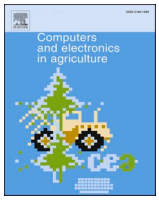
Computers and Electronics in Agriculture 229 (2025) 109708

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

journal homepage: www.elsevier.com/locate/compag



Original papers

# Apple tree architectural trait phenotyping with organ-level instance segmentation from point cloud

Lizhi Jiang [a,b], Changying Li [b,*], Longsheng Fu [a,*]

[a] College of Mechanical and Electronic Engineering, Northwest A&F University, Yangling, Shaanxi 712100 China
[b] Bio-Sensing, Automation, and Intelligence Laboratory, Department of Agricultural and Biological Engineering, University of Florida, Gainesville, FL, USA

ARTICLE INFO

*Keywords:*
Plant phenotyping
Apple tree
3D segmentation
PointNeXt
Point Transformer V2
SoftGroup++

ABSTRACT

Three-dimensional (3D) plant phenotyping techniques measure organ-level traits effectively and provide detailed plant growth information to breeders. In apple tree breeding, architectural traits can determine photosynthesis efficiency and characterize the developmental stages of trees. The overall goal of this study was to develop a deep learning-based organ-level instance segmentation method to quantify the 3D architectural traits of apple trees. This study utilized PointNeXt for the semantic segmentation of apple tree point clouds, classifying them into trunks and branches, and benchmarked its performance against several competitive models, including PointNet, PointNet++, and Point Transformer V2 (PTv2). A cylinder-based constraint method was introduced to refine the semantic segmentation results. Next, the branches were identified with the density-based spatial clustering of applications with noise (DBSCAN) algorithm. The type of 3D skeleton vertices determined whether a cluster represented a single branch or multiple branches. If multiple, a graph-based technique further separated them. This study also directly applied the instance segmentation model SoftGroup++ to the apple tree point clouds and analyzed the segmentation results on the apple tree dataset. Finally, seven architectural traits of apple trees were extracted, including height, volume, and crown width of the tree, as well as height and diameter for the trunk, and length and count for the branches. The experimental results showed that the post-processed mIoU values for PointNet, PointNet++, PTv2, and PointNeXt were 0.8495, 0.8535, 0.9500, and 0.9481, respectively. The final instance segmentation results based on SoftGroup++ and PointNeXt achieved mAP_50 of 0.815 and 0.842, respectively. For traits such as tree height, trunk length and diameter, branch length, and branch count, the method based on PointNeXt achieved $R^2$ values of 0.987, 0.788, 0.877, 0.796, and 0.934, with mean absolute percentage errors of 0.86 %, 2.17 %, 5.93 %, 10.24 %, and 13.55 %, respectively. The segmentation results of PTv2 and SoftGroup++ were also used to extract the phenotypic traits of apple trees, achieving results comparable to those of PointNeXt. The proposed method demonstrates a cost-effective and accurate approach for extracting the architectural traits of apple trees, which will benefit apple breeding programs as well as the precision management of apple orchards.

## 1. Introduction

Apples (*Malus domestica*), a globally cultivated fruit crop, hold significant prominence in agriculture (Velasco et al., 2010). Apples comprised 12.26 % of all fruit production in the world from 2012 to 2014, and the global apple cultivation area reached approximately 4.7 million hectares, with production surpassing 87.2 million tons (Wang et al., 2019; FAO, 2020). Owing to the health benefits and delicious taste of apples, as well as association with healthy lifestyles, the sustained demand for apples in the global market is steadily increasing, making apples a favorable choice in dietary preferences (Bohn and Bouayed, 2020). Given its importance, it has become necessary to develop new apple cultivars through breeding programs for desirable phenotypic traits.

Apple tree architectural traits, such as tree trunk height, number of branches, and branch lengths, determine the structure of an apple tree and affect light interception and thus the biomass of the tree (Béland and Baldocchi, 2021). The topology of tree branches represents the tree's gene expression and optimal adaptation to the environment. Thus, an accurate description of apple tree architecture leads to a better

* Corresponding authors.
*E-mail addresses:* cli2@ufl.edu (C. Li), fulsh@nwafu.edu.cn (L. Fu).

https://doi.org/10.1016/j.compag.2024.109708
Received 28 February 2024; Received in revised form 24 October 2024; Accepted 25 November 2024
Available online 4 December 2024

understanding of how the form is driven by the function (Lau et al., 2018). The number of branches determines the density of the leaves which are the organs of photosynthesis that produce organic matter. The trunk provides support for the growth of branches while transporting water and nutrients absorbed by the roots as well as the organic matter made by the leaves to ensure the normal growth of the fruit trees (Jurjević et al., 2020). The trunk height and diameter of single trees are important indicators to assist in describing the growth status and development trend of apple trees (Krause et al., 2019). Hence, an organ-level phenotypic trait collection system needs to be developed to quantify the trait parameters of apple trees.

Some traditional methods have been proposed to obtain plant phenotypes. Initially, phenotypic parameters were collected manually, which was time-consuming, labor-intensive, and error-prone because of human factors. With the development of computer vision technology, scientists analyze organ-level traits from two-dimensional (2D) images (Li et al., 2014). Identifying individual plant organs is a challenge for researchers due to the complex backgrounds in fruit tree images, which often include the sky, green leaves, fruit, and branches. Some researchers distinguish plant organs based on organ differences in traits such as colors, and they used thresholding segmentation based on color space or histogram threshold to maximize between-class variances (Ji et al., 2016; Xu et al., 2017). Nevertheless, this method requires hand-designed features and is not suitable for plants of different morphologies, exhibiting relatively weak generalization. Recently, deep learning has been widely applied in the field of agriculture due to its powerful capability for automatic feature learning (Saleem et al., 2021; Tan et al., 2023). Apple branches in a fruiting wall structure were detected by combining a region-based convolutional neural networks (CNN) with RGB images, pseudo-color images, and depth images, obtaining results with average recall and precision of 92 % and 86 %, respectively (Zhang et al., 2018). Faster R-CNN was proposed to perform automatic multi-organ object detection on apples, tree trunks, and branches in the natural environment, using VGG19 feature extraction network with an average accuracy (mAP) of 82.4 % (J. Zhang et al., 2020). SegNet was chosen for semantic segmentation of the trunk of the apple tree and achieved trunk and branch segmentation accuracy of 0.92 and 0.93, respectively (Majeed et al., 2018). In aligned 2D images and depth maps, it is also possible to detect regions of interest in the 2D images and determine the precise positions of apple tree trunk grafted interface in the point clouds (Sun et al., 2022). However, organ segmentation from 2D images is easily affected by occlusion and poses difficulty in extracting three-dimensional (3D) traits such as volume.

Capturing phenotypic traits through a 3D point cloud is gaining interest because it preserves spatial and plant structural features, overcoming the limitation of lack of depth information in 2D images. The point cloud can be collected directly by a Light Detection and Ranging (LiDAR) or RGB-D camera, and it has been used widely in plant biomass estimation (Wallace et al., 2017), crop architectural trait analysis (Bao et al., 2019; Guo et al., 2018; Mao et al., 2024), and plant organ counting (Sun et al., 2020; Jiang et al., 2022, 2024). The quantitative structure model (QSM) is a typical point cloud modeling method that quantifies the topological structure, geometric characteristics, and volumetric parameters of the tree (Disney et al., 2018). Skeleton extraction is another method commonly used in plant structural analysis from the point cloud and is often employed for organ-level segmentation to extract plant traits (Miao et al., 2021). The two aforementioned methods primarily rely on the geometric characteristics of plants. Machine learning methods such as support vector machines (SVM) and deep learning methods have also been used to segment point clouds and analyze plant architecture. For example, fruits were segmented from pomegranate trees based on color and shape features and then counted through clustering (Zhang et al., 2021). PointNet (Qi et al., 2017a) and PointNet++ (Qi et al., 2017b) make it possible to apply deep learning to point clouds. PointNet++ was used to segment cucumber plants at different growth stages, which helps to analyze the phenotypic traits of complex plants (Boogaard et al., 2021). SPGNet was used to detect and identify pruned branches and demonstrated that jujube trunks and branches can be successfully segmented with class accuracies of 0.93 and 0.84 respectively (Ma et al., 2021). PVCNN has been utilized in the organ-level segmentation of cotton plants, extracting seven architectural features with a mean absolute percentage error of less than 10 % (Saeed et al., 2023). Although 3D deep learning networks have been applied on several crops in the past three years, its performance on apple trees is yet to be evaluated. Additionally, while research has been conducted on many crops, the data was often collected using expensive instruments or indoors. There is limited research on using RGB-D cameras for outdoor data collection, particularly in the study of apple trees. Furthermore, the semantic segmentation models used on different crops so far have been surpassed by newer models. Therefore, exploring the latest models and validating an end-to-end instance segmentation model on apple tree point clouds is warranted.

Accurate organ-level phenotyping of individual apple trees from point cloud data collected in the field, however, remains a challenge. The point clouds obtained in the field are affected by uncontrolled lighting conditions. Apple trees are densely planted, which poses challenges for the subsequent segmentation of individual trees. In apple tree phenotypic studies, a conventional method using the hierarchical growing (HG) algorithm has been employed to extract branches' length, quantity, and distribution traits from LiDAR-collected point cloud data (Zhang et al., 2024). The TreeQSM method was utilized to analyze point cloud data obtained through LiDAR, facilitating the extraction of topological structural information related to apple tree branches. This method has been proven to be viable for researching fruit tree branches (Zhang et al., 2020). Additionally, a method for point cloud skeleton extraction is employed to extract phenotypic traits of apple trees (Qiu et al., 2022). These methods all employed traditional algorithms for the phenotypic extraction of apple trees, relied on the costly LiDAR sensor to collect data, or collected the data in controlled indoor environments (Ma et al., 2021). Although some studies have employed deep learning-based semantic segmentation models to segment tree organs (Sun et al., 2024), there has been no direct research on segmenting apple trees using point cloud instance segmentation models. It is therefore worthwhile to explore the use of low-cost RGB-D cameras combined with deep learning methods to analyze point clouds of apple trees in the field.

The overall goal of this study was to measure the architectural traits of apple trees through a low-cost RGB-D camera and 3D deep learning models. Specific objectives were to (1) collect the point cloud data of apple trees using a low-cost RGB-D camera in the field; (2) develop a PointNeXt-based 3D segmentation workflow for organ-level instance segmentation of single apple trees and benchmark with several other competitive models; and (3) extract and evaluate architectural traits of apple trees.

## 2. Materials and methods

This workflow mainly includes three steps: semantic segmentation, instance segmentation, and phenotypic trait extraction (Fig. 1). A two-stage approach segmented a single apple tree point cloud into instance parts. In the first stage, the apple tree was segmented into branches and trunks by the 3D deep learning model PointNeXt (Qian et al., 2022), and the semantic segmentation results were corrected through post-processing based on cylinder constraints. The main task of the second stage was to perform instance segmentation of the branches that were in contact with the trunk. First, cluster the branches separated in the first stage based on density-based spatial clustering of applications with noise (DBSCAN), then extract the skeleton of each cluster and classify the skeleton vertices, and then judge whether the cluster contains multiple branches. Convert the skeleton of a cluster with multiple branches into a directed graph, and divide the directed graph into subgraphs, where each subgraph corresponds to a branch. Eventually, an apple tree was split into individual instance parts. Then, phenotypic

