Computers and Electronics in Agriculture 224 (2024) 109132

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

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

Original papers

PointResNet: A grape bunches point cloud semantic segmentation model based on feature enhancement and improved PointNet++

Jiangtao Luo [a], Dongbo Zhang [a,*], Lufeng Luo [b,*], Tao Yi [a]

[a] School of Automation and Electronic Information, Xiangtan University, Xiangtan 411105, China
[b] School of Mechatronic Engineering and Automation, Foshan University, Foshan 528000, China

ARTICLE INFO

Keywords:
Feature enhancement
3D semantic segmentation
Grape segmentation
Point cloud

ABSTRACT

As a type of representative bunch-type fruit, the collision-free and undamaged harvesting of grapes is of great significance. To obtain accurate 3D spatial semantic information, this paper proposes a method for multi-feature enhanced semantic segmentation model based on Mask R-CNN and PointResNet(improved PointNet++). Firstly, a depth camera is used to obtain RGBD images. The RGB images are then inputted into the Mask-RCNN network for fast detection of grape bunches. The color and depth information are fused and transformed into point cloud data, followed by the estimation of normal vectors. Finally, the nine-dimensional point cloud, which include spatial location, color information, and surface structure information, are inputted into the PointResNet network to achieve semantic segmentation of grape bunches, peduncles, and leaves. This process obtains the extraction of spatial semantic information from the surrounding area of the bunches. The experimental results show that by incorporating normal vector and color features, the overall accuracy of point cloud segmentation increases to 96.5%, with a mean accuracy of 90.3%. This represents a significant improvement of 7.9% and 16.6% compared to using only positional features. The results demonstrate that the model method presented in this paper can effectively provide precise 3D semantic information to the robot while ensuring both speed and accuracy. This lays the groundwork for subsequent collision-free and undamaged picking.

1. Introduction

In recent years, there has been a growing trend to replace manual labor with intelligent systems in various stages of crop cultivation, management, harvesting, and processing. This is due to high labor intensity, poor working conditions, and a significant increase in labor costs. For instance, some studies have utilized RGBD sensors for orchard monitoring and crop yield estimation (Moreno et al., 2020; Kurtser et al., 2020; Wang et al., 2018). Fruit harvesting is also a crucial area in this transition. Currently, various robotic systems for fruit harvesting have been reported (Zhou et al., 2022; Yan et al., 2021; Lin et al., 2021; Williams et al., 2020; Cao et al., 2019; Shamshiri et al., 2018). Among these, there has been a significant amount of research on the harvesting of individual fruits such as strawberries, apples, and bell peppers. Due to the notable differences in appearance, shape, and skin characteristics among different fruits, the harvesting methods and technologies for each fruit require specific research and exploration. The focus of these studies lies in the detection methods and harvesting techniques.

Kang et al. (2020) utilized the lightweight model Mobile-DasNet to detect apples, and then employed PointNet (Qi et al., 2017a) to predict suitable grasping and harvesting poses. The metric IoU3D for their pose estimation reaches 0.88, but the study does not provide on-site actual grasping experimental results. Brown and Sukkarieh (2021) designed a soft four-finger picking mechanism, which achieves non-destructive picking of apples through flexible grasping, but did not provide test results on success rate. Ning et al. (2022) proposed an enhanced YOLO-V4-CBAM network model, building upon YOLO-V4 (Bochkovskiy et al., 2020), for extracting picking points of bell peppers. The novelty of this work lies in its designed bell pepper picking planning algorithm, which effectively addresses collision issues during picking. Lehnert et al. (2017) and Sa et al. (2017) utilized color and geometric information to describe the target, and then employed an SVM classifier to detect bell peppers. Finally, they achieved picking by first adhering to the surface of the pepper and then shearing the stem. Their robot designed for harvesting bell peppers in a protected environment demonstrated a 90% success rate in adhesion and a 58% success rate in harvesting. This method is validated in orchard experiments but requires further improvement in stem detection to reduce the problem of damaging bell peppers due to improper stem cutting.

The fruits of apples or bell peppers typically come in a single form and can be easily harvested by grasping. Additionally, fruits such as

* Corresponding authors.
E-mail addresses: 202121623013@smail.xtu.edu.cn (J. Luo), zhdb@xtu.edu.cn (D. Zhang), luolufeng617@163.com (L. Luo).

https://doi.org/10.1016/j.compag.2024.109132
Received 8 January 2024; Received in revised form 9 April 2024; Accepted 3 June 2024
Available online 12 June 2024

apples and bell peppers are inherently easy to identify, and they do not easily come loose when touched. Therefore, the relative difficulty of harvesting such fruits is considered low. Grapes, in contrast, are distributed in bunches, making them a type of bunch fruit. They have large bunches, small peduncles, thin peels, and are prone to dropping easily. Therefore, when harvesting grapes, the focus is on cutting the peduncle. However, the peduncle is relatively small and can be easily hidden, which undoubtedly increases the difficulty of detecting the cutting point and ensuring undamaged picking.

As a typical representative of bunch fruits, grape harvesting has received considerable attention in recent years. Pérez-Zavala et al. (2018) conducted extensive research by employing histograms of oriented gradients (HOG) and local binary patterns (LBP) to extract shape and texture features of grapes. Subsequently, they utilized algorithms such as Support Vector Machine (SVM) for classification and Density-Based Spatial Clustering of Applications with Noise (DBSCAN) for clustering to detect grape bunches. The average precision and recall rates achieved were 88.61% and 80.34%, respectively. Their work partially overcomes the influence of lighting conditions on grape recognition. Luo et al. (2016) constructed a robust classifier based on the AdaBoost framework. They employed multiple color spaces to extract features and determine whether each pixel belonged to a grape bunch, thereby achieving image segmentation and bunch detection. Before the widespread application of deep learning, their use of traditional methods was relatively successful in accurately identifying grapes in complex environments at that time. However in recent years, as deep learning has matured and become widely adopted, an increasing number of researchers have used deep neural network models to detect grape bunches. Sozzi et al. (2022) and Santos et al. (2020) both validated the performance of the YOLO (Redmon et al., 2016) model in detecting grape bunches in experiments. The difference lies in the fact that the former tested multiple versions of the YOLO algorithm (v3, v4, and v5) and provided detection accuracy and speed metrics, while the latter conducted a training comparison using YOLOv3, YOLOv4, and Mask R-CNN (He et al., 2017) models. Ning et al. (2021) proposed a deep learning-based method for grape peduncle recognition and optimal harvest point localization, achieving an average detection accuracy of 88% and a precision of 99.43% in optimal harvest point localization. Despite this, detection and localization solely on two-dimensional images still provide insufficient three-dimensional pose information for robots to cut the peduncles. Luo et al. (2022) used Mask-RCNN for grape bunch detection and segmentation. They constructed a region of interest for the fruit peduncle to identify the optimal cutting point. The LOWESS algorithm was employed to predict the pose of the fruit peduncle. They achieved a standard deviation of approximately 17 degrees in peduncle pose estimation angle without occlusion. Nevertheless, the limitation lies in the inability of a single 6D pose to represent peduncles with significant curvature, which are widely present. Romeo et al. (2023) utilized depth values to assist deep learning models in grape semantic segmentation. Experimental results demonstrate a significant improvement in both the accuracy and intersection over union (IoU) of grape segmentation using this method.

Most of the above researches carry out the detection of grape bunches or peduncles on the collected two-dimensional images. However, harvesting robots must operate in three-dimensional space. This is especially important for vineyards with complex environments and grape harvesting robots that require precise operation. Obtaining accurate three-dimensional spatial semantic information is crucial for guiding the harvesting mechanism to realize collision-free and undamaged operation.

