmenting branches on individual apple trees was obtained, and branch recognition and segmentation were subsequently performed on the reconstructed models of individual apple trees. The training of PointNeXt took a total of 49 hours, with GPU memory usage reaching 21.4 GB (out of 24 GB), and the training speed was maintained at 140 samples per second. To evaluate the experimental results, this study trained the PointNet and PointNet++ network models using the same dataset, and performed branch recognition and segmentation on the same fruit tree model as a control group for comparative discussion. The training results of PointNeXt for apple trees are presented in Table 5, with OA, mAcc, and mIoU reaching 0.9367, 0.9025, and 0.8917, respectively. Compared to PointNet and PointNet++, PointNeXt achieves improvements of 8.29% and 6.46% in OA, 4.02% and 3.12% in mAcc, and 4.81% and 2.33% in mIoU. Based on the training parameters, the segmentation results of the PointNeXt network are closer to the ground truth region than those of the other two networks. To further analyze the classification results of the two methods, the segmented point clouds are visualized, and the segmentation results are statistically analyzed.

Table 5 Model Segmentation accuracy

Model	OA	mACC	mIoU
PointNet	08538	0.8623	0.8436
PointNet++	0.8721	0.8713	0.8684
PointNeXt	0.9367	0.9025	0.8917

Using CloudCompare software, the apple tree point clouds identified by the PointNeXt, PointNet++, and PointNet models were visualized. Figures 6a, 6b, 6c, and 6d respectively show the full point cloud of the fruit tree, the recognition results obtained by PointNet, the recognition results by PointNet++, and the recognition results by PointNeXt. Based on a comparative analysis of the experimental results, PointNet, PointNet++, and PointNeXt all achieve effective recognition of apple tree branches. However, PointNet and PointNet++ exhibit omissions in the identification of pruning branches, as they struggle to accurately recognize thin branches and locally fractured structures. In particular, in the distal regions of thin branches, the completeness of point cloud feature extraction is poor, resulting in the failure to accurately detect some branches that require pruning. In contrast, the PointNeXt model demonstrates comprehensive advantages in branch recognition tasks. It exhibits outstanding structural integrity, significantly reduces branch breakage, and achieves more complete extraction of fine branches and terminal twigs. In terms of recognition accuracy, the model can more clearly distinguish branches from unstructured noise while maintaining a low false recognition rate. Furthermore, PointNeXt better preserves the topological connectivity of branches, and the reconstructed branch skeleton outperforms the other two models in terms of structural coherence and physical realism. The experimental results demonstrate that PointNeXt, leveraging its advanced network architecture, outperforms traditional models in completeness, accuracy, and structural fidelity for branch recognition in complex fruit tree point clouds, thereby providing a more reliable technical foundation for three-dimensional reconstruction, growth monitoring, and automated operations of fruit trees.



a.Point cloud of fruit trees	b.Branches detected by PointNet	c.Branches detected by PointNet++	d.Branches detected by PointNeXt

Fig 6   Branch Detection Results

The pruning branch data extracted by the PointNeXt algorithm were compared with the manually measured and labeled ground truth pruning branch data from field surveys. The recognition performance of pruning branches on 10 fruit trees was statistically analyzed, and the overall experimental results are presented in Table 6. A total of 186 branches were examined in the experiment, of which 136 required pruning. The algorithm identified 121 pruning branches, including 28 false positives, while missing 34 true pruning branches. A comparison between the total number of pruning branches detected by the algorithm on each tree and the actual number of pruning branches on that tree reveals that for 8 trees the detected count is lower than the actual count, for 1 tree it is equal to the actual count, and for 1 tree it is higher than the actual count. The average overall recognition accuracy of the algorithm is 75.15%, and the average false negative rate is 24.85%. However, the recognition accuracy of pruning branches in Group 4 was 66.7%, which was significantly lower than that under the other nine experimental conditions. The main reason is that this group of samples has a large total number of branches, severe mutual occlusion among branches, and a high proportion of thin branches, resulting in breakage and missing parts of some branches in the point cloud data, which significantly increases the number of false negative branches during the recognition process, thereby reducing the overall plant recognition accuracy.