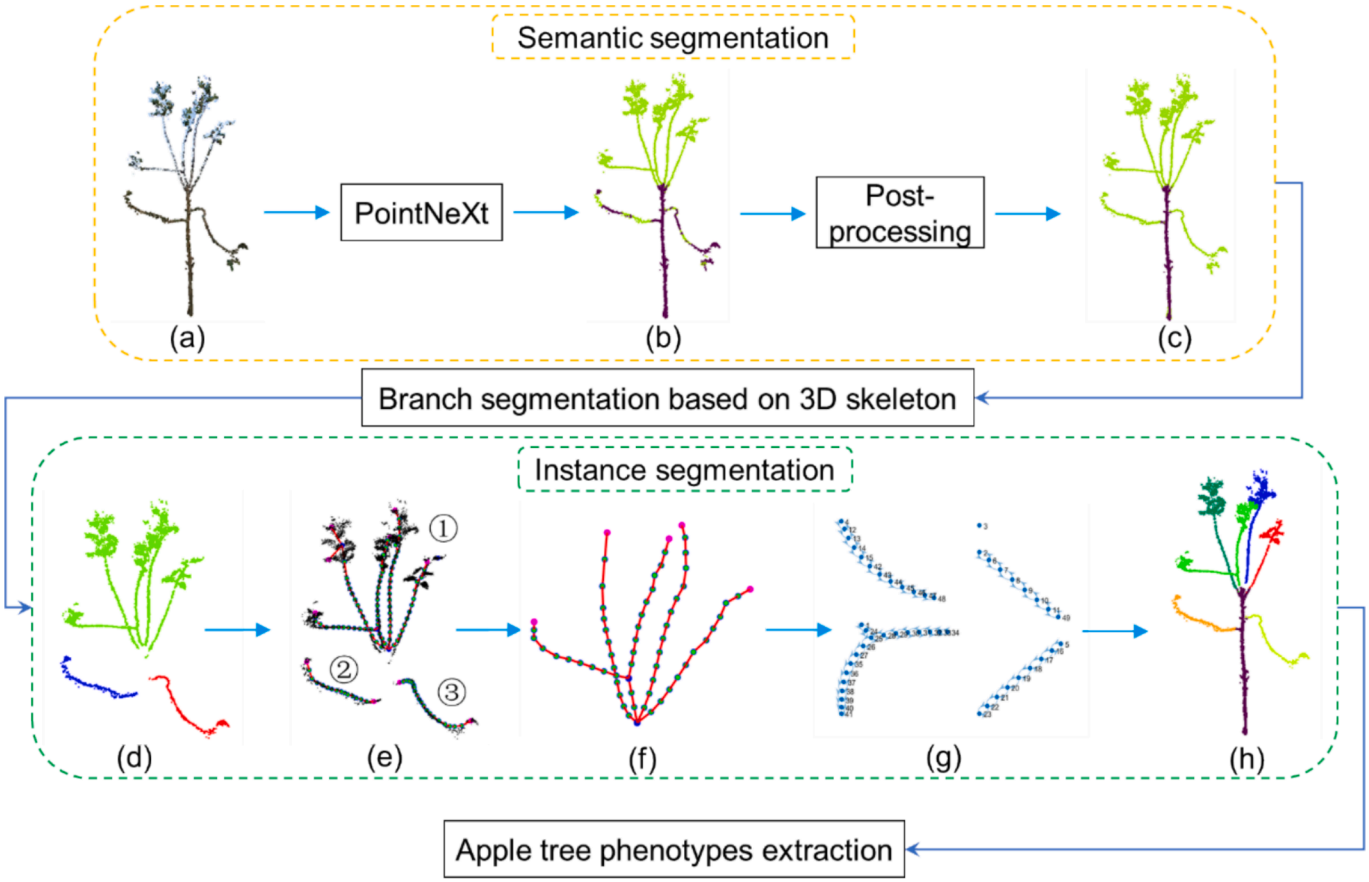



**Fig. 1.** 3D apple tree part segmentation and traits extraction pipeline. The whole process first segments a single apple tree into branches and trunks, then segments the branches into individual parts, and finally extracts the phenotype traits of each part. (a) Raw point cloud. (b) Semantic segmentation results of PointNeXt. (c) Corrected segmentation results. (d) DBSCAN clustering. (e) Skeleton extraction based on Laplacian. The skeleton is extracted for each cluster and the vertices are categorized based on the number of neighbors. (f) A skeleton containing multiple branches. (g) Skeleton decomposition based on the graph. (h) Instance segmentation results.

traits were extracted from the segmented apple tree organs.

### 2.1. Data collection

The apple tree materials used in this study were collected from Qian County (latitude: 34.33 N, longitude: 108.71 W) in Shaanxi Province, China, in October 2020 (Fig. 2a). A Kinect V2 sensor (Microsoft Corp., Redmond, WA, USA) mounted on a tripod was used to scan apple tree samples randomly selected in each row to acquire point cloud data (Fig. 2b-c). The height of the sensor was about 1.0 m, and the distance from the trunk was about 1.5 m for apple tree data collection. Meanwhile, during the acquisition process, the sensor was adjusted slightly to ensure that a complete apple tree was in the center of the image. The RGB images have a resolution of 1920 × 1080, and the point cloud contains x, y, and z coordinates and RGB color information (Fig. 2d). The apple trees were cross-bred (Yan 6 × HuaShuo, YingWei × MiCui), and the age of the trees was 2 years old. The trees were pruned one year prior to the data collection and had grown new branches. The distinctive characteristics of the apple trees are as follows: The trunk stands erect, providing a sturdy central structure. Some branches exhibit a unique growth pattern, spiraling around the trunk, while the rest of the branches form a dense cluster at the tree's summit. According to the growth characteristics of plants, the branches generally grow upward and outward. In this study, a total of 46 apple tree point clouds were collected, with 34 used for training and 12 for testing.

### 2.2. Point cloud data processing

Preprocessing was applied before point cloud segmentation (Fig. 3). First, an individual apple tree was obtained by removing irrelevant point clouds manually with the open-source software CloudCompare V2.12 alpha. Then the noise points that would affect the segmentation results during data acquisition were removed using a statistical outlier removal (SOR). The point cloud of the apple tree has been meticulously categorized into two distinct classes: branches and trunks. These structured data have been formatted to align with the conventions of the ShapeNet dataset. The raw point cloud in the figure only shows the coordinates and hides the RGB information, and the labeled point cloud uses different colors to represent different categories.

### 2.3. Semantic segmentation of apple tree

An apple tree was semantically segmented into branches and trunks using a trained semantic segmentation model on the labeled point cloud, followed by the application of a cylinder rule-based constraint to correct mis-segmented points, accurately executing the segmentation process.

#### 2.3.1. Semantic segmentation of apple tree based on PointNeXt

PointNeXt is an expansion of PointNet++, enhanced by innovative structural changes and the integration of Inverted Residual Multi-Layer Perception (InvResMLP) blocks (Fig. 4). The model begins with an MLP layer to map the input point cloud to higher dimensions, and then progresses through a series of Set Abstraction (SA) modules, each

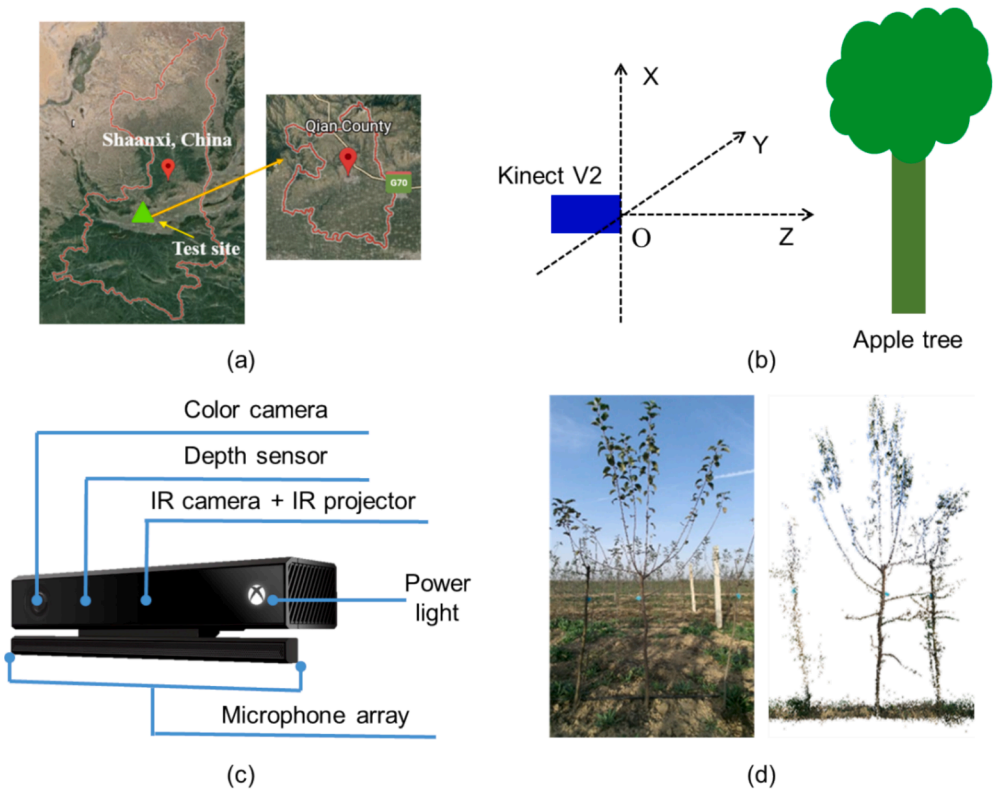



**Fig. 2.** Overview of data collection location and methods. (a) The Experimental site. (b) Data capturing method from one side of apple trees. (c) Kinect V2 sensor used for data collection. (d) Sample output of the RGB image and corresponding point cloud of an apple tree.

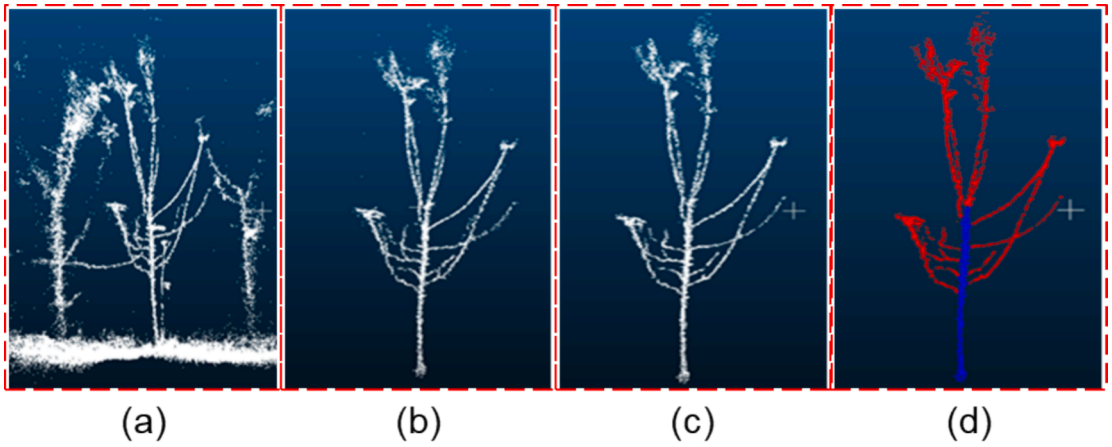



**Fig. 3.** Sequence of preprocessing steps for point cloud data. (a) Raw point cloud. (b) Cropped point cloud following the removal of the ground plane. (c) Enhanced point cloud after denoising. (d) segmentation and annotation: color-coded for clarity, with blue indicating the trunk and red highlighting the branches. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

followed by an InvResMLP block. These InvResMLP blocks significantly bolster the model's depth and width, optimizing its performance. The model's scalability is controlled through width and depth scaling strategies, increasing the channel number C and the number of InvResMLP blocks B, respectively. The InvResMLP, built on the SA module, incorporates residual connections to mitigate the issue of vanishing gradients. It introduces separable MLPs to decrease computational load and enhance point-wise feature extraction. Moreover, the adoption of an inverted bottleneck design significantly improves the capability of feature extraction.

In the PointNeXt architecture, after feature extraction through the SA and InvResMLP blocks, the features at various levels need to be combined or propagated back to the original point cloud resolution. Feature Propagation allows the network to recover spatial information that might have been lost during downsampling processes, which is achieved by interpolating features from coarse to fine resolution and then merging these with the features from earlier layers. This process ensures that the detailed local information is not lost, and the final prediction is both globally consistent and locally detailed.