In order to accurately obtain the spatial semantic information around the grape bunch, this paper proposes a 3D semantic segmentation model based on feature enhancement for point cloud of grape bunches. Firstly, we employ Mask R-CNN to extract the region of interest(ROI) of grape bunches and the corresponding color information. Then, based on the improved point cloud segmentation model PointResNet, semantic segmentation is achieved by fusing features such as color, position, and normal vectors. This process aims to obtain the spatial positions of grapes, peduncles, and leaves, laying the groundwork for subsequent collision-free and undamaged harvesting by robotic systems. The contributions of this paper are:

(1) A model method is proposed for 3D semantic segmentation of the surrounding area of grape bunches using RGBD camera data, which combining 2D detection and 3D semantic segmentation, comprehensively utilizing surface appearance and spatial structure information. This method aims to lay a solid foundation for obtaining accurate 3D spatial semantic information.

(2) The PointResNet model and ResMLP module are proposed, which enhance the learning capability of the model by increasing the depth of PointNet++ and improving the MLP perception module. This results in an increase of **11.6%** and **5.2%** in mAcc and mIoU for peduncle detection, respectively.

(3) In this study, we collected 105 manually labeled 2D image datasets and 149 manually labeled point cloud datasets. These datasets will be made publicly available on GitHub, accompanied by an open-source link, thereby facilitating further research in this area. The datasets is available at https://github.com/Ljt-XTU/grape.

2. Materials and methods

2.1. Image acquisition and dataset processing

As illustrated in Fig. 1, to facilitate data collection and model algorithm debugging, we constructed a simulated vineyard environment in the laboratory. This environment consists of grapes, grapevines, and support structures. A depth camera is mounted at the end of a 6-DoF robotic arm, allowing for real-time adjustments to the camera's pose by controlling the arm's movements. This capability enables the camera to capture images from various perspectives. The depth camera used in the experiment is the Intel RealSense D435i, which captures RGBD images directly from the current perspective. These RGBD images consist of pixel values in the RGB color space, along with corresponding depth values.

We sequentially suspended the grape model at different positions on the "grape trellis", adjusting the robotic arm's posture to position the camera at various angles, heights, and distances toward the grape model. A total of 105 RGBD images with a resolution of 1280 × 720 were captured during this process. Next, we utilized the Labelme annotation software to mark the positions of grape and peduncle in the images. Following the common practice of an 8:2 ratio, we divided the dataset into a training set and a test set. Specifically, 84 images were selected for the training set, while the remaining 21 images were designated for the test set.

2.2. 3D segmentation model under feature enhancement

This paper proposes a 3D semantic segmentation model by combining Mask R-CNN and an enhanced PointNet++ model with feature enhancement. To streamline computations and avoid unnecessary processing in non-task areas, we focus exclusively on refining 3D semantic segmentation within local regions containing grape bunches, bypassing the need for computations across the entire image space. Fig. 2 illustrates the three components of the entire model framework: ROI extraction, point cloud features enhancement, point cloud semantic segmentation. The following sections provide detailed explanations for each of these components.

Fig. 1. Simulated experimental environment.

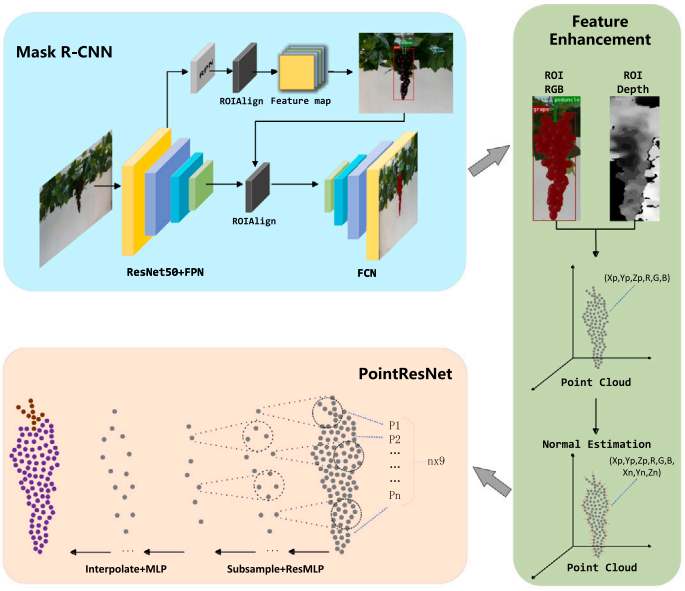

Fig. 2. Structure diagram of grape point cloud segmentation system.

2.2.1. ROI extraction

We locate the region of interest (ROI) around the grape bunch. The reason for doing this is that first, the grape-picking task requires real-time performance. If the subsequent 3D semantic segmentation is trained and inferred in the entire image space, it will take a lot of time. Second, after detecting the fruit bunch, the robot arm only needs to complete the picking in a small area around the fruit bunch. Therefore, extracting the ROI can significantly reduce the computational cost of subsequent 3D semantic segmentation while ensuring accurate picking operations and speeding up the robot's picking operation speed.

$$\begin{cases} X_{roi} = X_g \\ Y_{roi} = Y_p \\ H_{roi} = Y_g + H_g - Y_p \\ W_{roi} = W_g \end{cases} \tag{1}$$

Object detection is the most direct and effective approach for achieving Region of Interest (ROI) extraction. We chose the Mask R-CNN (He et al., 2017) network model, which combines ResNet50 (He et al., 2016) and FPN (Lin et al., 2017) for feature extraction, to detect grape bunches and peduncles. In the upper-left region as illustrated in Fig. 2, the image captured by the camera is input to the model on the left. After performing regression and classification in the object detection branch, as well as up-sampling and decoding in the semantic segmentation branch, the final output consists of the boundary box and segmentation results for the grape bunch and grape peduncle.

The position and size of the rectangular detection box and ROI are represented in the form of (X, Y, H, W), where (X, Y) denotes the coordinates of the upper-left corner of the rectangular detection box in the image, and (H, W) represent the height and width of the rectangular box, as shown in Fig. 3. The subscript p indicates the detection box for the peduncle, and the subscript g indicates the detection box for the grape bunch. The position and size of ROI on the image are defined by Eq. (1).

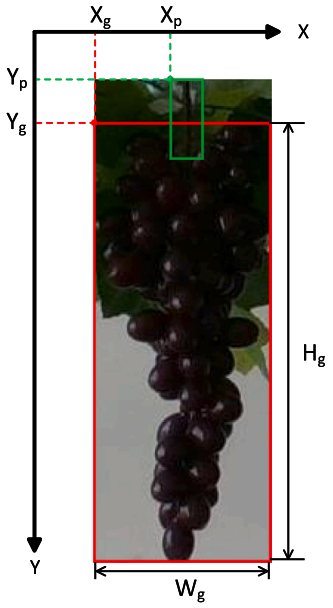

Fig. 3. ROI schematic diagram. The image ROI obtained according to the definition of Eq. (1). The red rectangle is the bounding box of the grape bunch, and the green is the bounding box of the peduncle.

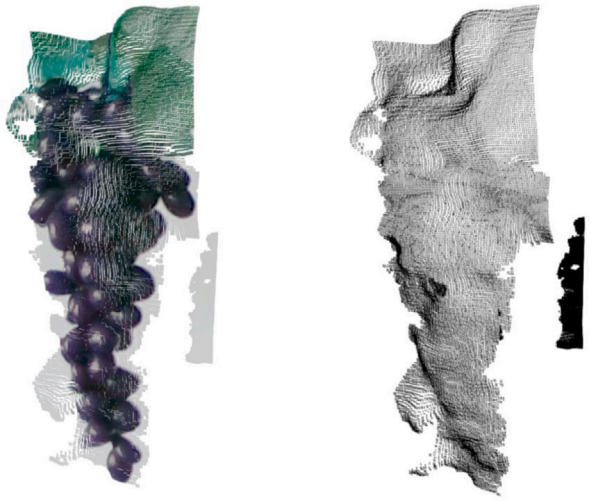

Fig. 4. Comparison of point clouds with or without color features.