Table 6 Results of Pruning Branch Identification

Number	Total Branch Count	Actual Pruning Branches	Detected Pruning Branches	FPB	FNB	FNR/%	Recognition Accuracy/%
1	17	12	11	2	3	25	75
2	20	14	13	3	4	28.6	71.4
3	15	14	12	3	2	15.4	84.6
4	23	12	10	4	4	33.3	66.7
5	21	17	14	3	5	29.4	70.6
6	14	13	13	2	2	15.4	84.6
7	17	9	10	2	2	22.2	77.8
8	21	15	14	3	4	26.7	73.3
9	18	13	11	2	3	23.1	76.9
10	20	17	14	4	5	29.4	70.6
Total	186	136	121	28	34	/	/
Mean	18.6	13.6	12.1	2.8	3.4	24.85	75.15

Using the total number of branches of the fruit tree as a measure of environmental complexity, the data were sorted and the relationship between environmental complexity and recognition accuracy was plotted, as shown in Figure 7. As the environmental complexity increases, that is, as the total number of branches of the fruit tree increases, the recognition rate generally exhibits a decreasing trend, reflecting a negative correlation between environmental complexity and recognition accuracy. As the number of fruit tree branches increases, the environment becomes more complex, and the interleaving and occlusion among branches intensify, making it difficult for the recognition algorithm to distinguish between pruning and non-pruning branches, thereby gradually reducing recognition accuracy. Specifically, when the total number of branches is 15 or fewer, the recognition accuracy remains above 84%; when the total number of branches is 20 or more, the accuracy generally decreases to 70% to 73%; and when the total number of branches reaches 23, the accuracy drops to 66.7%. This indicates that environmental complexity is an important factor affecting pruning branch recognition performance. The greater the number of branches and the more severe the occlusion, the higher the risk of missed detection and false detection by the algorithm.

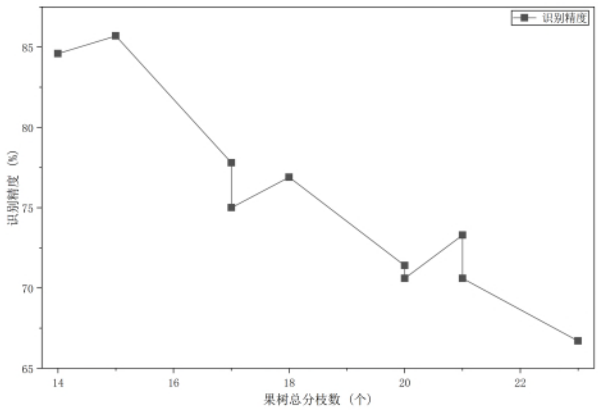


Figure 7 Relationship between the number of fruit tree branches and recognition accuracy

Comprehensive analysis indicates that the PointNeXt algorithm has achieved a robust capability for pruning branch recognition in complex fruit tree point clouds and exhibits stable performance across all samples. As the total number of branches increases, the recognition accuracy exhibits a decreasing trend. Despite certain missed detections and false positives, the algorithm can still effectively identify most of the branches requiring pruning, demonstrating high reliability particularly in regions with clear branch structures.

3.Discussion

The NeRF-based fruit tree reconstruction method proposed in this study exhibits strong robustness for dormant fruit trees, and its representation capability is significantly enhanced compared to the traditional MVS method. The NeRF model demonstrates excellent performance in tree shape data consistency, with a mean error of 3.38% for its reconstructed point cloud, which is 11.35% lower than that of the MVS model and closer to the measured values. In terms of color consistency, the color data generated by the NeRF model at various positions are closer to the real-world images, and its stability in hue and brightness metrics is significantly superior to that of the MVS model, thereby providing a high-quality data foundation for subsequent recognition tasks.

In terms of branch identification, the PointNeXt algorithm achieved scores of 0.9367, 0.9025, and 0.8917 for OA, mAcc, and mIoU, respectively. Compared with PointNet and PointNet++, PointNeXt improved OA by 8.29% and 6.46%, mAcc by 4.02% and 3.12%, and mIoU by 4.81% and 2.33%, respectively. However, traditional segmentation metrics primarily measure classification accuracy at the pixel level, while the core targets for pruning decisions and operations are branches with three-dimensional structures. Therefore, it is necessary to introduce branch-level evaluation metrics such as FPB, FNB, and FNR. Compared with traditional point cloud processing methods such as PointNet and PointNet++, this method demonstrates significant advantages in branch integrity, structural coherence, and misjudgment control. In this study, 10 apple trees were selected for pruning branch recognition. A total of 186 branches were detected in the experiment, of which 136 required pruning. The method identified 121 pruning branches, including 28 false positive branches, while missing 34 true pruning branches. The overall recognition accuracy of the algorithm reaches 75.15% with a false negative rate of 24.85%, demonstrating good capability in maintaining structural integrity and controlling misjudgment.