The performance of the model was evaluated by the training and testing results. Experiments were performed on the HiPerGator high-performance computing cluster with 8 AMD EPYC ROME CPU cores, and one NVIDIA DGX A100 GPU node (80 GB). The model was trained on the dataset for 500 epochs. The training data used in this experiment

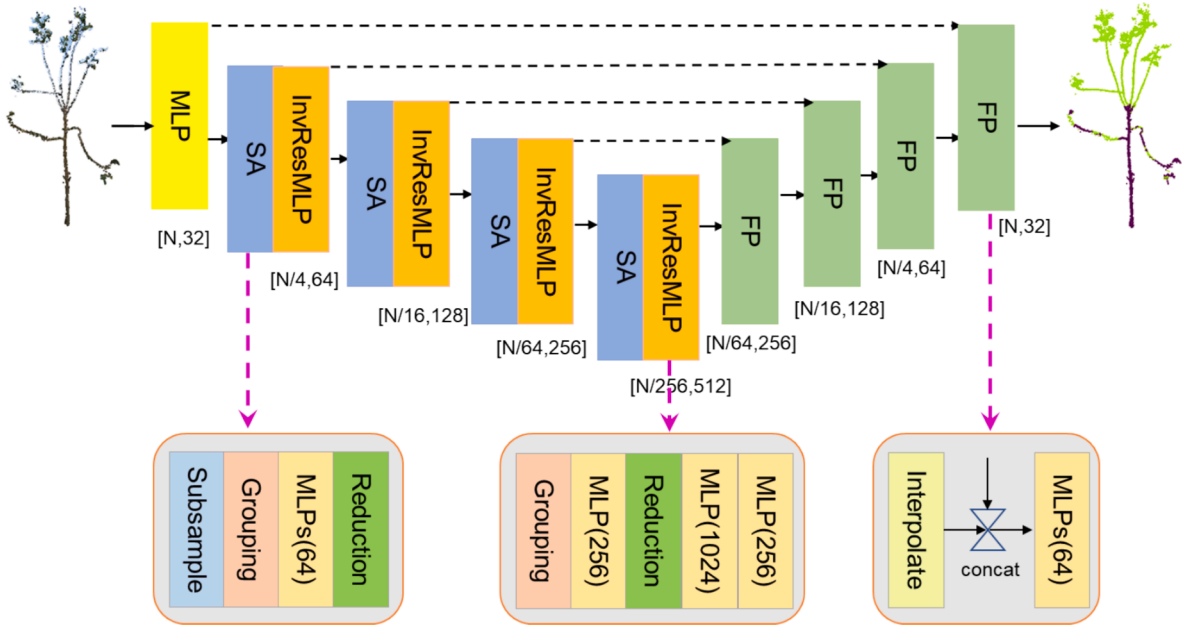



**Fig. 4.** PointNeXt architecture. The model consists of an encoder and a decoder. The encoder hierarchically abstracts features of point clouds using set abstraction (SA) blocks, while the decoder gradually interpolates the abstracted features by the same number of feature propagation (FP) blocks. An Inverted Residual MLP (InvResMLP) block is closely followed each SA block to implement model scaling.

was 34 samples and the test data was 12 samples. Data augmentation and optimization techniques are effective methods to modernize model training strategies. Data augmentation boosts network performance, especially with small datasets. PointNeXt updated more methods such as data scaling (e.g. point clouds resampling (Yu et al., 2022) and loading the whole scene as input (Hu et al., 2020), random scaling, random rotation, translation to shift point clouds, jittering to add independent noise to point clouds, height appending (Thomas et al., 2019), color auto-contrast (Zhao et al., 2021) and color drop that randomly replaces colors with zero values. During model training, different datasets for PointNeXt adopted varying data augmentation strategies. In this study, the ShapeNet data format was used. When training PointNet and PointNet++ models, data shift and rotation were applied. For PointNeXt, the default augmentation methods from the original code were used, including data scaling, jittering, and randomly dropping color information. Optimization techniques including loss functions, optimizers, learning rate schedulers, and hyperparameters are also important to the performance of a network. In general, cross entropy loss with label smoothing, AdamW, and Cosine Decay can decently optimize models in various tasks. PointNeXt verifies the effectiveness of train strategies and only keeps the augmentations that give a better validation accuracy. In this study, the search radius of the point cloud is 0.01 m, and the scaling of the radius is 2. The hyperparameter settings for data augmentation are as follows: scale is [0.8, 1.2], gravity_dim is 2. The default values were used for the remaining parameters in the original code of PointNeXt.

### 2.3.2. *Semantic segmentation of apple tree based on Point Transformer*

Point Transformer V2 (PTv2) (Wu et al., 2022) is a recently developed effective and efficient point cloud semantic segmentation model that improves upon the original Point Transformer (Zhao et al., 2020). It introduces several improvements, including Grouped Vector Attention (GVA), Position Encoding Multiplier (PEM) and Partition-based Pooling. As the network deepens and the number of feature encodings increases, the number of parameters in the weight encoding layers grows rapidly. To address this challenge, GVA enhances the model's efficiency and generalization by partitioning the channels of the value vectors into multiple groups, with shared weights within each group. Due to the uneven distribution of point clouds in continuous Euclidean space, their spatial relationships are highly complex. The introduction of PEM enhances position encoding, enabling the model to better capture and learn these intricate spatial relationships. In the original pooling method, the point cloud density and overlap between query sets are uncontrollable. Therefore, Partition-based Pooling is used to divide the point cloud into non-overlapping subsets by employing a uniform grid to partition the point cloud space.

The dataset used in the experiments is in S3DIS format, and the training was conducted on HiperGator as described in an earlier section. During the training process of PTv2, the sampling grid was set to three different resolution levels— [0.02, 0.04, 0.06] meters—based on the density of the apple tree point clouds. Smaller grids capture finer geometric details, while larger grids focus more on the overall geometric structure. In the experiments, the batch size was set to 2, the optimizer was AdamW, and the initial learning rate was 0.006. All other hyperparameters were kept as the default settings from the original code.

### 2.3.3. *Semantic segmentation of apple tree based on PointNet series*

PointNet (Qi et al., 2017a) is a deep learning model designed for processing point clouds, capable of directly extracting features from unordered point sets, thereby facilitating effective learning and recognition of point cloud data. It consists of three key modules. First, a max pooling layer is utilized as a symmetric function to aggregate information from all points, ensuring invariance to point arrangement. The Local and Global Information Aggregation module connects global features with each point feature, allowing the extraction of new point features from the combined data through a Multilayer Perceptron (MLP), which enables the simultaneous understanding of both local and global information. Lastly, two joint alignment networks align the input points and point features, making the network resilient to both affine and rigid transformations.

PointNet++ (Qi et al., 2017b) is an improved version of PointNet, enabling it to learn the relationships between points within the domain. It completes the down-sampling and up-sampling of the point cloud by the two modules of set abstraction (SA) and feature propagation (FP), and gradually captures the feature information in the point cloud. The SA module consists of sampling, grouping, and PointNet, which

performs sub-sampling, region proposal, and point feature extraction on point cloud. Stacking multiple SA layers allows for capturing features at different scales. FP is used to propagate high-level features to lower levels, thereby generating dense point cloud features. The purpose of stacking multiple FP layers is to progressively recover and refine feature information, enabling the model to produce more accurate and detailed point clouds. Each FP layer gradually propagates high-level features to lower levels and uses the inverse distance weighting method for interpolation, achieving step-by-step feature refinement.

#### 2.3.4. Semantic segmentation post-processing operations based on cylindrical constraints

A cylinder constraint-based approach is utilized to correct the semantic segmentation results (Fig. 5). The whole process of post-processing consists of two steps. The first step involves redefining the trunk area. This is essential for constructing cylindrical constraints, as the fitting line for the trunk is required. However, semantic segmentation sometimes results in the mistaken segmentation of branches as part of the trunk, leading to inaccuracies in the fitting line. K-means was used to determine the centroid of the trunk, which was always close to the trunk (shown as the red in Fig. 5b). A circle was designed with the centroid's y and z coordinates as the center and a 0.1 m radius. The trunk was projected onto the YOZ plane, and points within the circle were identified as the optimized trunk area. The second step is to design cylinder rules based on the optimized trunk area. Firstly, the spatial straight line of the tree trunk was fitted based on the singular value decomposition (SVD), and a cylinder was constructed with the straight line as the center. This approach addresses the influence of camera angles during data collection, ensuring the fitted line accurately reflects the trunk's true direction. The height of the cylinder is the height of the optimized trunk area, and the radius is set to 0.05 m (Fig. 5c). Finally, the semantic segmentation was refined using the cylindrical constraint: points within the cylinder were classified as the trunk, while those outsides were marked as branches (Fig. 5d).

### 2.4. Instance segmentation of apple tree

#### 2.4.1. Branch instance segmentation based on 3D skeleton and graph

Branch instance segmentation was achieved by a 3D clustering method DBSCAN, skeleton extraction, and graph-based rules (Fig. 6). DBSCAN was used to cluster the branches segmented by PointNeXt, where the two parameters of the maximum search radius and the minimum number of points required to form a dense area were set to 4.5 cm and 6 respectively. Each small cluster is either a single branch or multiple branches connected together, which requires further segmentation. At the same time, this experiment removes some clusters that are too small. By comparing the axis-aligned bounding box (AABB) of each cluster and obtaining the longest side of the box. If the edge is smaller than 20 cm, the point cloud in the cluster is deleted.

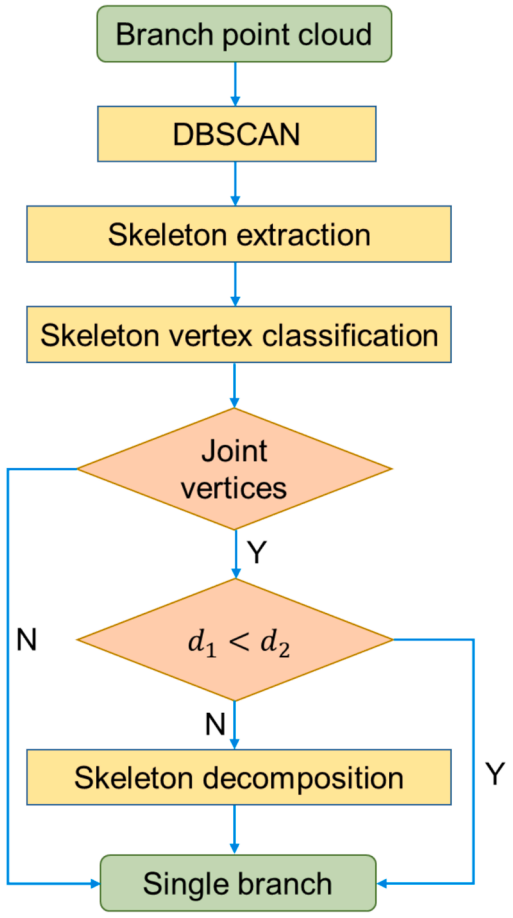



**Fig. 6.** Branch instance segmentation workflow. $d_1$ represents the distance from vertex 1 to the main stem, and $d_2$ represents the distance from vertex 2 to the trunk.

Next, a method based on point cloud skeleton vertex classification and graph was used to achieve instance segmentation of branches. This approach involves four main steps: skeleton extraction, skeleton vertex classification, determining if the cluster is a single or multiple branches, and decomposing multiple branches into single ones using graph theory.

**Skeleton extraction**. The Laplacian-based method (Cao et al., 2010), which is robust to noise and can handle a moderate amount of missing data, was used to generate the branch skeleton. Given an apple

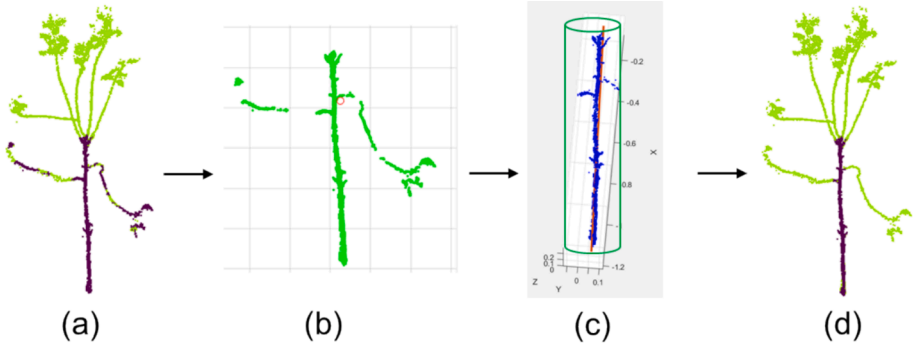



**Fig. 5.** Post-processing of point cloud segmentation results using cylindrical constraints. (a) Example of initial semantic segmentation results. (b) Centroid determination of the trunk class, highlighted with a red circle. (c) Cylinder modeling illustrating the trunk (in blue), its fitting line (in red), and the constructed cylinder (in green). (d) Corrected semantic segmentation results. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

branch point cloud $P$, first, find the $K$ nearest neighbors of each point in $P$ and calculate the Laplace coordinates. The Laplace contraction operation was used to contract $P$ to a zero-volume point set $C$. Then the farthest point sampling was used to obtain the skeleton vertex set $U$ from $C$, and each vertex in $U$ corresponded to a little point cloud set in $P$. Then the adjacent skeleton vertices in $U$ were connected to generate the edge set $E1$. The edge contraction operation was used to remove the redundant edges in $E1$ to form the final edge set $E$. Thus, the graph $G = (U, E)$ represented by the skeleton vertex set $U$, and the edge set E can be obtained, which is the apple tree skeleton. In this study, $K$ was set to 16. The sampling scale of the skeleton vertices was set to 0.05 m. The larger the value, the farther the distance between two vertices and the sparser the skeleton. The remaining parameters were set to default values in the original paper.

**Classification of skeleton vertices**. The categorization of skeleton vertices is essential for decomposing individual branches. A 3D branch skeleton consists of skeleton vertices and edges, and defines vertices on the same edge as adjacent (Fig. 7). Vertices fall into one of three classifications, determined by the number of neighboring vertices: (1) A Terminal vertex, which is connected to only one neighbor; (2) A Joint vertex, which has connections to three or more neighbors; and (3) A Connect vertex, which is characterized by two neighboring vertices.

**Distinguishing a single branch from multiple branches**. The branches were clustered into three cases based on the biological characteristics of upward and outward growth (Fig. 7): (1) Single branch with no joint vertices. (2) Single branch with sub-branch. If there are joint vertices, it needs to be judged again. Next, calculate the distances $d_1$ and $d_2$ from vertex 1 (terminal) and vertex 2 (Joint) to the trunk, respectively. If $d_1 < d_2$, it is a single branch; (3) Multiple branches. If at least one joint vertex is closer to the trunk than the terminal vertex, the cluster is composed of multiple branches. From the above definition, it is clear that (1) and (2) are single branches and do not need to be segmented again, while (3) consists of multiple branches and needs to be segmented again.