2.2.2. Point cloud feature enhancement

The perception of spatial information in a scene depends on the representation of 3D data. Point cloud data retains the original geometric structure information in 3D space (Guo et al., 2020), so it is more suitable for tasks related to scene understanding, such as the perception problem of grape bunches and their surrounding areas that this study needs to deal with.

Due to the fact that point cloud data inherently only contain spatial positional features, it is challenging to provide sufficient information to the network model for discerning the category of each point. For the task of this study, if color information can be appended to the point cloud features, it becomes easier to distinguish between grape, peduncles, and leaves. Fig. 4 provides an example of point cloud data for a grape bunch. The left image represents the point cloud with appended color information, while the right image represents the original point cloud.

As shown in Fig. 2, the RGB image undergoes Region of Interest (ROI) extraction to obtain the region containing the grape bunch. We cut out the corresponding region on the depth image, mapping pixel points in the image coordinate system to point cloud coordinates in the camera coordinate system. The three-channel RGB information is utilized to endow the point cloud with color features. Let the coordinates of a pixel point P in the image coordinate system be (U, V), and the corresponding 3D spatial coordinates of the point cloud be (X, Y, Z). The transformation from 2D pixel coordinates to 3D point cloud coordinates is illustrated in Fig. 5.

In the pixel coordinate system, the origin is at the top left corner of the image, and the coordinates represent discrete pixel positions (U, V). The image coordinate system has its origin at the center of the image, with coordinates measured in meters. Based on the intrinsic parameters of the camera, the coordinates of pixel point P are translated and scaled to the image coordinate system. Then, according to the pinhole camera model, point P in the image coordinate system is further transformed into the camera coordinate system.

The transformation process is illustrated in Algorithm 1. We utilize matrix multiplication to calculate the coordinates of the entire image, significantly speeding up computation to meet real-time requirements. Lines 3 and 4 of the algorithm generate matrices for the horizontal and vertical coordinates of the image. In line 6, based on imaging principles, the X and Y coordinates of each point on the image are computed, then concatenated with the Z-axis coordinates and flattened to obtain an $N \times 3$ point cloud data. Finally, lines 10–11 filter out invalid points

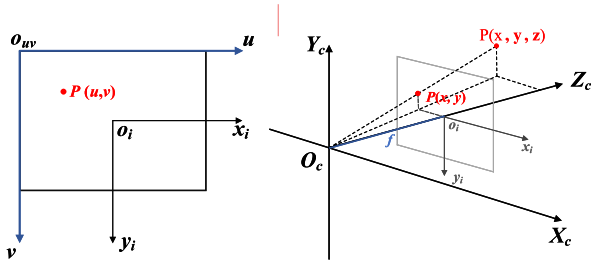

Fig. 5. Schematic diagram of coordinate transformation. On the left is an illustration of the transformation from pixel coordinates to image coordinates, and on the right is an illustration of the transformation from image coordinates to camera coordinates.

Algorithm 1 Generation of point cloud with additional color features in ROI

Input: $color$ ▷ RGB image of ROI
$depth$ ▷ depth image of ROI
(X, Y, H, W) ▷ defined area of ROI
(f_x, f_y, c_x, c_y) ▷ internal parameters
$depth\_scale$ ▷ depth scale
Output: $point\_cloud$ ▷ point cloud with color features

1: $vec\_v \leftarrow \begin{bmatrix} Y & Y+1 & \dots & Y+H-1 \end{bmatrix}^{\mathrm{T}}$
2: $vec\_u \leftarrow \begin{bmatrix} X & X+1 & \dots & X+W-1 \end{bmatrix}^{\mathrm{T}}$
3: $V \leftarrow (v_{ij})_{H\times W} \leftarrow \begin{bmatrix} vec\_v & vec\_v & \dots & vec\_v \end{bmatrix}$
4: $U \leftarrow (u_{ij})_{H\times W} \leftarrow \begin{bmatrix} vec\_u & vec\_u & \dots & vec\_u \end{bmatrix}^{\mathrm{T}}$
5: $Z \leftarrow depth/depth\_scale$
6: $X \leftarrow (U - c_x) \times Z/f_x$, $Y \leftarrow (V - c_y) \times Z/f_y$
7: $P \leftarrow dstack(X, Y, Z).reshape(-1, 3)$
8: $C \leftarrow dstack(colors/255).reshape(-1, 3)$
9: **for** $i \leftarrow 0, n$ **do**
10: **if** $P[i, 2] < 3.0$ and $P[i, 2] > 0.001$ **then**
11: $point\_cloud[i] \leftarrow \begin{bmatrix} P[i] & C[i] \end{bmatrix}$
12: **end if**
13: **end for**

based on distance while appending RGB values, resulting in an $N \times 6$ point cloud with color information.

The normal vector represents the orientation of the local surface and contains information about the surface changes of the object. This is useful for describing the object's appearance. Therefore, we also consider adding it to the point cloud data. To this end, after obtaining the point cloud of the grape bunch and its surrounding area, we also need to estimate the normal vector of each point.

We use mainstream normal vector estimation algorithms to extract this feature. The algorithm needs to infer the normal vector by fitting the local surface around the selected point (Du et al., 2023). In order to fit the local plane around the selected point p_s in the point cloud, we use the KD-Tree construction algorithm of the K-Nearest Neighbor(KNN) algorithm to select K points in the neighborhood of point p_s. Here, we analyze the number and density of local point clouds, and choose $K = 30$ as the number of neighborhood points. We then employ the least squares method to fit these local points into a plane (Fig. 6), specifically to determine a plane P_l that minimizes the distance between the K neighboring points and the plane:

$$\min_{m,n,\|n\|=1} \sum_{i=1}^{K} \left((p_i - m)^T n \right)^2 \tag{2}$$

$$m = \frac{1}{K} \sum_{i=1}^{K} p_i \tag{3}$$

where vector n represents the unit normal vector of plane P_l, K represents the number of selected points in the neighborhood, p_i represents

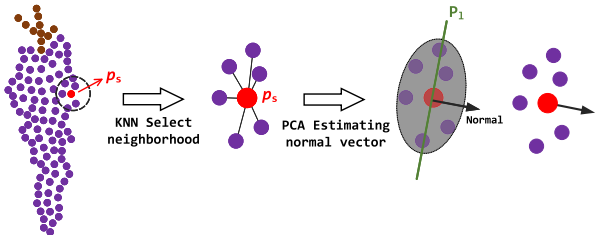

Fig. 6. The flowchart for estimating point cloud normal vectors.

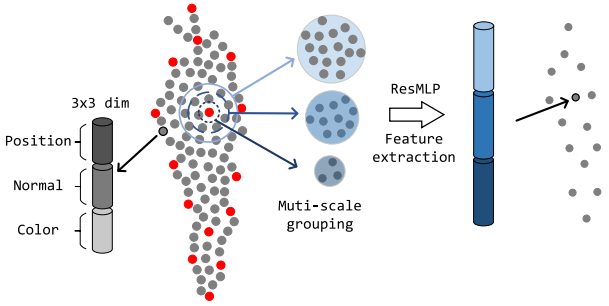

Fig. 7. Input and local feature extraction of PointResNet.

the points selected in the neighborhood of point p_s, and m can generally be regarded as the center of these N points, and then ensure that:

$$y_i = p_i - m \tag{4}$$

the optimization objective function can be transformed into:

$$\begin{aligned} &\min_n n^T S n \\ &s.t \quad n^T n = 1 \\ &\qquad S = \sum_{i=1}^{K} y_i y_i^T \end{aligned} \tag{5}$$

The matrix S is the covariance matrix of the three-dimensional coordinates of the selected points in the neighborhood. This problem can also be regarded as a solution to Principal Component Analysis(PCA). The eigenvector corresponding to the minimum eigenvalue of matrix S is the normal vector n of the selected point. Repeating the above process for each point cloud in the grape bunch region can estimate the normal vector feature of the corresponding point.

By adding color and normal vectors to the original point cloud data that only contains spatial positions, the enhanced point cloud data with nine-dimensional features is obtained.