4. Conclusion

This study focuses on dormant apple trees and proposes a branch recognition method that integrates neural radiance field based three-dimensional reconstruction with PointNeXt point cloud segmentation technology. By acquiring multi-angle images and constructing a high-precision fruit tree point cloud model based on neural radiance fields, branch recognition and structural extraction are subsequently achieved using the PointNeXt network. Experimental results demonstrate that this method maintains high recognition accuracy and robustness under complex branch structures and background interference. The main discussions are as follows:

1. A pruning-oriented 3D reconstruction framework based on NeRF is proposed for dormant apple trees. Unlike conventional MVS methods, the proposed approach generates high-fidelity point cloud models that preserve fine branch structures,

geometric continuity, and color consistency under leaf-off conditions. This provides a reliable 3D representation tailored for pruning-related perception tasks rather than generic visualization.

2. A task-driven pipeline integrating NeRF-derived point clouds with PointNeXt for branch-level pruning identification is developed. By coupling NeRF-based dense reconstruction with a modern point cloud segmentation network, the proposed pipeline enables accurate discrimination between pruning-required branches and retained branches in complex orchard environments, especially for thin and overlapping branch structures.

3. A branch-level pruning evaluation protocol is introduced to complement conventional segmentation metrics. Beyond standard point-wise metrics such as OA, mAcc, and mIoU, this study incorporates pruning-oriented indicators including FPB, FNB, and FNR, which more directly reflect the practical reliability of automated pruning decision-making.

4. Comprehensive experimental validation demonstrates the superiority of the proposed method in both geometric fidelity and pruning applicability. Quantitative comparisons with MVS-based reconstruction and classical point cloud segmentation networks show that the proposed method achieves higher structural accuracy, improved color consistency, and more reliable pruning branch identification, confirming its potential for robotic pruning and intelligent orchard management.

FUNDING

This work was funded by the China Scholarship Council (CSC) (202401040020, 202401040021)and Natural Science Foundation of Shandong Province (ZR2021MC026).

Compliance with Ethical Standards

The authors declare no potential conflicts of interest. This research did not involve human participants or animals, and therefore informed consent was not applicable.

Reference

[1]Zhang Dexue, Qin Xitian, Liu Xuefeng, et al. Research Progress and Supporting Equipment for Fruit Tree Pruning in China and Abroad[J]. China Fruits, 2021(02): 6-12. DOI: 10.16626/j.cnki.issn1000-8047.2021.02.003.

[2]He Long, Schupp James. Sensing and Automation in Pruning of Apple Trees: A Review[J]. Agronomy, 2018, 8(10): 211. DOI: 10.3390/agronomy8100211.

[3]Botterill T, Paulin S, Green R, et al. A Robot System for Pruning Grape Vines[J]. Journal of Field Robotics, 2017, 34(6): 1100-1122. DOI: 10.1002/rob.21680.

[4]Ma Baojian, Jiang Huanyu. Research Status of Visual Perception System for Dormant Period Fruit Tree Pruning Robot[J]. Agricultural Engineering, 2024, 14(02): 29-33. DOI: 10.19998/j.cnki.2095-1795.2024.02.004.

[5]Karkee M, Adhikari B. A Method for Three-Dimensional Reconstruction of Apple Trees for Automated Pruning[J]. Transactions of the ASABE, 2015, 58(3): 565-574. DOI: 10.13031/trans.58.10799.

[6]Hai T, Wang W, Yan F, et al. A Deep Learning-Based Approach to Apple Tree Pruning and Evaluation with Multi-Modal Data for Enhanced Accuracy in Agricultural Practices[J]. Agronomy, 2025, 15(5): 1242-1242. DOI: 10.3390/AGRONOMY15051242.

[7]Bai J, Xing H, Ma S, et al.Studies on Parameter Extraction and Pruning of Tall-spindle Apple Trees Based on 2D Laser Scanner[J]. IFAC PapersOnLine, 2019, 52(30): 349-354. DOI: 10.1016/j.ifacol.2019.12.564.