**Graph-based methods decompose clusters with multiple branches**. After the extracted skeleton vertices were classified by the number of neighbors of the skeleton vertices, the joint point with the smallest Euclidean distance from the trunk was determined (the bottom blue point in Fig. 8a). The adjacency matrix of each vertex of the point cloud skeleton can be obtained, and the 3D skeleton can be directly converted into a directed graph through the adjacency matrix (Fig. 8b). The directed graph consists of vertices and edges and maintains the neighborhood relationship of the original skeleton. Next, the directed graph was disconnected at the vertex (vertex 3) closest to the trunk, resulting in multiple subgraphs, each subgraph corresponding to a branch (Fig. 8c). In the skeleton vertex set $U$, each vertex saves a corresponding original point cloud set. We discard the point cloud corresponding to vertex 3. Finally, the point clouds corresponding to the vertices in each branch were merged to form a single branch point cloud set to complete the instance segmentation.

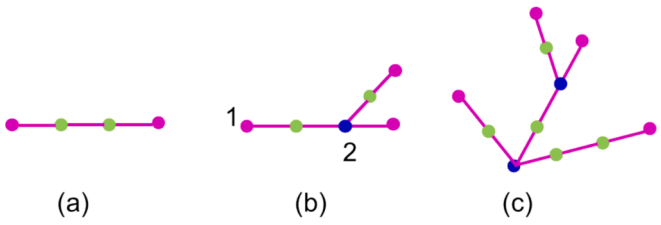



**Fig. 7.** Distinguishing single branch and multiple branches in point clouds after DBSCAN clustering. (a) single branch. (b) single branch with sub-branches. (c) multiple branches. Pink is the terminal vertex. Blue is the joint vertex. Green is the connecting vertex. All vertices are connected by pink edges. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

#### 2.4.2. Instance segmentation of apple tree based on SoftGroup++

SoftGroup++ (Vu et al., 2024) is an advanced 3D instance segmentation network that can segment point clouds into individual instances for different categories. The architecture of SoftGroup++ comprises two stages: bottom-up grouping and top-down refinement. The model takes a point cloud as input, with a U-Net backbone initially employed to extract point features. The semantic and offset branches (offset prediction module) predict semantic scores and offset vectors, which are then used by the soft grouping module to generate instance proposals. A feature extractor layer subsequently extracts backbone features from these proposals. The features of each proposal are then processed through a smaller U-Net, which provides classification, segmentation, and mask scoring branches for the point cloud, ultimately producing the final instances. During training, the dataset was formatted in S3DIS format, and the training was conducted on the HiperGator cluster as described earlier. The hyperparameters used during training were the same as those used for training on the publicly available S3DIS dataset.

#### 2.4.3. Instance segmentation of apple tree based on JSIS3D

JSIS3D (Pham et al., 2019) combines semantic information to achieve final instance segmentation. The model first scans the input point cloud using overlapping 3D windows, segmenting it into blocks fed into the multi-task pointwise network module (MT-PNet). Within this module, both semantic labels and instance embeddings are predicted simultaneously. The results are then passed to the multi-value conditional random field module (MV-CRF), where semantic and instance labels are jointly optimized to produce the final instance segmentation of the point cloud. MT-PNet is an improvement based on PointNet, primarily modifying the output into two branches: one for a semantic segmentation head and the other for an instance embedding head. The MV-CRF module was designed for 3D scene point clouds, optimizing each point's semantic and instance labels simultaneously to achieve joint semantic and instance segmentation. During training, the data set is still formatted in S3DIS format.

### 2.5. Phenotypic trait extraction

Volume, height, and crown width were extracted from individual apple tree point clouds (Table 1). There are three methods to evaluate tree volume: convex hull (Fig. 9a), concave hull (Fig. 9b) and voxel-based (Fig. 9c). In apple tree volume estimation based on voxel, the OBB was segmented into small grids, and the grids were colored red when they contained point clouds. Next, count the number of red squares to calculate the tree volume. In actual measurement, the point cloud can be divided into small blocks (such as $80 \times 40 \times 30$) according to its density. The tree height, east–west crown width, and north–south crown width were obtained from the oriented bounding box (OBB) of the point cloud (Fig. 9f), and the average value of the east–west crown width and the north–south crown width was taken as the tree crown size.

Trunk height and diameter were extracted from the segmented trunks (Table 1). Trunk height ($H_T$) was estimated by calculating the absolute value of the difference between the maximum and minimum values of the trunk point cloud x-axis. Trunk diameter was defined as the position at 10 cm of the trunk height (Fig. 9d). Because of the uneven density of the point cloud, the point cloud within 1 cm above and below the diameter position was selected on the x-axis for cylindrical fitting, and the cylinder diameter was taken as the trunk diameter.

$$H_T = |x_{Tmax} - x_{Tmin}| \tag{1}$$

where $H_T$ is the trunk height, $x_{Tmax}$ is the highest point of the trunk, $x_{Tmin}$ is the lowest point of the trunk.

Branch number and length were estimated from individual branches (Table 1). The number of branches was the total number of branches on a tree after instance segmentation. In this study, only the primary

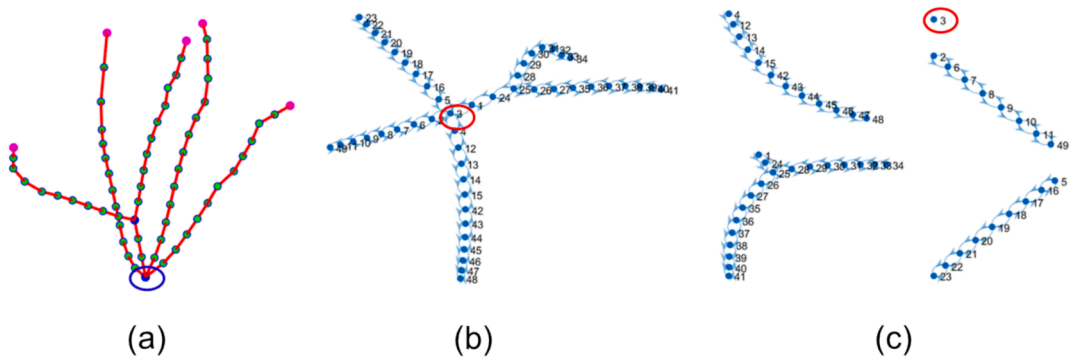



**Fig. 8.** Skeleton decomposition based on graph theory. (a) Point cloud skeleton. Pink vertices indicate there is only one neighbor. A green vertex indicates that it has two neighbors. Blue vertices indicate more than two neighbors. (b) A directed graph generated by branch skeletons. (c) A directed graph decomposition. The circled vertex in the figure represents disconnections here. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

**Table 1**
Phenotypic traits and corresponding measurement methods. $x_{Tmax}$ represents the maximum value of the x-axis coordinates on the trunk. $x_{Tmin}$ represents the minimum value of the x-axis coordinates on the trunk.

| Point cloud | Traits | Methods |
|---|---|---|
| Single apple tree | Tree volume | Convex hull, concave hull, voxel-based |
| | Tree height | OBB |
| | Crown diameter | OBB |
| Trunk | Trunk height | $H_T = \lvert x_{Tmax} - x_{Tmin} \rvert$Cylindrical fitting |
| | Trunk diameter | |
| Single branch | Branch length | PCA |
| | Number of branches | Instance segmentation |

branches in direct contact with the trunk were counted, and there were no statistics for the secondary and tertiary branches that grew on the primary branches. Branch length (Fig. 9e) was defined as the shortest path between the two endpoints of the first principal component axis obtained by principal component analysis (PCA).

### 2.6. Evaluation metrics

To evaluate the robustness and efficiency of the method, metrics based on a confusion matrix were used to visualize its performance. The confusion matrix is the summary of the prediction results of the classifier, and it mainly consists of the number of true positives (TP), the number of false positives (FP), the number of true negatives (TN), and the number of false negatives (FN). Intersection over union (IoU), and mean intersection over union (mIoU) were used to evaluate the performance of the point cloud segmentation algorithm. IoU indicates the similarity of the predicted region of a category to the ground truth region (Eq. 2). mIoU is the average of IoU over all categories and was used to evaluate the overall performance of the segmentation, where $N$ was the total number of labels, and $i$ was the $i^{th}$ label (Eq. 3). Accuracy (Eq. 4) is the ratio of the number of correctly classified samples to the total number of samples, where mAcc is mean of class-wise accuracy and allAcc is overall point-wise accuracy.

To assess the robustness of the model on the apple tree dataset, this study conducted a 4-fold cross-validation. The 46 datasets of −point clouds were divided into four groups: 12, 12, 12, and 10. In each fold, one group was used as the test set, while the remaining groups served as the training set. Using mIoU as the evaluation metric, this study performed 4-fold cross-validation on PointNet, PointNet++, PTv2, and PointNeXt models. Based on the mIoU obtained by 4-fold cross validation, the standard deviation (SD) can be calculated, which indicates the stability of the model on different datasets.

Precision (P) recall (R) and mean average precision (mAP) were used to evaluate the performance of the instance segmentation. P (Eq. 5) and R (Eq. 6) denote the proportion of correctly predicted points to the total predicted points and total ground truth points, respectively. AP (Eq. 7) is the area under the PR curve, used to summarize precision and recall. The AP for all classifications is averaged to obtain the mAP (Eq. 8). When IoU is 0.5, this study refers to it as mAP_50.

$$IoU_i = \frac{TP_i}{TP_i + FP_i + FN_i} \tag{2}$$

$$mIoU = \frac{\sum_i^N IoU_i}{N} \tag{3}$$

$$Accuracy = \frac{TP_i + TN_i}{TP_i + FP_i + FN_i + TN_i} \tag{4}$$

$$P = \frac{TP}{TP + FP} \tag{5}$$

$$R = \frac{TP}{TP + FN} \tag{6}$$

$$AP = \int_0^1 P(R_i) dR_i \tag{7}$$

$$mAP = \frac{1}{C} \sum_{n=1}^{c} AP_i \tag{8}$$

In addition, the trait extraction algorithm needed to be evaluated. Assume that the number of categories predicted by the algorithm was $N_i$, the number of manually labeled categories is $m_i$, and $n$ is the number of plants. There is also the root mean square error (RMSE), mean absolute error (MAE) and mean absolute percentage error (MAPE). The coefficient of determination ($R^2$) was also calculated to assess the performance.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (N_i - m_i)^2} \tag{4}$$

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |N_i - m_i| \tag{5}$$

$$MAPE = \frac{100\%}{n} \sum_{i=1}^{n} \left| \frac{N_i - m_i}{m_i} \right| \tag{6}$$