2.2.3. Point cloud semantic segmentation

In order to further utilize the enhanced point cloud data to directly perform semantic segmentation and obtain the spatial positions of grape bunches, peduncles and leaves, we need to introduce a 3D point cloud semantic segmentation model based on deep neural networks.

We enhanced the PointNet++ (Qi et al., 2017b) model with multi-scale grouping by increasing the number of Set Abstraction (SA) layers and correspondingly improving the structure of the perception layers. This enhancement allows the model to learn better on deep networks. Additionally, we increased the input feature channels from 3 or 6 dimensions to 9 dimensions to accommodate point clouds with 9-dimensional features. We named the improved network PointResNet. As shown in Fig. 7, PointResNet first utilizes the farthest point sampling(FPS) to select key points based on the position of the point cloud. Ball query searches for all points p_i surrounded by a spherical region with the key point p_s as the center and R as the radius. We determine

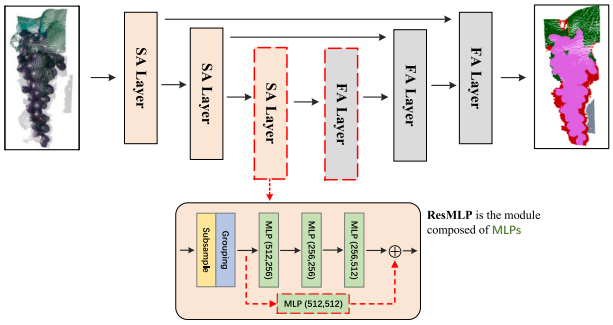

Fig. 8. Diagram of PointResNet model.

the radius based on multiple experimental optimizations. These points are considered as the neighborhood of the key point (Eq. (6))

$$U\left(p_s, R\right) = \left\{p_i \middle| \left\|p_s - p_i\right\| < R, i = 1, 2, 3, \ldots, n\right\} \tag{6}$$

Unlike the ball query with a single radius, multi-scale grouping selects different radius to query the neighborhood of key points. It then uses ResMLP to extract features for all points in each neighborhood and concatenates them to obtain the local features of the current key point. After each key point undergoes the aforementioned process, the current layer achieves downsampling and local feature extraction. As shown in Fig. 8, the modules and structures added on PointNet++ are indicated by red dashed lines. Taking into account the model size of current deep learning techniques, we increased the original two layers of Set Abstraction (SA) in PointNet++ to three layers, enhancing the perception capability of PointResNet. However, experimental results indicate that the average loss on the test set is much higher than that on the training set. To address the issue of enhanced learning capability but decreased performance of the model, we applied the residual structure of ResNet to each MLP in the SA layer, naming it ResMLP. Such a perception module enables the model to learn better solutions when stacked with deeper layers.

In the segmentation process of the PointResNet, we aim to predict the category for each point in the original point cloud and then learn based on the ground truth. Therefore, it is necessary to upsample the downsampled points with high-dimensional features, change their feature dimensions, and ultimately obtain a point cloud with dimensions $N \times C$, where N is the number of points in the original point cloud, and C is the number of segmentation categories. This process is similar to the UNet structure, where each Feature Aggregation(FA) layer performs upsampling to aggregate high-dimensional features from the upper layer and extracts features obtained during downsampling.

After training, the PointResNet model takes in 9-dimensional grape point cloud data containing position, normal vector, and color information. Features are extracted using ResMLP in the downsampling layers, while the upsampling layers aggregate features using MLP. Finally, the model outputs class predictions for each point, achieving semantic segmentation of the point cloud.

3. Results and analysis

3.1. Dataset

The experimental dataset used in this study is divided into two parts: a two-dimensional object detection dataset and a point cloud segmentation dataset. The former has been introduced in Section 2.1, and this section focuses on the dataset employed for point cloud segmentation. We extracted ROI(regions of interest) containing grape bunches from 105 RGB images, resulting in 149 ROI images (some RGB images containing multiple grapes). Subsequently, as described in Section 2.2.1, we generated 149 ROI point cloud images using the

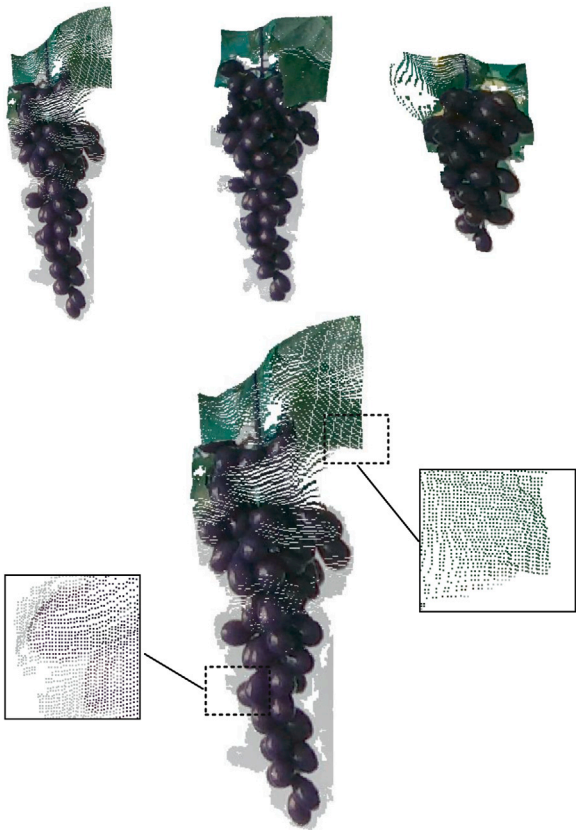

Fig. 9. Examples of grape point cloud data.

mentioned method. These images were annotated using tools to identify grapes, peduncles, leaves, background, and noise. Finally, the point cloud data were organized in ShapeNet (Chang et al., 2015) format. The dataset was split into 118 training images and 31 testing images in an 8:2 ratio. Fig. 9 illustrates several examples of point cloud images. We define noise as points in the point cloud that do not belong to grape clusters, leaves, peduncles, or background, such as the white-colored points surrounding the grape clusters shown in Fig. 9.

3.2. Evaluation metrics

The experiment employs two commonly used evaluation metrics in the 3D semantic segmentation field, namely accuracy(Acc) and intersection over union (IoU), to assess the semantic extraction performance of the proposed method on grapes and their surrounding areas. Let n_{ij} represent the number of samples where the model classifies points belonging to class i as class j, and C denote the total number of classes. The overall accuracy(OAcc), class accuracies(Acc), and mean class accuracy(mAcc) are defined as follows:

$$OAcc = \frac{\sum_{i=1}^{C} n_{ii}}{\sum_{i=1}^{C}\sum_{j=1}^{C} n_{ij}} \tag{7}$$

$$Acc_i = \frac{n_{ii}}{\sum_{j=1}^{C} n_{ij}}, i = 1, 2, \ldots, C \tag{8}$$

$$mAcc = \frac{1}{C}\sum_{i=1}^{C} Acc_i \tag{9}$$

In addition, the computation of class intersection over union (IoU) and mean class intersection over union(mIoU) metrics is as follows:

$$IoU_i = \frac{n_{ii}}{\sum_{j=1}^{C} n_{ij} + \sum_{j=1}^{C} n_{ji} - n_{ii}}, i = 1, 2, \ldots, C \tag{10}$$