[8]Zhao Yonghui, Liu Xueyan, Lv Yong, et al. 3D Reconstruction of Individual Tree Skeletons Based on Laser Point Cloud Data[J]. Forest Engineering, 2024, 40(01): 128-134.

[9]Peng Xiaodong, He Jing, Shi Lei, et al. 3D Reconstruction of Individual Trees Based on Point Cloud Fusion of LiDAR and Kinect Camera[J]. Journal of Huazhong Agricultural University, 2023, 42(02): 224-232. DOI: 10.13300/j.cnki.hnlkxb.2023.02.028.

[10]Ren Dongyu, Li Xiaojuan, Lin Tao, et al. Three-Dimensional Reconstruction Method of Fruit Tree Branches Based on Kinect v2 Sensor[J]. Transactions of the Chinese Society for Agricultural Machinery, 2022, 53(S2): 197-203.

[11]Zhou Jingjing, Guo Xinyu, Wu Sheng, et al. Research Progress on Plant Three-Dimensional Reconstruction Based on Multi-View Images[J]. Journal of Agricultural Science and Technology, 2019, 21(02): 9-18. DOI: 10.13304/j.nykjdb.2018.0391.

[12]Gong Jinliang, Liu Binxiao, Wei Peng, et al. Three-Dimensional Reconstruction Method of Fruit Trees Based on Camera Pose Recovery and Neural Radiance Field Theory[J]. Transactions of the Chinese Society of Agricultural Engineering, 2023, 39(22): 157-165.

[13]Zhu Lei, Jiang Wei, Sun Boyan, et al. Three-Dimensional Modeling and Phenotypic Parameter Acquisition of Seedling Crops Based on Neural Radiance Fields[J]. Transactions of the Chinese Society for Agricultural Machinery, 2024, 55(04): 184-192+230.

[14]MüLLER T, EVANS A, SCHIED C, et al. Instant neural graphics primitives with a multiresolution hash encoding[J]. ACM Transactions on Graphics (ToG), 2022, 41(4): 1-15.

[15]Tang Hui, Liu Xin, Hu Biwei, et al. KD-NeRF: A Knowledge Distillation Enhanced Neural Radiance Field Reconstruction Method for Fruit Trees[J]. Computer and Modernization, 2025, (12): 66-73.

[16]Zhao Qiang, She Jiangfeng, Wan Qifeng, et al. Research Progress on the Application of Neural Radiance Fields to Large-Scale Realistic 3D Scene Visualization[J]. Journal of Remote Sensing, 2024, 28(05): 1242-1261.

[17]Zhang H, Furuhashi K, Kaizu Y. An efficient and low-cost 3D phenotyping framework for tomato seedlings via neural radiance fields and PointNet++[J]. Smart Agricultural Technology, 2026, 13101722-101722. DOI: 10.1016/J.ATECH.2025.101722.

[18]Sun X, He L, Jiang H, et al. Morphological estimation of primary branch length of individual apple trees during the deciduous period in modern orchard based on PointNet++[J]. Computers and Electronics in Agriculture, 2024, 220108873-. DOI: 10.1016/J.COMPAG.2024.108873.

[19]Kashefi A. PointNet with KAN versus PointNet with MLP for 3D classification and segmentation of point sets[J]. Computers & Graphics, 2025, 131104319-104319. DOI: 10.1016/J.CAG.2025.104319.

[20]Dong S, Fan X, Li X, et al. Automatic 3D Plant Organ Instance Segmentation Method Based on PointNeXt and Quickshift++[J]. Plant Phenomics, 2025, 7(3): 100065-100065. DOI: 10.1016/J.PLAPHE.2025.100065.

[21]Sun H, Ye Q, Chen Q, et al. Tree Canopy Volume Extraction Fusing ALS and TLS Based on Improved PointNeXt[J]. Remote Sensing, 2024, 16(14): 2641-2641. DOI: 10.3390/RS16142641.

[22]Karkee M, Adhikari B, Amatya S, et al. Identification of pruning branches in tall spindle apple trees for automated pruning[J]. Computers and Electronics in Agriculture, 2014, 103127-135. DOI: 10.1016/j.compag.2014.02.013.

[23]Zhang J, He L, Karkee M, et al. Branch detection for apple trees trained in fruiting wall architecture using depth features and Regions-Convolutional Neural Network (R-CNN)[J]. Computers and Electronics in Agriculture, 2018, 155386-393. DOI: 10.1016/j.compag.2018.10.029.