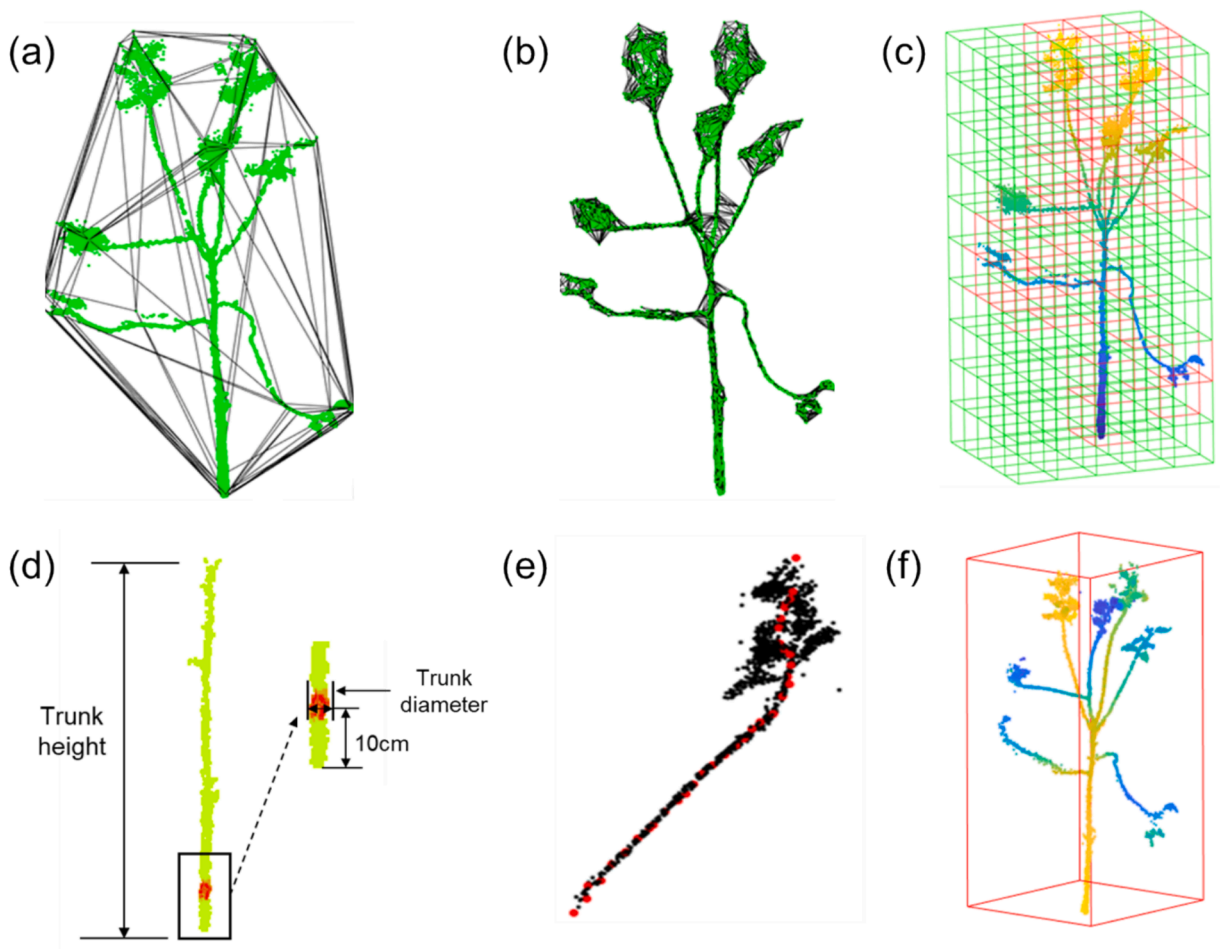



**Fig. 9.** Methods for phenotypic trait quantification. (a) Volume estimation using a convex hull. (b) Volume measurement with a concave hull. (c) Volumetric analysis via voxels: red grids indicate the presence of point clouds, while green grids signify their absence. (d) Assessing trunk height and diameter. (e) Branch length determination using principal component analysis (PCA). (f) Determining plant height and crown width via an oriented bounding box (OBB). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

## 3. Results

### 3.1. Performance of point cloud semantic segmentation

A comparison of semantic segmentation results from different models revealed that PointNeXt exhibits outstanding segmentation performance along with a more lightweight model size on apple tree dataset (Table 2). It is worth noting that PointNeXt and PTv2 achieved similar mIoU results in testing, both surpassing PointNet and PointNet++. We averaged the results from the four-fold cross-validation, and PointNeXt achieved an mIoU of 0.943, surpassing PointNet by 16.5 % and PointNet++ by 9.6 %. These findings underscored the superior performance of PointNeXt in accurately delineating objects in the context of the apple tree dataset. PTv2 is also a very competitive semantic segmentation model with comparable results as PointNeXt. One notable advantage of PointNeXt is its lightweight architecture, with a model size of only 11.9 MB. In addition, compared to PTv2, PointNeXt requires fewer computational resources and trains at a faster speed. This highlights its efficiency in resource utilization and its convenience for cross-platform deployment. In the four-fold cross-validation, PointNeXt demonstrated greater robustness on different test data with the lowest SD. Qualitative results revealed that PointNeXt is robust, with an SD of only 0.0037 (Fig. 10).

**Table 2**
Segmentation performance of different semantic segmentation models on the apple tree dataset, and mIoU as the evaluation metric. SD stands for standard deviation.

| | PointNet | PointNet++ | PTv2 | PointNeXt |
|---|---|---|---|---|
| Fold 1 | 0.763 | 0.836 | 0.950 | 0.940 |
| Fold 2 | 0.767 | 0.866 | 0.934 | 0.945 |
| Fold 3 | 0.776 | 0.853 | 0.942 | 0.941 |
| Fold 4 | 0.807 | 0.832 | 0.950 | 0.948 |
| Average | 0.778 | 0.847 | 0.944 | 0.943 |
| SD | 0.0199 | 0.0136 | 0.0077 | 0.0037 |
| Model size (MB) | 95.5 | 20 | 45.5 | 11.9 |

Compared to traditional methods, semantic segmentation based on deep learning offers enhanced flexibility without the need for manually designed features or numerous steps, thereby preventing the accumulation of errors during computation (Zhang et al., 2024). Unlike 2D images, extracting traits from 3D point clouds offers several advantages (An et al., 2017; Shi et al., 2019; Ni et al., 2021), such as mitigating the issue of occlusion between branches and fruit (Mu et al., 2023; Sapkota et al., 2023). Furthermore, the apple tree point clouds we collected are consistent with their physical scale, allowing for direct extraction of object scale traits from the point clouds, unlike in 2D (Majeed et al., 2018), where a calibration board needs to be placed, such as tree height and branch length. In comparisons with networks such as PointNet and PointNet++, PointNeXt has been identified as a more accurate and lightweight model for segmentation. PointNeXt has shown great advantages in apple tree point clouds.

Post-processing was applied to the initial semantic segmentation

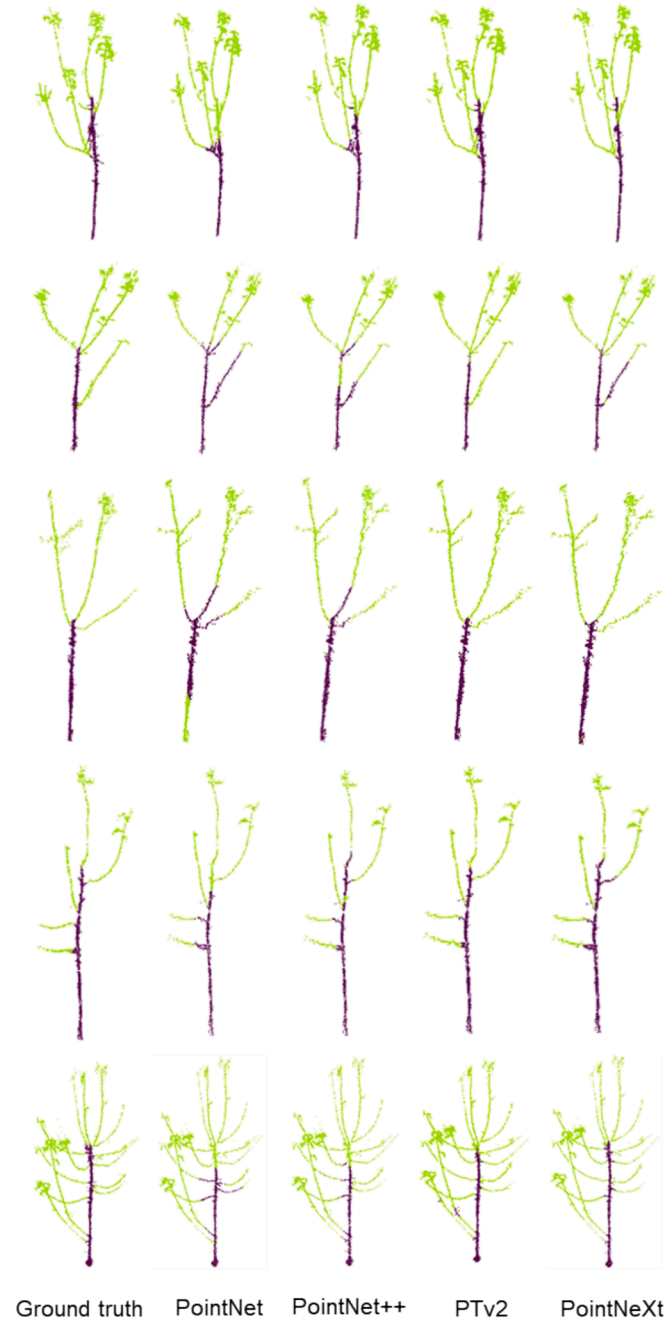



**Fig. 10.** Visualization of the semantic segmentation results from different models. Each row represents one apple tree sample. Green points represent the branches, and dark-colored points represent the trunk. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

results from various models and the results revealed that the models with poorer initial segmentation performance (such as PointNet and PointNet++) experienced more noticeable mIoU gains than those models performed well initially (such as PointNeXt and PTv2). The semantic segmentation results of PointNet showed the highest gain in segmentation performance due to post-processing (test data from Fold 1). However, post-processing is based on the original semantic segmentation results, so it only led to limited improvement over the initial performance. Even after post-processing, PointNet and PointNet++ still show a significant gap compared to PointNeXt and PTv2 (Table 3). As shown in Fig. 11, the error of the semantic segmentation result after post-processing is mainly near the trunk.

**Table 3**
Comparison of model performance after post-processing.

| Model | mIoU | Increase (%) |
|---|---|---|
| PointNet | 0.8495 | 8.70 |
| PointNet++ | 0.8535 | 1.75 |
| PTv2 | 0.9500 | 0 |
| PointNeXt | 0.9481 | 0.81 |

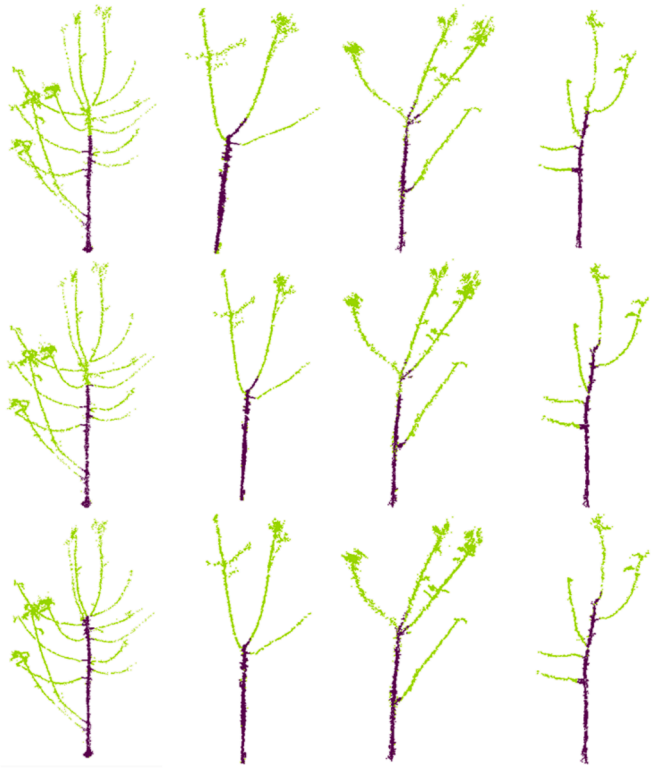


**Fig. 11.** Post-processed semantic segmentation results. The first row shows PointNet, the second row shows PointNet++, and the third row shows PointNeXt.

### 3.2. Performance of point cloud instance segmentation

The apple tree point cloud was successfully segmented into trunks and individual branches by our approach (PointNeXt with postprocessing methods) and the results revealed that the mAP_0.5 achieved by our approach was 37 % and 2.7 % higher than that of JSIS3D and SoftGroup++, respectively. Our method is highly competitive, although there are some errors at the intersections between the trunk and branches, as well as between different branches. In the comparative models, JSIS3D struggles to achieve the required segmentation on apple tree point clouds. On the other hand, SoftGroup++ proves to be a valuable model for our dataset, but it also exhibits noticeable errors, such as incorrect segmentation at the points where the main trunk intersects with branches, and where branches contact each other. The PointNeXt-based segmentation algorithm designed in this study is more suitable for our phenotypic trait collection due to its excellent performance and requiring fewer computing resources.