$$mIoU = \frac{1}{C}\sum_{i=1}^{C} IoU_i \tag{11}$$

3.3. Experimental and segmentation results

We initially utilized 84 annotated RGB grape images out of a total of 105 for training the Mask R-CNN. We employ two techniques to ensure optimal training of the model on our collected dataset. Firstly, we augment the training process based on two datasets: MSCOCO (Lin et al., 2014) and OnlyGrape (Ariel522, 2023). We utilize the weights pretrained on MSCOCO to further pre-train on the OnlyGrape dataset, which contains 300 semantic segmentation images of grapes. The pre-trained model already possesses feature extraction capabilities, and fine-tuning enables the model to perform well on the target task under limited sample conditions (Han et al., 2021; Pan and Yang, 2009; Szegedy et al., 2015; He et al., 2016). Secondly, during the fine-tuning phase on our dataset, we use data augmentation to increase the dataset size. After extracting small batches of data, we randomly flip and crop images within each batch before feeding them into the model. Additionally, we increase the number of training iterations to enhance model performance. After training, the model was capable of detecting grape regions and segmenting grape bunches and peduncles. Subsequently, we employed this trained model to detect grapes in the entire dataset of 105 RGB images, thereby extracting Regions of Interest (ROI). If the number of detected grapes in a single image is greater than or equal to 2, and there is no overlap between these grape clusters, multiple ROIs need to be extracted. Upon extracting each ROI, we promptly utilized its corresponding RGBD image to generate a point cloud and fused its position, color, and normal vector features. For overlapping grape bunches, we merge the overlapping regions of interest (ROIs) to obtain a larger rectangular ROI. Subsequently, we follow the same steps to generate point clouds and annotate them. Before training the point cloud semantic segmentation model, we normalized the point cloud images of grape bunches and sampled 15 000 points to input into the PointResNet model. During feature extraction, we adopted a multi-scale sampling approach to capture local information as comprehensively as possible.

3.3.1. Comparative experiment

The mainstream detection of grapes method uses a 2D semantic segmentation model to segment the grape and peduncle in the RGBD image. During the picking process, the position (u, v, d) of the target on the RGBD image is mapped to the coordinate value (x, y, z) in the world coordinate system. This mapping is used to guide the end effector to reach the target position. In this context, (u, v, d) represents the coordinates of the target on the image and its corresponding depth value.

To compare with the traditional 2D segmentation method, we used Mask R-CNN to convert the 2D segmentation results of the ROI in the grape scene into point clouds. We then compared these results with the point cloud segmentation results of the multi-feature enhancement model and single-feature model in the same area. The results are shown in Table 1.

Comparing the first and last rows in the table, all metrics of our method, PointResNet, are higher than those of the mainstream solution Mask R-CNN. The segmentation accuracy of grape and peduncles reaches 98.2% and 62.4%, respectively, representing improvements of 1.6% and 1.4%. In terms of the three overall metrics, the overall accuracy and mean class accuracy have increased by 33.0% and 9.4%, respectively, while the mean class intersection over union has improved by 14.4%. To specifically compare the impact of feature enhancement, we apply the feature enhancement method to several well-known 3D segmentation networks such as PointNet++, RS-CNN (Liu et al., 2019),

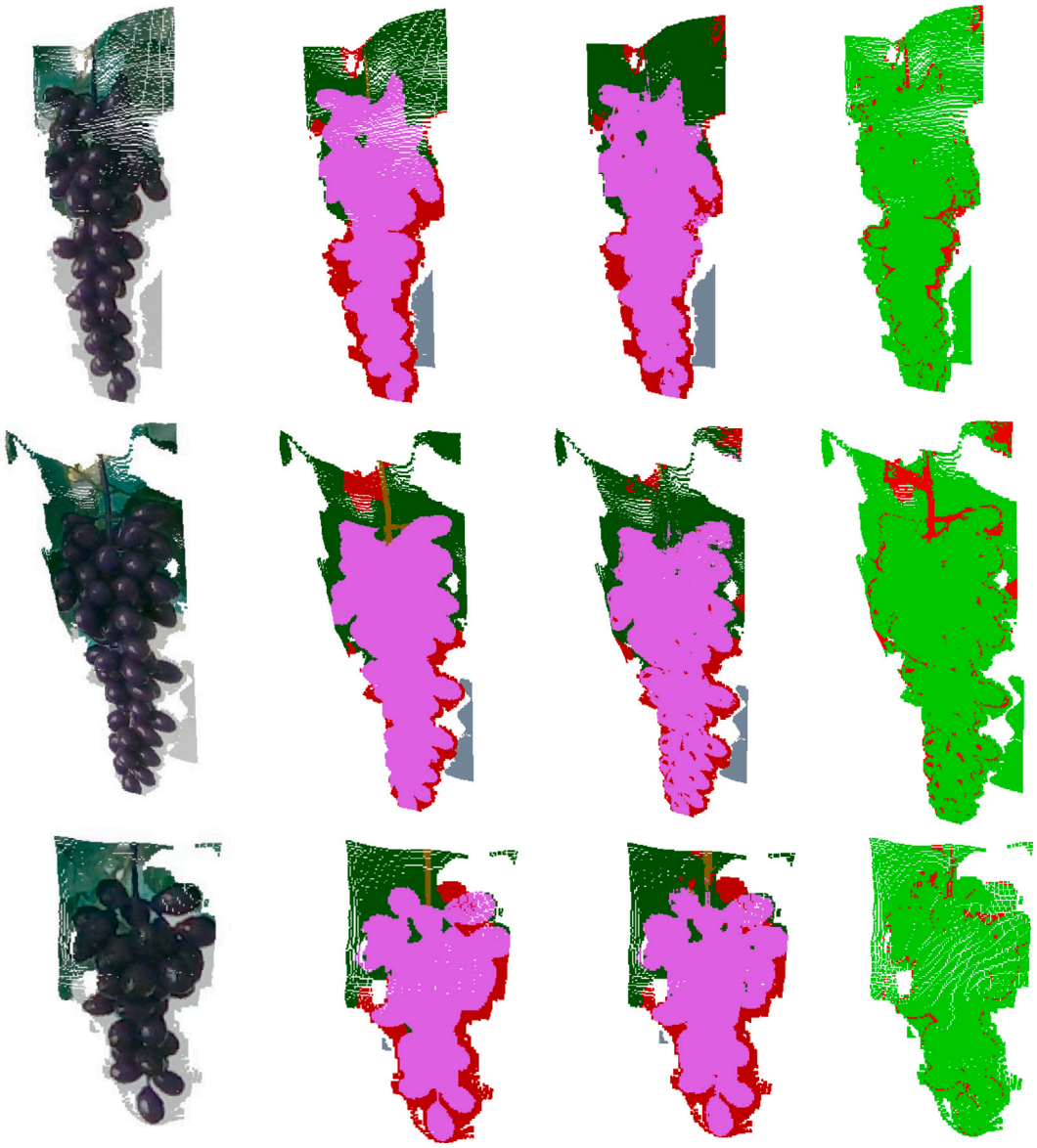

Fig. 10. Examples of segmentation results. The first column displays the physical image of the grape bunch point cloud, the second column shows the ground truth of the sample, and the third column presents the model's prediction result for the sample. Green indicates leaves, purple indicates grapes, brown indicates peduncles, gray indicates the background, and red indicates noises. The fourth column compares the model's correct and incorrect predictions. Green represents correct predictions, while red represents incorrect predictions.

Table 1
Comparison of segmentation results for grape data across various methods.

Method	Grape (Class Acc)	Peduncle	Leaf	Background	Noise	OAcc	mAcc	mIoU
Mask R-CNN (He et al., 2017)	0.966	0.610	–	–	–	0.635	0.808	0.698
RS-CNN (Liu et al., 2019)	0.922	0.210	0.793	0.985	0.650	0.886	0.760	0.676
PPNet (Liu et al., 2020)	0.919	0.088	0.788	0.994	0.662	0.886	0.742	0.667
KPConv (Thomas et al., 2019)	0.930	0.053	0.774	0.941	0.558	0.873	0.709	0.638
PointNet++ (Qi et al., 2017b)	0.941	0.032	0.782	0.975	0.880	0.715	0.650	0.650
Nine-dim RS-CNN	**0.986**	0.464	0.910	0.990	0.888	0.964	0.873	0.824
Nine-dim PPNet	0.956	0.310	0.842	0.983	0.760	0.922	0.802	0.738
Nine-dim KPConv	0.958	0.120	0.854	0.990	0.800	0.931	0.790	0.730
Nine-dim PointNet++	0.982	0.603	0.912	**0.991**	0.877	0.963	0.895	0.831
PointResNet	0.982	**0.624**	**0.930**	0.978	**0.901**	**0.965**	**0.902**	**0.852**

PPNet (Liu et al., 2020), and KPConv (Thomas et al., 2019), and comparative experiments were conducted. Table 1 demonstrates that the performance of these models surpasses that of the mainstream Mask R-CNN solution in terms of overall metrics such as OAcc, mAcc, and mIoU. The final segmentation results of our model are illustrated in Fig. 10. Visually, the segmentation performance of the model is commendable, with the majority of points being accurately classified. The primary misclassifications are concentrated around the peduncles and the boundaries between different categories. The discrimination of grape peduncles in the point cloud is significantly dependent on the

Table 2
Comparison of 2D semantic segmentation results across different models.

Method	Grape (Class Acc/Class IoU)	Peduncle	OAcc	mAcc	mIoU
DeeplabV3 (Chen et al., 2017)	0.936/0.855	0.511/0.305	0.611	0.728	0.574
MobileNetV3 (Howard et al., 2019)	0.943/0.876	0.429/0.274	0.618	0.689	0.572
FCN (Long et al., 2015)	0.933/0.857	0.449/0.263	0.610	0.695	0.561
PSPNet++ (Zhao et al., 2017)	0.934/0.860	0.400/0.228	0.610	0.672	0.547
Mask R-CNN	**0.966/0.883**	**0.610/0.402**	**0.635**	**0.808**	**0.698**

precision of the sensor and the density of the point cloud. For structures like grape peduncles, which are slender and irregular, depth cameras face challenges in accurately capturing their depth at long distances, resulting in considerable errors in the point cloud. Additionally, the influence of downsampling leads to a loss of information in the peduncle region. Consequently, the detection of peduncles is less accurate compared to other categories, leading to lower accuracy. Subsequent experiments in real environments with higher-quality point cloud data have shown better experimental results and higher accuracy.

In order to compare the performance of other 2D segmentation models and Mask R-CNN on our dataset, we experimented with several commonly used 2D semantic models, including DeeplabV3 (Chen et al., 2017), MobileNetV3 (Howard et al., 2019), FCN (Long et al., 2015), and PSPNet (Zhao et al., 2017). Among them, the first two are more commonly used in the field of robotic harvesting along with Mask R-CNN (Kang et al., 2020; Luo et al., 2022; Romeo et al., 2023). From the second column of the Table 2, it can be seen that the segmentation accuracy and intersection over union (IoU) of Mask R-CNN for grape are both higher than those of the second-ranked MobileNetV3, with an improvement of 3% and 2.8% respectively. Meanwhile, the third column of the table shows that the detection metrics of Mask R-CNN for grape peduncles far exceed those of other models by around 10% to 20%. This is because the dataset images have a resolution of 1280 × 720, where even in samples with the largest proportion of peduncles, they only account for approximately 0.5% of the entire image. When optimizing, the other four models focus more on correctly segmenting grape and background. In contrast, Mask R-CNN segments grape bunches and peduncles within the detection boxes after object detection, significantly increasing the proportion of peduncles. Therefore, Mask R-CNN can segment peduncles in the images, and the corresponding overall metrics are also higher than those of other models.

3.3.2. Ablation experiment

To evaluate the impact of feature enhancement and model improvements on 3D semantic segmentation models, we conducted ablation experiments. The baseline model utilized only positional features. Subsequently, experiments were conducted by incorporating normal vector features, color features, and both normal vector and color features. Table 3 present the results of PointResNet under four different conditions on the test set. With the increase in feature dimensions, the accuracy(Acc) and intersection over union(IoU) show a generally upward trend. Compared to the model using only positional features, the semantic segmentation model with feature enhancement shows improvements in overall accuracy by 7.9%, mean class accuracy by 16.6%, and mean class intersection over union by 17.5%. Table 4 displays the data of the PointNet++ model under the same experiments to compare the effects of model improvements. Comparing the highest values(highlighted in bold) in each column of Tables 3 and 4, PointResNet slightly lags behind PointNet++ in class intersection over union for grape and background. The class accuracy for noise is 6% lower. However, other metrics are higher than those of PointNet++. Notably, for peduncles, two metrics are improved by 11.6% and 5.2%, respectively, which are crucial for guiding robot harvesting. The overall accuracy, mean class accuracy, and mean intersection over union are higher by 0.2%, 1.8%, and 1.2%, respectively.

Fig. 11. The grape in a real scene.

In addition, we conducted similar ablation experiments on several other well-known point cloud segmentation models mentioned in Section 3.3.1. The experiments based on the RS-CNN model are shown in Table 5. In comparison to using only positional features, the Nine-dim RS-CNN exhibits improvements of 7.8%, 11.3%, and 14.8% in the three metrics OAcc, mAcc, and mIoU, respectively. The experiments based on the PPNet model show varying degrees of improvement across the three metrics (Table 6), with all three metrics being the highest across all four feature scenarios. Table 7 presents the experiments based on the KPConv model, and the three metrics of Nine-dim KPConv are also higher than the experimental data using only positional features. From the ablation experiments results across multiple models, it can be observed that the feature enhancement method we employed in point cloud segmentation models has a certain universality. Each model, after channel expansion, demonstrates significant improvements in segmentation accuracy on the feature-enhanced grape point clouds.

3.4. Segmentation results in real scene

To verify the generalization and effectiveness of the PointResNet model and feature enhancement technique in real-world scenarios, we applied the method described in Section 2.2 to conduct experiments on grape data collected in a real orchard environment. We filtered out the background in two steps and then added color and normal vector features. Firstly, following Algorithm 1, we filtered out distant background by depth value, removing point clouds with a depth distance exceeding 3 m. Secondly, for other background elements close to the grapes, we manually labeled and removed them according to the labels. Finally, we obtained a concise point cloud image as shown at the right of Fig. 11.

The annotation stage differs from the previous experiments. Due to the connection between the real grape peduncles and branches in the orchard, we replaced the original background label (after background removal) with branches. Eventually, we annotated five classes: grapes, peduncles, leaves, branches, and noise, with no change in the definition of noise. We organized the collected 48 point cloud data into a dataset with a ratio of 3:1. During training, we used the weights pre-trained on the previous experimental data. Additionally, considering the differences in local density and quantity of grape point clouds between the real environment and the simulation environment, we reduced the radius of the ball query stage. After training, we input the grape point

Table 3
Comparison of ablation experiment results in four different features(PointResNet).

Feature	Grape (Class Acc/Class IoU)	Peduncle	Leaf	Background	Noise	OAcc	mAcc	mIoU
Position	0.953/0.830	0.147/0.063	0.746/0.641	0.995/0.976	0.581/0.498	0.886	0.737	0.667
Position+Normal	0.940/0.842	0.040/0.006	0.805/0.676	0.994/0.943	0.627/0.536	0.900	0.734	0.667
Position+Color	**0.983**/0.952	**0.719**/**0.440**	0.912/0.852	**0.993**/**0.974**	0.882/0.819	0.962	**0.913**	0.839
Position+Normal+Color	0.982/**0.955**	0.624/0.422	**0.930**/**0.873**	0.978/0.966	**0.902**/**0.834**	**0.965**	0.903	**0.842**

Table 4
Comparison of ablation experiment results in four different features(PointNet++).

Feature	Grape (Class Acc/Class IoU)	Peduncle	Leaf	Background	Noise	OAcc	mAcc	mIoU
Position	0.941/0.823	0.032/0.000	0.782/0.651	0.975/0.950	0.880/0.476	0.715	0.650	0.650
Position+Normal	0.944/0.830	0.065/0.022	0.800/0.653	0.962/0.940	0.886/0.503	0.725	0.657	0.658
Position+Color	0.981/**0.956**	0.554/0.384	**0.924**/**0.853**	0.981/0.962	**0.962**/0.814	0.888	0.828	0.828
Position+Normal+Color	**0.982**/0.955	**0.603**/**0.388**	0.912/0.850	**0.991**/**0.975**	0.877/**0.818**	**0.963**	**0.895**	**0.830**

Table 5
Comparison of ablation experiment results in four different features(RS-CNN).