### 3.3. Apple tree phenotypic trait extraction results

Comparisons of the different methods in estimating the volume of apple trees revealed that the volume measured by the convex hull method was significantly larger than the other two methods (Fig. 13a, 13b), making it more suitable for predicting light interception for each

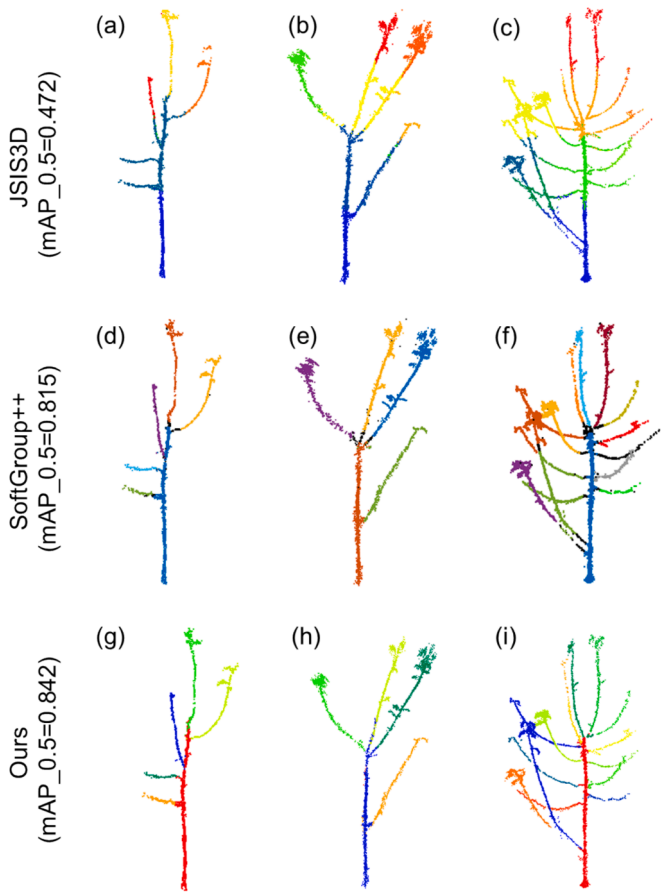



**Fig. 12.** Instance segmentation visualization. (a), (b) and (c) are the segmentation results of JSIS3D, (d), (e) and (f) are the segmentation results of SoftGroup++, where the black points indicate segmentation errors. (g), (h), and (i) are our methods. Each organ was assigned a different color.

plant. The larger the volume based on the convex hull, the more the branches tended to grow outward, or the longer the branches, the more likely the leaves were to be exposed to sunlight. The convex hull volume is more conducive to measuring the light utilization efficiency of plants, while the voxel- and concave hull-based methods estimate the wood content more effectively. Certainly, there are also cases that deviate from the overall trend. In the volume calculation based on the concave hull, Sample 2 has a smaller volume than Sample 4, whereas the other two methods exhibit the opposite trend. This discrepancy is attributed to variations in tree structure. For instance, the branches of Sample 2 grow upwards with a smaller angle to the main stem, and Sample 4 retains a greater number of leaves. The canopy width was directly obtained from the point cloud, and the different sizes of canopies reflected the growth status of different plants (Fig. 13c).

By conducting a comprehensive evaluation of the branch length, number of branches, trunk height, trunk diameter, and tree height, it is demonstrated that there is a high degree of correlation between the predicted values and manual measurements (Fig. 14, Fig. 15). The analysis also indicates a certain disparity between manually measured values and the predicted values for trunk height and branch length. This discrepancy can be attributed to errors introduced during the entire segmentation process. Upon comparison, it was found that the R2 values for branch length and quantity obtained in this study are relatively higher than Zhang's paper (Zhang et al., 2020), indicating that the method proposed in this research has certain advantages for our apple tree data. Conversely, tree height exhibits a strong correlation, as it is directly derived from the collected point cloud, thus avoiding the introduction of segmentation errors. The inaccuracies in trunk diameter primarily stem from the precision of point cloud acquisition. When utilizing Kinect V2 outdoors, susceptibility to lighting variations may lead to point cloud data loss, impacting the accuracy of trunk diameter measurements. These findings highlight the nuanced nature of the segmentation process and the impact of environmental factors, such as lighting conditions, on the accuracy of point cloud data in outdoor settings. Among the segmentation-related phenotypes, the segmentation results of PTv2, SoftGroup++ and PointNeXt were used to extract apple tree phenotypic traits and achieved comparable results.

## 4. Discussion

### 4.1. Comparison with the newly released semantic segmentation model

This study developed semantic segmentation of point clouds using PointNeXt combined with post-processing operations. We compared it with the state-of-the-art transformer-based methods. 3D segmentation is a rapidly evolving research field with many new semantic segmentation models emerging constantly. PTv2 is the latest model we chose to compare with our segmentation results (Wu et al., 2022). During network training, to accommodate the point cloud density of apple trees, we set the grid size for point cloud processing to three multi-scales of [0.02, 0.04, 0.06] meters for feature extraction. The threshold has a significant impact on the segmentation results of PTv2, and during model training, it requires manual parameter tuning based on point cloud density to achieve the best segmentation results. Compared to our proposed method, the test metrics obtained by PTv2 did not show significant differences Furthermore, models based on Transformers require significant computational resources, inevitably leading to higher experimental costs (Han et al., 2023).

### 4.2. Comparison with 3D instance segmentation models

#### 4.2.1. Comparison with 3D skeleton extraction methods

Point cloud skeletons were used for segmentation in crops such as maize and sorghum (Miao et al., 2021; Xiang et al., 2019), and this study also attempted direct instance segmentation of apple trees using this method. We tested Miao et al's method on our dataset and presented some instance segmentation results (Fig. 16). From the test results, we found that the major drawback of this method on the apple dataset is its inability to segment the trunk or produce a complete trunk segmentation accurately. In the results shown in Fig. 16d, it can be seen that the trunk was segmented into two parts: the lower part was incorrectly segmented as branches, while the upper part incorrectly grouped the trunk and branches into one category. Therefore, we introduced a deep learning point cloud semantic segmentation model to initially segment the trunk and branches.

#### 4.2.2. Comparison with 3D instance segmentation methods

An end-to-end instance segmentation model for segmenting apple tree point clouds would be more practical. Although many point cloud instance segmentation models currently exist, their generalizability to apple tree point cloud segmentation is not strong. In the early stages of our research, we chose the JSIS3D model for instance segmentation, but this model failed to segment branches and trunks. The bandwidth for the mean shift in JSIS3D was set to 1 m to achieve the best test results. As shown in the visualization results in Fig. 12 (a, b, c), this model does not work on apple tree point clouds.

With the development of 3D instance segmentation technology, we tested a new model, SoftGroup++. This model showed significant improvement over JSIS3D on apple tree point clouds, but the visualization results in Fig. 12 (d, e, f) also illustrate some unsatisfactory segmentation. Some points were marked in black, indicating segmentation errors, including incorrect segmentation of branches, and some points at the intersections between trunks and branches. The SoftGroup++ model still cannot achieve perfect segmentation. Compared to our proposed

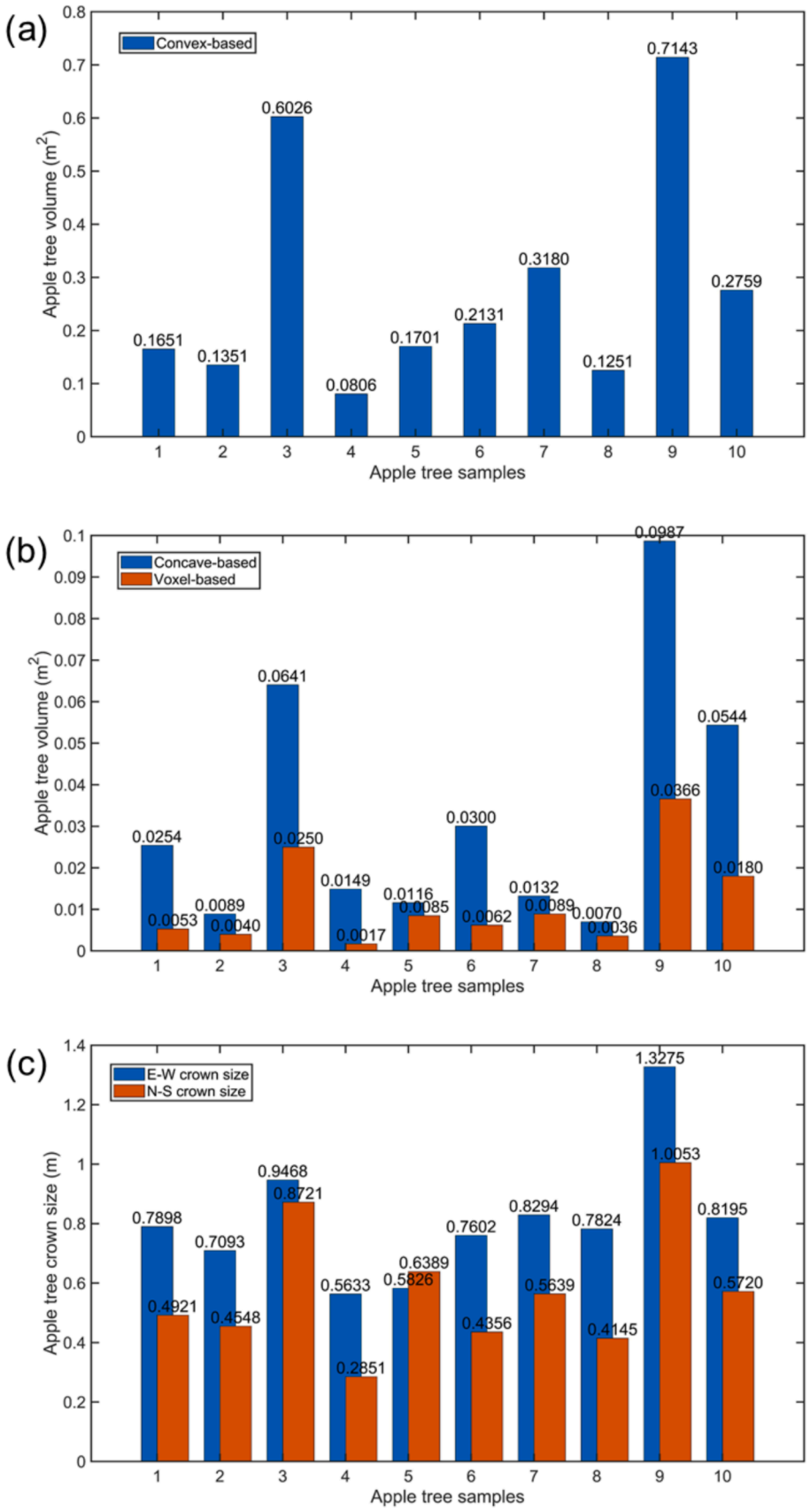



**Fig. 13.** Apple tree volume and crown size assessment. (a) Volume estimation of apple trees based on convex hull method. (b) Volume estimation of apple trees based on concave and voxel methods. (c) Crown size estimated from point cloud. E-W represents the crown size in the east–west direction. N-S represents the crown size in the north–south direction.

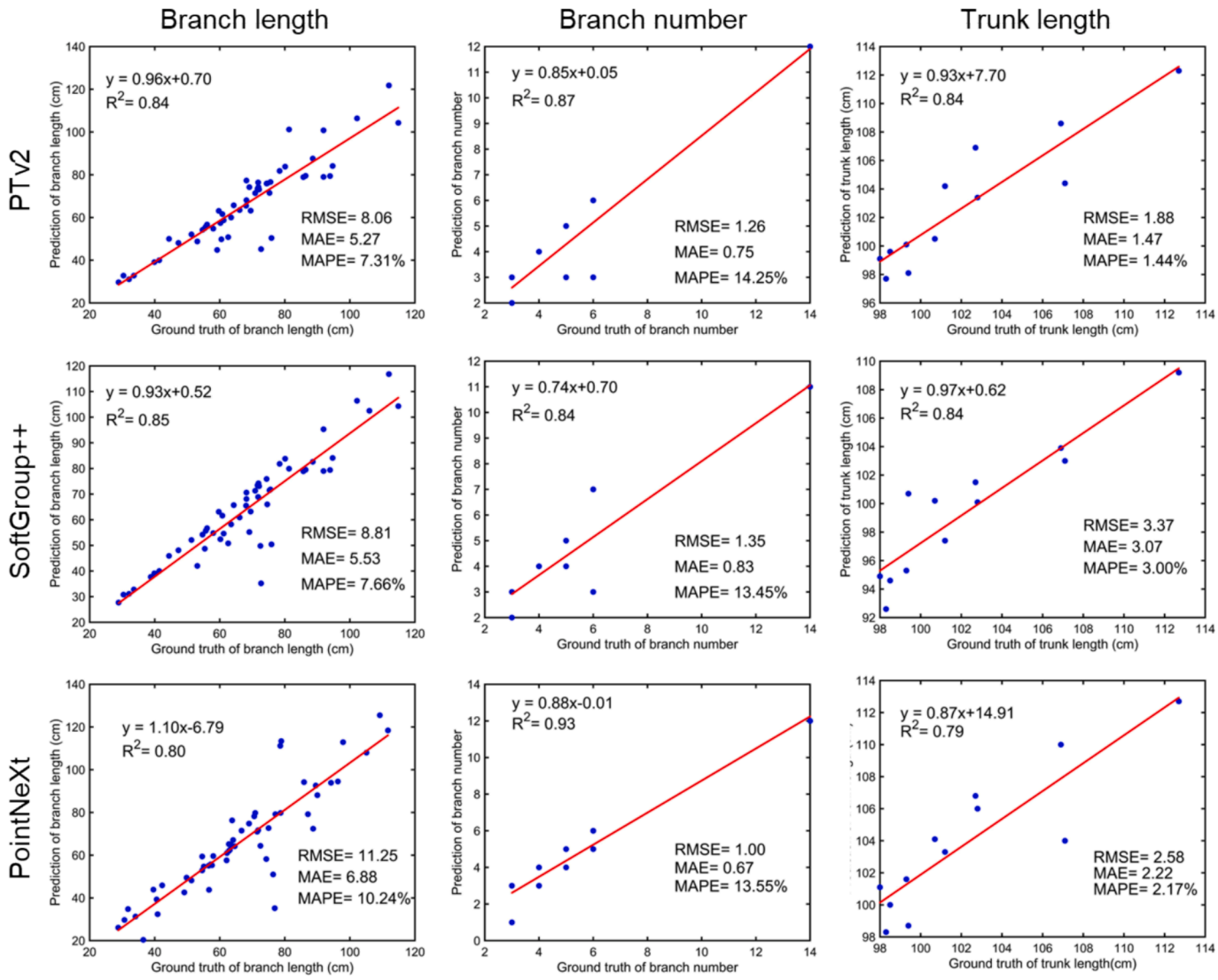



**Fig. 14.** Linear regression analysis of apple tree architectural traits. Each row shows the results from each model while each column shows the results from each trait.