Feature	Grape (Class Acc/Class IoU)	Peduncle	Leaf	Background	Noise	OAcc	mAcc	mIoU
Position	0.922/0.832	0.210/0.081	0.793/0.655	0.985/0.961	0.650/0.527	0.886	0.760	0.676
Position+Normal	0.944/0.844	0.137/0.062	0.782/0.661	0.995/0.967	0.616/0.519	0.616	0.746	0.675
Position+Color	**0.989**/0.953	**0.485**/0.345	**0.919**/**0.855**	**0.998**/0.964	0.857/0.819	0.963	**0.875**	0.823
Position+Normal+Color	0.986/**0.957**	0.464/**0.354**	0.910/0.846	0.990/**0.966**	**0.888**/**0.822**	**0.964**	0.873	**0.824**

Table 6
Comparison of ablation experiment results in four different features(PPNet).

Feature	Grape (Class Acc/Class IoU)	Peduncle	Leaf	Background	Noise	OAcc	mAcc	mIoU
Position	0.919/0.831	0.088/0.038	0.788/0.642	0.994/0.961	0.662/0.532	0.886	0.742	0.667
Position+Normal	0.926/0.834	0.185/0.114	0.773/0.641	0.916/0.902	0.689/0.537	0.882	0.748	0.671
Position+Color	**0.960**/0.885	0.204/0.115	0.807/0.725	0.972/**0.957**	**0.772**/**0.664**	0.922	0.786	0.724
Position+Normal+Color	0.956/**0.891**	**0.308**/**0.198**	**0.842**/**0.740**	**0.983**/0.945	0.760/0.652	**0.923**	**0.808**	**0.738**

Table 7
Comparison of ablation experiment results in four different features(KPConv).

Feature	Grape (Class Acc/Class IoU)	Peduncle	Leaf	Background	Noise	OAcc	mAcc	mIoU
Position	0.930/0.816	0.053/0.018	0.774/0.623	0.941/0.903	0.558/0.468	0.873	0.709	0.638
Position+Normal	0.925/0.837	0.096/0.052	0.764/0.641	0.989/0.953	0.676/0.530	0.887	0.742	0.669
Position+Color	**0.962**/0.891	**0.137**/**0.080**	0.824/0.743	0.979/0.951	0.783/0.686	0.927	0.781	0.725
Position+Normal+Color	0.958/**0.896**	0.120/0.068	**0.854**/**0.757**	**0.989**/**0.955**	**0.798**/**0.707**	**0.931**	**0.786**	**0.730**

Table 8
Experiments of various dimension-expanded models on real grape point clouds with feature enhancement.

Method	Grape (Class Acc/Class IoU)	Peduncle	Leaf	Branch	Noise	OAcc	mAcc	mIoU
Nine-dim RS-CNN	0.997/0.997	0.910/0.831	0.994/0.973	0.824/0.754	0.546/0.501	0.972	0.830	0.784
Nine-dim PPNet	0.996/0.983	0.917/0.859	0.958/0.956	0.918/0.853	0.698/0.547	0.976	0.881	0.814
Nine-dim KPConv	0.996/0.982	0.859/0.851	0.991/0.971	0.848/0.833	0.760/0.606	0.978	0.898	0.829
Nine-dim PointNet++	0.997/0.981	0.799/0.778	0.992/0.976	0.915/0.863	0.645/0.550	0.976	0.840	0.795
PointResNet	0.995/0.977	0.994/0.746	0.856/0.809	0.996/0.870	0.688/0.603	0.976	0.937	0.796

clouds from the test set into the trained model to obtain segmentation results.

As shown in Fig. 12, the segmentation performance of our method in real-world scenarios even surpasses that in the experimental environment. Particularly noteworthy is the significantly higher accuracy of segmentation for grape peduncles compared to the simulated environment. Table 8 presents the experimental results for the four mentioned point cloud segmentation models from Section 3.3.1. After feature channel expansion and experimentation on feature-enhanced real grape data, the class accuracy (class acc) and class intersection over union (class IoU) for grape, leaf, and branch categories all exceed 90%. Moreover, these two metrics for the peduncle category are significantly higher than the experimental results in the simulated environment. Although the accuracy of PointResNet on leaves is slightly lower, the improvement in segmentation accuracy for peduncles and branches is significant. Experimental results in real-world scenarios also indicate that the segmentation results for noise categories are less than satisfactory. Unlike simulated environments, noise in real-world settings is more diverse, and the features are not as pronounced. For instance, we consider scattered points far from the entire fruit cluster as noise, and at the same time, we identify points in other categories that are distinctly inappropriate as noise. However, the point cloud quality in real-world environments is relatively high, with the quantity of noise much lower than in other categories. Consequently, across the three overall metrics,