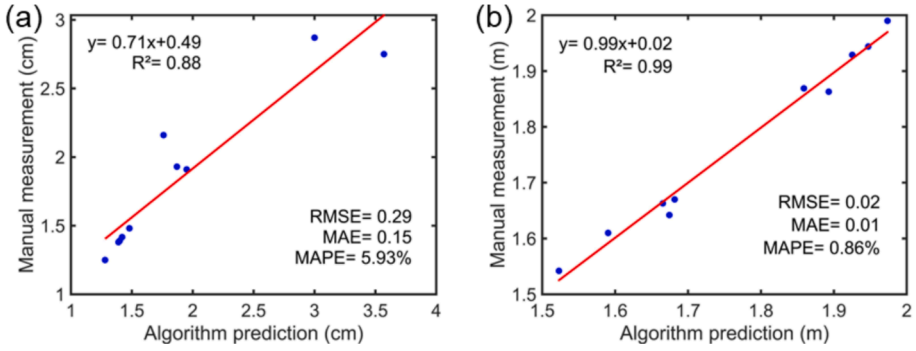



**Fig. 15.** Linear regression analysis of trunk diameter and tree height. (a) Trunk diameter. (b) tree height.

method (Fig. 12 (g, h, i)), its advantage lies in being an end-to-end model, but its segmentation accuracy lags behind our method.

SoftGroup++ shows three notable segmentation errors when applied to apple tree point clouds. One major issue occurs at points where different categories come into contact. Due to the highly similar features in these areas, achieving completely accurate segmentation is very challenging. Another error is that different branches were segmented into the same group (Fig. 17a). This occurs because the point cloud offset prediction module fails to move points of different apple tree branches toward their respective instance centroids. Instead, due to limited learning capacity, the points of different branches were shifted toward the same centroid, leading to the clustering of different branches into one group. The third error is that a single branch is segmented into multiple instances (Fig. 17b). This error is also related to the offset

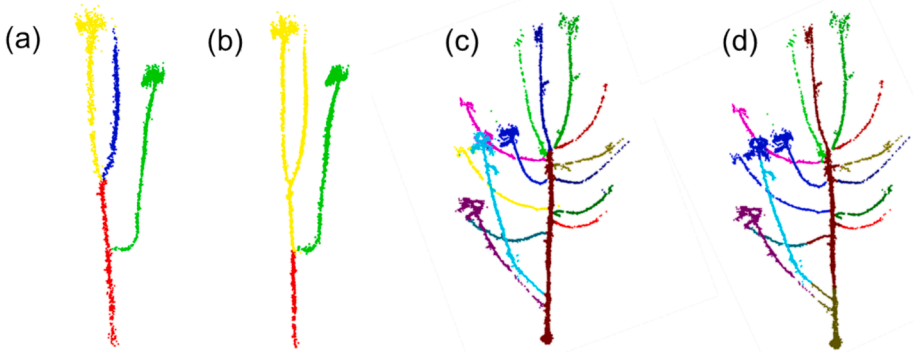



**Fig. 16.** Instance segmentation of apple tree point cloud based on skeleton extraction. (a) and (c) are instance annotations, and (b) and (d) are instance prediction results. Different colors represent different instances.

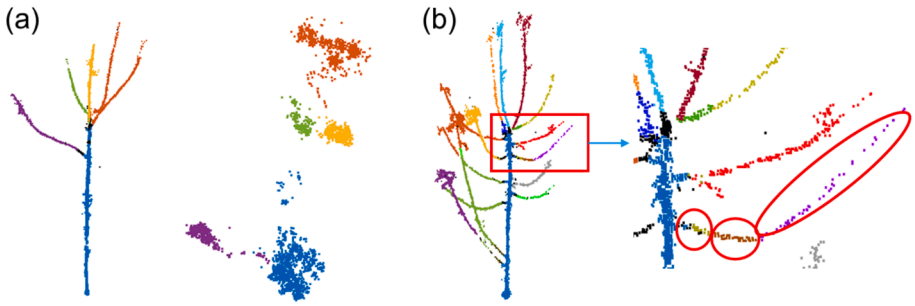



**Fig. 17.** SoftGroup++ instance segmentation error analysis. (a) Instance segmentation results and shifted coordinates. The color of the shifted coordinates corresponds to the color of the apple tree organ instance. (b) A single branch was segmented into multiple instances.

prediction module. When the shifted coordinates of a single branch are not together, clustering tends to segment them into different instances.

The point cloud segmentation method designed in this study is an important tool for quantitative phenotypic analysis and helps to gain a deeper understanding of apple tree growth patterns. The ability to measure physical attributes directly from the segmented point cloud components is of significant promise for advancing phenomics research and enhancing cultivar development strategies, offering a scalable solution for automated plant analysis (Zhao et al., 2019). On the other hand, it is essential to acknowledge certain limitations. The method may face challenges when branches overlap or come into contact with each other, making the differentiation of individual organs more complex in such situations. Additionally, errors may occur at points where branches and the trunk come into contact. Nevertheless, our point cloud instance segmentation method demonstrates notable success in apple tree organ segmentation, offering valuable insights for agricultural applications.

### 4.3. Advantages, limitations, and prospects

This study explored the method of phenotypic acquisition of apple tree point clouds in the field with Kinect V2. The fast and inexpensive acquisition of apple tree point cloud is one of the advantages of Kinect V2. Despite the availability of more cost-effective methods for acquiring point cloud data, such as using any camera device to capture the target of interest and utilizing some 3D reconstruction software like COLMAP (Schonberger and Frahm, 2016), the quality of the reconstructed point cloud is related to the quantity of captured images. Furthermore, the reconstruction speed significantly decreases as the number of images increases. Our method of acquiring point clouds achieves a balance between cost and speed.

However, data collection requires the collaboration of two individuals – one handling the mobile device and adjusting angles to capture a complete image of an apple tree, while the other operates a laptop to facilitate the capture. Due to human fatigue, extended working hours can impact the quality of data collection. The development of an autonomous on-site data collection system, similar to Husky (Rodriguez-Sanchez et al., 2022) and MARS (Xu and Li, 2022), is necessary, as it can reduce manual operations.

The quality of collected point clouds significantly impacts subsequent segmentation. Outdoor lighting conditions can distort depth information within the point cloud, particularly on objects featuring intricate surface details. Employing a shield on the data collection system to shield against sunlight can alleviate these challenges (Jiang et al., 2020). Furthermore, capturing data from diverse angles and positions, and subsequently registering this data together, helps mitigate occlusion and enhances overall data quality (Ma et al., 2021). In this study, individual plants are manually segmented from point clouds, which limits practicality. A future direction involves directly segmenting individual plants from point clouds collected in the field and testing on larger-scale datasets. This study analyzed the segmentation results of the point cloud instance segmentation model SoftGroup++ on apple trees. However, a robust end-to-end instance segmentation model would be more practical. Future work will focus on researching point cloud instance segmentation models.

## 5. Conclusions

This study successfully integrated a 3D deep learning model PointNeXt and Laplacian-based 3D skeleton extraction techniques to achieve organ-level instance segmentation of apple trees from point clouds acquired with a low-cost Kinect V2. Our customized algorithms extracted architectural traits from the entire tree as well as from the organs. Our study found that low-cost depth sensors could be used for rapid data collection and phenotypic trait extraction of apple trees, but the accuracy of segmentation could be enhanced with higher-quality point cloud data. In addition, cross-contact between tree branches is a key challenge for segmentation. Future work will focus on automating the data collection process, obtaining high-quality point clouds through multi-

view photogrammetry methods, and improving the accuracy of the segmentation algorithm on overlapping branches. This work contributes to providing quantitative apple tree architectural traits to improve orchard management and breeding research.

## CRediT authorship contribution statement

**Lizhi Jiang:** Writing - original draft, Software, Methodology, Investigation, Data curation. **Changying Li:** Writing - review & editing, Supervision, Methodology. **Longsheng Fu:** Writing - review & editing, Supervision, Resources and funding, Methodology.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgments

Authors LJ and LF acknowledge the funding support from National Natural Science Foundation of China (32171897). The authors sincerely appreciates the valuable contributions made by all participants in the research on apple tree phenotype extraction.

## Data availability

Data will be made available on request.

## References

An, N., Welch, S.M., Markelz, R.J.C., Baker, R.L., Palmer, C.M., Ta, J., Maloof, J.N., Weinig, C., 2017. Quantifying time-series of leaf morphology using 2D and 3D photogrammetry methods for high-throughput plant phenotyping. Comput. Electron. Agric. 135, 222–232. https://doi.org/10.1016/j.compag.2017.02.001.

Bao, Y., Tang, L., Srinivasan, S., Schnable, P.S., 2019. Field-based architectural traits characterisation of maize plant using time-of-flight 3D imaging. Biosyst. Eng. 178, 86–101. https://doi.org/10.1016/j.biosystemseng.2018.11.005.

Béland, M., Baldocchi, D.D., 2021. Vertical structure heterogeneity in broadleaf forests: Effects on light interception and canopy photosynthesis. Agric. For. Meteorol. 307, 108525. https://doi.org/10.1016/j.agrformet.2021.108525.

Bohn, T., Bouayed, J., 2020. Apples: an apple a day, still keeping the doctor away?', in: Jaiswal, A.K.B.T.-N.C. and A.P. of F. and V. (Ed.), . Academic Press, pp. 595–612. doi: 10.1016/B978-0-12-812780-3.00037-4.

Boogaard, F.P., van Henten, E.J., Kootstra, G., 2021. Boosting plant-part segmentation of cucumber plants by enriching incomplete 3D point clouds with spectral data. Biosyst. Eng. 211, 167–182. https://doi.org/10.1016/j.biosystemseng.2021.09.004.

Cao, J., Tagliasacchiy, A., Olsony, M., Zhangy, H., Su, Z., 2010. Point cloud skeletons via Laplacian-based contraction. In: SMI 2010 - International Conference on Shape Modeling and Applications, Proceedings. https://doi.org/10.1109/SMI.2010.25.

Disney, M.I., Boni Vicari, M., Burt, A., Calders, K., Lewis, S.L., Raumonen, P., Wilkes, P., 2018. Weighing trees with lasers: Advances, challenges and opportunities. Interface Focus 8. https://doi.org/10.1098/rsfs.2017.0048.

FAO, 2020. FAO statistical databases. doi: http://www. fao.org.

Guo, Q., Wu, F., Pang, S., Zhao, X., Chen, L., Liu, J., Xue, B., Xu, G., Li, L., Jing, H., Chu, C., 2018. Crop 3D—a LiDAR based platform for 3D high-throughput crop phenotyping. Sci. China Life Sci. 61, 328–339. https://doi.org/10.1007/s11427-017-9056-0.

Han, K., Wang, Y., Chen, H., Chen, X., Guo, J., Liu, Z., Tang, Y., Xiao, A., Xu, C., Xu, Y., Yang, Z., Zhang, Y., Tao, D., 2023. A Survey on Vision Transformer. IEEE Trans. Pattern Anal. Mach. Intell. 45, 87–110. https://doi.org/10.1109/TPAMI.2022.3152247.

Hu, Q., Ang, B., Xie, L., Rosa, S., Guo, Y., Wang, Z., Trigoni, N., Markham, A., 2020. RandLA-Net: Efficient Semantic Segmentation of Large-Scale Point Clouds. doi: 10.1109/CVPR42600.2020.01112.

Jiang, Y., Li, C., Xu, R., Sun, S., Robertson, J.S., Paterson, A.H., 2020. DeepFlower: a deep learning-based approach to characterize flowering patterns of cotton plants in the field. Plant Methods 16, 1–17. https://doi.org/10.1186/s13007-020-00698-y.

Jiang, L., Li, C., Fu, L., 2022. 3D Deep Learning-based Segmentation to Reveal the Spatial Distribution of Cotton Bolls. ASABE Annual International Meeting 2022, 1–10. https://doi.org/10.13031/aim.202200361.

Jiang, L., Li, C., Sun, J., Chee, P., Fu, L., 2024. Estimation of Cotton Boll Number and Main Stem Length Based on 3D. In: Gaussian Splatting Written for Presentation at the 2024 ASABE Annual International Meeting Sponsored by ASABE, pp. 1–11. https://doi.org/10.13031/aim.202400898.

Ji, W., Qian, Z., Xu, B., Tao, Y., Zhao, D., Ding, S., 2016. Apple tree branch segmentation from images with small gray-level difference for agricultural harvesting robot. Optik 127, 11173–11182. https://doi.org/10.1016/j.ijleo.2016.09.044.

Jurjević, L., Liang, X., Gašparović, M., Balenović, I., 2020. Is field-measured tree height as reliable as believed – Part II, A comparison study of tree height estimates from conventional field measurement and low-cost close-range remote sensing in a deciduous forest. ISPRS J. Photogramm. Remote Sens. 169, 227–241. https://doi.org/10.1016/j.isprsjprs.2020.09.014.

Krause, S., Sanders, T.G.M., Mund, J.P., Greve, K., 2019. UAV-based photogrammetric tree height measurement for intensive forest monitoring. Remote Sens. (Basel) 11, 1–18. https://doi.org/10.3390/rs11070758.

Lau, A., Bentley, L.P., Martius, C., Shenkin, A., Bartholomeus, H., Raumonen, P., Malhi, Y., Jackson, T., Herold, M., 2018. Quantifying branch architecture of tropical trees using terrestrial LiDAR and 3D modelling. Trees - Structure and Function 32, 1219–1231. https://doi.org/10.1007/s00468-018-1704-1.

Li, L., Zhang, Q., Huang, D., 2014. A review of imaging techniques for plant phenotyping. Sensors (switzerland). 11, 20078-20111. doi:10.3390/s141120078.

Ma, B., Du, J., Wang, L., Jiang, H., Zhou, M., 2021. Automatic branch detection of jujube trees based on 3D reconstruction for dormant pruning using the deep learning-based method. Comput. Electron. Agric. 190, 106484. https://doi.org/10.1016/j.compag.2021.106484.

Majeed, Y., Zhang, J., Zhang, X., Fu, L., Karkee, M., Zhang, Q., Whiting, M.D., 2018. Apple tree trunk and branch segmentation for automatic trellis training using convolutional neural network based semantic segmentation. IFAC-PapersOnLine 51, 75–80. https://doi.org/10.1016/j.ifacol.2018.08.064.

Mao, W., Murengami, B., Jiang, H., Li, R., He, L., Fu, L., 2024. UAV-based high-throughput phenotyping to segment individual apple tree row based on geometrical features of poles and colored point cloud. J. ASABE 67 (5), 1231–1240. https://doi.org/10.13031/ja.15895.

Miao, T., Zhu, C., Xu, T., Yang, T., Li, N., Zhou, Y., Deng, H., 2021. Automatic stem-leaf segmentation of maize shoots using three-dimensional point cloud. Comput. Electron. Agric. 187. https://doi.org/10.1016/j.compag.2021.106310.

Mu, X., He, L., Heinemann, P., Schupp, J., Karkee, M., 2023. Mask R-CNN based apple flower detection and king flower identification for precision pollination. Smart Agric. Technol. 4, 100151. https://doi.org/10.1016/j.atech.2022.100151.

Ni, X., Li, C., Jiang, H., Takeda, F., 2021. Three-dimensional photogrammetry with deep learning instance segmentation to extract berry fruit harvestability traits. ISPRS J. Photogramm. Remote Sens. 171, 297–309. https://doi.org/10.1016/j.isprsjprs.2020.11.010.

Pham, Q.H., Nguyen, T., Hua, B.S., Roig, G., Yeung, S.K., 2019. JSIS3D: Joint semantic-instance segmentation of 3D point clouds with multi-task pointwise networks and multi-value conditional random fields. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. https://doi.org/10.1109/CVPR.2019.00903.

Qian, G., Li, Y., Peng, H., Mai, J., Hammoud, A.K., H.A., Elhoseiny, M., Ghanem, B., 2022. PointNeXt: revisiting PointNet++ with improved training and scaling strategies. Adv. Neural Inf. Proces. Syst. 35, 1–18.

Qi, C.R., Su, H., Mo, K., Guibas, L.J., 2017. PointNet: Deep learning on point sets for 3D classification and segmentation. In: Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition. https://doi.org/10.1109/CVPR.2017.16.

Qi, C.R., Yi, L., Su, H., Guibas, L.J., 2017b. PointNet++: Deep hierarchical feature learning on point sets in a metric space. Advances in Neural Information Processing Systems 2017-Decem, 5100–5109. doi: doi: 10.48550/arXiv.1706.02413.

Qiu, T., Cheng, L., Jiang, Y., 2022. 3D characterization of tree architecture for apple crop load estimation. ASABE Annual International Meeting 2022, 1–10. https://doi.org/10.13031/aim.202201119.

Rodriguez-Sanchez, J., Li, C., 2022. An autonomous ground system for 3D LiDAR-based crop scouting written for presentation at the 2022 ASABE Annual International Meeting Sponsored by ASABE 1–10.

Saeed, F., Sun, S., Sanchez, J.R., Snider, J., Liu, T., Li, C., 2023. Cotton plant part 3D segmentation and architectural trait extraction using point voxel convolutional neural networks. Plant Methods 19, 1–23. https://doi.org/10.1186/s13007-023-00996-1.

Saleem, M.H., Potgieter, J., Arif, K.M., 2021. Automation in Agriculture by Machine and Deep Learning Techniques: A Review of Recent Developments, Precision Agriculture. Springer US. doi: 10.1007/s11119-021-09806-x.

Sapkota, R., Ahmed, D., Karkee, M., 2023. Comparing YOLOv8 and Mask RCNN for object segmentation in complex orchard environments. arXiv preprint arXiv: 2312.07935. doi: 10.48550/arXiv.2312.07935.

Schonberger, J.L., Frahm, J.M., 2016. Structure-from-motion revisited. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2016-Decem, 4104–4113. https://doi.org/10.1109/CVPR.2016.445.

Shi, W., van de Zedde, R., Jiang, H., Kootstra, G., 2019. Plant-part segmentation using deep learning and multi-view vision. Biosyst. Eng. 187, 81–95. https://doi.org/10.1016/j.biosystemseng.2019.08.014.

Sun, S., Li, C., Chee, P.W., Paterson, A.H., Jiang, Y., Xu, R., Robertson, J.S., Adhikari, J., Shehzad, T., 2020. Three-dimensional photogrammetric mapping of cotton bolls in situ based on point cloud segmentation and clustering. ISPRS J. Photogramm. Remote Sens. 160, 195–207. https://doi.org/10.1016/j.isprsjprs.2019.12.011.

Sun, X., Fang, W., Gao, C., Fu, L., Majeed, Y., Liu, X., Gao, F., Yang, R., Li, R., 2022. Remote estimation of grafted apple tree trunk diameter in modern orchard with RGB and point cloud based on SOLOv2. Comput. Electron. Agric. 199, 107209. https://doi.org/10.1016/j.compag.2022.107209.

Sun, X., He, L., Jiang, H., Li, R., Mao, W., Zhang, D., Majeed, Y., Andriyanov, N., Soloviev, V., Fu, L., 2024. Morphological estimation of primary branch length of individual apple trees during the deciduous period in modern orchard based on

PointNet++. Comput. Electron. Agric. 220, 108873. https://doi.org/10.1016/j.compag.2024.108873.

Tan, C., Li, C., He, D., Song, H., 2023. Anchor-free deep convolutional neural network for tracking and counting cotton seedlings and flowers. Comput. Electron. Agric. 215, 108359. https://doi.org/10.1016/j.compag.2023.108359.

Thomas, H., Qi, C.R., Deschaud, J.E., Marcotegui, B., Goulette, F., Guibas, L., 2019. KPConv: Flexible and deformable convolution for point clouds. In: Proceedings of the IEEE International Conference on Computer Vision 2019-October, 6410–6419. https://doi.org/10.1109/ICCV.2019.00651.

Velasco, R., Zharkikh, A., Affourtit, J., Dhingra, A., Cestaro, A., Kalyanaraman, A., Fontana, P., Bhatnagar, S.K., Troggio, M., Pruss, D., 2010. The genome of the domesticated apple (Malus× domestica Borkh.). Nat. Genet. 42, 833–839. https://doi.org/10.1038/ng.654.

Vu, T., Kim, K., Nguyen, T., Luu, T.M., Kim, J., Yoo, C.D., 2024. Scalable SoftGroup for 3D instance segmentation on point clouds. IEEE Trans. Pattern Anal. Mach. Intell. 46, 1981–1995. https://doi.org/10.1109/TPAMI.2023.3326189.

Wallace, L., Hillman, S., Reinke, K., Hally, B., 2017. Non-destructive estimation of above-ground surface and near-surface biomass using 3D terrestrial remote sensing techniques. Methods Ecol. Evol. 8, 1607–1616. https://doi.org/10.1111/2041-210X.12759.

Wang, Y., Li, W., Xu, X., Qiu, C., Wu, T., Wei, Q., Ma, F., Han, Z., 2019. Progress of apple rootstock breeding and its use. Hortic. Plant J. 5, 183–191. https://doi.org/10.1016/j.hpj.2019.06.001.

Wu, X., Lao, Y., Jiang, L., Liu, X., Zhao, H., 2022. Point transformer V2: grouped vector attention and partition-based pooling. Adv. Neural Inf. Proces. Syst. 35, 1–16.

Xiang, L., Bao, Y., Tang, L., Ortiz, D., Salas-Fernandez, M.G., 2019. Automated morphological traits extraction for sorghum plants via 3D point cloud data analysis. Comput. Electron. Agric. 162, 951–961. https://doi.org/10.1016/j.compag.2019.05.043.

Xu, R., Li, C., 2022. A modular agricultural robotic system (MARS) for precision farming: Concept and implementation. J. Field Rob. 39, 387–409. https://doi.org/10.1002/rob.22056.

Xu, C., Huang, S., Tian, B., Ren, J., Meng, Q., Wang, P., 2017. Manipulating planting density and nitrogen fertilizer application to improve yield and reduce environmental impact in Chinese Maize production. Front. Plant Sci. 8, 1–11. https://doi.org/10.3389/fpls.2017.01234.

Yu, X., Tang, L., Rao, Y., Huang, T., Zhou, J., Lu, J., 2022. Point-bert: Pre-training 3d point cloud transformers with masked point modeling, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19313–19322. doi: 10.48550/arXiv.2111.14819.

Zhang, J., He, L., Karkee, M., Zhang, Q., Zhang, X., Gao, Z., 2018. Branch detection for apple trees trained in fruiting wall architecture using depth features and Regions-Convolutional Neural Network (R-CNN). Comput. Electron. Agric. 155, 386–393. https://doi.org/10.1016/j.compag.2018.10.029.

Zhang, C., Yang, G., Jiang, Y., Xu, B., Li, X., Zhu, Y., Lei, L., Chen, R., Dong, Z., Yang, H., 2020a. Apple tree branch information extraction from terrestrial laser scanning and backpack-LiDAR. Remote Sens. (Basel) 12, 3592. https://doi.org/10.3390/rs12213592.

Zhang, J., Karkee, M., Zhang, Q., Zhang, X., Yaqoob, M., Fu, L., Wang, S., 2020b. Multi-class object detection using faster R-CNN and estimation of shaking locations for automated shake-and-catch apple harvesting. Comput. Electron. Agric. 173, 105384. https://doi.org/10.1016/j.compag.2020.105384.

Zhang, C., Zhang, K., Ge, L., Zou, K., Wang, S., Zhang, J., Li, W., 2021. A method for organs classification and fruit counting on pomegranate trees based on multi-features fusion and support vector machine by 3D point cloud. Sci. Hortic. 278, 109791. https://doi.org/10.1016/j.scienta.2020.109791.

Zhang, Y., Wu, J., Yang, H., Zhang, C., Tang, Y., 2024. A hierarchical growth method for extracting 3D phenotypic trait of apple tree branch in edge computing. Wirel. Netw 30, 5951–5966. https://doi.org/10.1007/s11276-023-03385-7.

Zhao, C., Zhang, Y., Du, J., Guo, X., Wen, W., Gu, S., Wang, J., Fan, J., 2019. Crop phenomics: Current status and perspectives. Front. Plant Sci. 10, 714. https://doi.org/10.3389/fpls.2019.00714.

Zhao, H., Jiang, L., Jia, J., Torr, P., Koltun, V., 2020. Point Transformer. In: Proceedings of the IEEE International Conference on Computer Vision 16239–16248. https://doi.org/10.48550/arxiv.2012.09164.

Zhao, H., Jiang, L., Jia, J., Torr, P., Koltun, V., 2021. Point Transformer. IEEE/CVF International Conference on Computer Vision (ICCV) 2021, 16239–16248. https://doi.org/10.1109/ICCV48922.2021.01595.