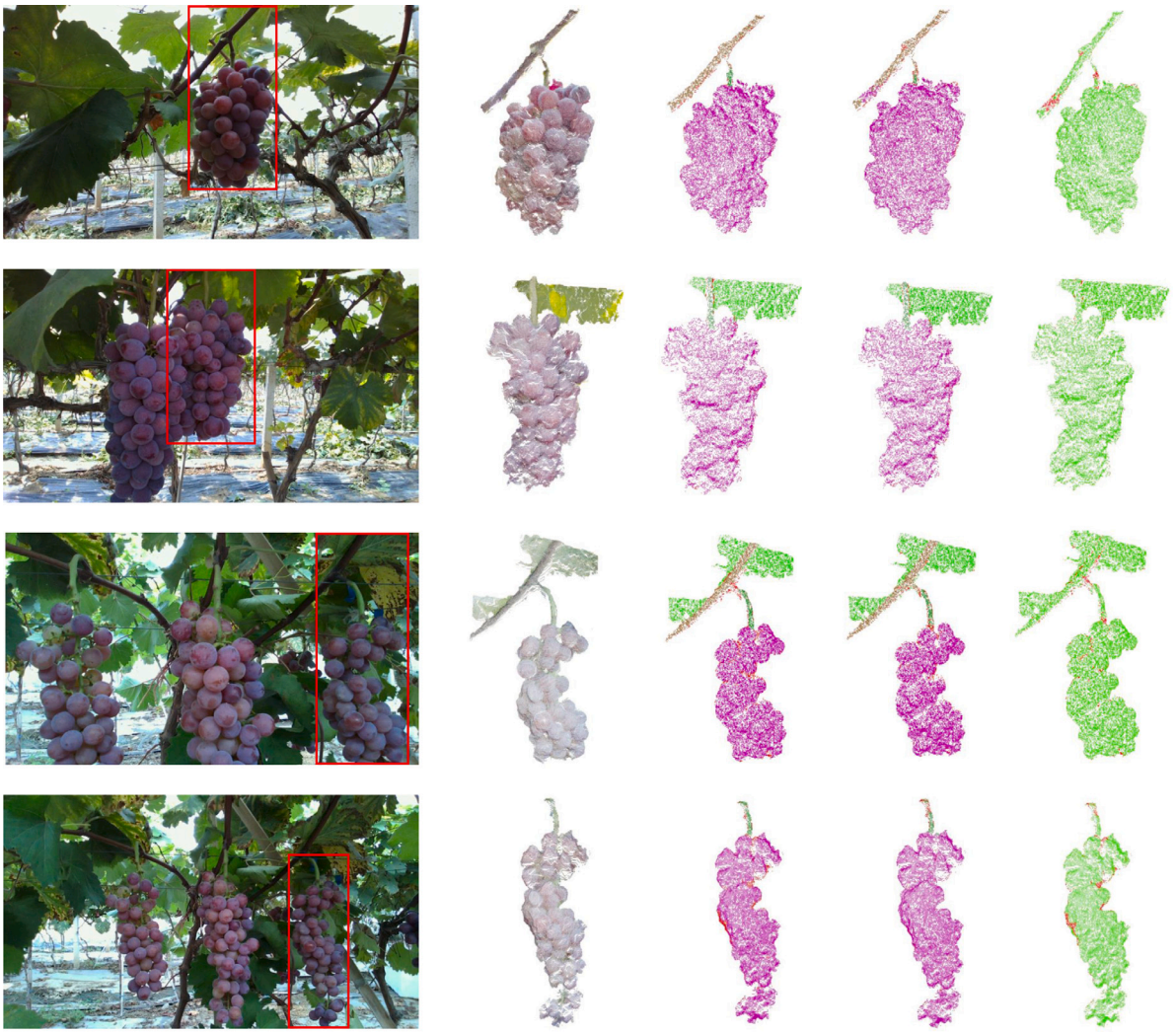

Fig. 12. Examples of segmentation results in the real scene. The left image depicts a grape scene in the orchard, while the right side consists of columns representing: the original grape point cloud image after background removal, the ground truth of the test sample, and the model's prediction for the sample. Light green indicates leaves, dark green represents grape peduncles, purple denotes grape clusters, brown signifies branches, and red marks miscellaneous points. The fourth column provides a comparison between the model's predictions and the ground truth, with green indicating correct predictions and red indicating errors.

experimental results also consistently surpass those obtained in simulated environments. The model's ability to achieve segmentation results superior to those in simulated environments is primarily attributed to the substantial improvement in the quality of point clouds collected in real-world scenarios. Additionally, the peduncles of real grapes are much thicker than those in the grape model, resulting in smaller data collection errors. This allows the model to learn and fit more effectively on data with reduced noise and errors. Secondly, the method proposed in this paper demonstrates strong learning capabilities, allowing it to achieve effective generalization on grape bunches in real-world environments, provided that the point cloud quality is guaranteed.

The experimental results are promising and valuable. However, even though data augmentation was used to expand the dataset before training, it is still challenging for the data to cover all scenarios in actual orchards. Factors such as weather, lighting conditions, and occlusions may affect the final experimental results. We will address these shortcomings and limitations in future work.

4. Conclusion

This work proposes a feature-enhanced method for 3D semantic segmentation of grape bunches' regions of interest and the PointResNet model with improved learning capabilities. Based on 2D semantic segmentation, the method maps 2D images to 3D space to assign color features to point clouds and incorporate point cloud normal vector features. Because the enhanced features integrate more surface appearance and spatial structural information, we can obtain more accurate 3D spatial semantic information around the grape bunches. Experimental results verify the effectiveness of this method, which achieves higher overall accuracy and intersection over union compared to traditional methods. The proposed method, applied to the PointNet++ for segmentation, achieves a overall accuracy of 96.3% and an mean accuracy of 89.5% for segmentation with a scene inference prediction time of 0.3 s, which is an improvement of 24.8% and 24.5% respectively compared to segmentation using only positional features. Additionally, there is a 33.3% improvement in total accuracy compared to the traditional method. On the premise of ensuring speed and accuracy, this method can provide precise three-dimensional semantic information for robots and establish a necessary foundation for subsequent collision-free and undamaged picking. It should be noted that our method still needs to be experimented with and tested in more actual farms to verify its universality. Future work will focus on this aspect of research.

CRediT authorship contribution statement

Jiangtao Luo: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology. **Dongbo Zhang:** Writing – review & editing, Writing – original draft, Validation, Supervision, Formal analysis, Conceptualization. **Lufeng Luo:** Writing – review & editing, Writing – original draft, Formal analysis, Conceptualization. **Tao Yi:** Writing – review & editing, Writing – original draft.

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.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported in part by the Joint Fund for Areal Innovation and Development of NSFC, China (U19A2083), Key Project of Guangdong Provincial Basic and Applied Basic Research Fund Joint Fund, China (2020B1515120050).

References

Ariel522, 2023. OnlyGrape. URL https://aistudio.baidu.com/datasetdetail/239270/0. Data set.

Bochkovskiy, A., Wang, C.-Y., Liao, H.-Y.M., 2020. Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934.

Brown, J., Sukkarieh, S., 2021. Design and evaluation of a modular robotic plum harvesting system utilizing soft components. J. Field Robotics 38 (2), 289–306.

Cao, X., Zou, X., Jia, C., Chen, M., Zeng, Z., 2019. RRT-based path planning for an intelligent litchi-picking manipulator. Comput. Electron. Agric. 156, 105–118.

Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., et al., 2015. Shapenet: An information-rich 3d model repository. arXiv preprint arXiv:1512.03012.

Chen, L.-C., Papandreou, G., Schroff, F., Adam, H., 2017. Rethinking atrous convolution for semantic image segmentation. arXiv preprint arXiv:1706.05587.

Du, H., Yan, X., Wang, J., Xie, D., Pu, S., 2023. Rethinking the approximation error in 3d surface fitting for point cloud normal estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9486–9495.

Guo, Y., Wang, H., Hu, Q., Liu, H., Liu, L., Bennamoun, M., 2020. Deep learning for 3d point clouds: A survey. IEEE Trans. Pattern Anal. Mach. Intell. 43 (12), 4338–4364.

Han, X., Zhang, Z., Ding, N., Gu, Y., Liu, X., Huo, Y., Qiu, J., Yao, Y., Zhang, A., Zhang, L., et al., 2021. Pre-trained models: Past, present and future. AI Open 2, 225–250.

He, K., Gkioxari, G., Dollár, P., Girshick, R., 2017. Mask r-cnn. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2961–2969.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778.

Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., et al., 2019. Searching for mobilenetv3. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1314–1324.

Kang, H., Zhou, H., Wang, X., Chen, C., 2020. Real-time fruit recognition and grasping estimation for robotic apple harvesting. Sensors 20 (19), 5670.

Kurtser, P., Ringdahl, O., Rotstein, N., Berenstein, R., Edan, Y., 2020. In-field grape cluster size assessment for vine yield estimation using a mobile robot and a consumer level RGB-D camera. IEEE Robot. Autom. Lett. 5 (2), 2031–2038.

Lehnert, C., English, A., McCool, C., Tow, A.W., Perez, T., 2017. Autonomous sweet pepper harvesting for protected cropping systems. IEEE Robot. Autom. Lett. 2 (2), 872–879.

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S., 2017. Feature pyramid networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2117–2125.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L., 2014. Microsoft coco: Common objects in context. In: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13. Springer, pp. 740–755.

Lin, G., Zhu, L., Li, J., Zou, X., Tang, Y., 2021. Collision-free path planning for a guava-harvesting robot based on recurrent deep reinforcement learning. Comput. Electron. Agric. 188, 106350.

Liu, Y., Fan, B., Xiang, S., Pan, C., 2019. Relation-shape convolutional neural network for point cloud analysis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8895–8904.

Liu, Z., Hu, H., Cao, Y., Zhang, Z., Tong, X., 2020. A closer look at local aggregation operators in point cloud analysis. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIII 16. Springer, pp. 326–342.

Long, J., Shelhamer, E., Darrell, T., 2015. Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3431–3440.

Luo, L., Tang, Y., Zou, X., Wang, C., Zhang, P., Feng, W., 2016. Robust grape cluster detection in a vineyard by combining the AdaBoost framework and multiple color components. Sensors 16 (12), 2098.

Luo, L., Yin, W., Ning, Z., Wang, J., Wei, H., Chen, W., Lu, Q., 2022. In-field pose estimation of grape clusters with combined point cloud segmentation and geometric analysis. Comput. Electron. Agric. 200, 107197.

Moreno, H., Rueda-Ayala, V., Ribeiro, A., Bengochea-Guevara, J., Lopez, J., Peteinatos, G., Valero, C., Andújar, D., 2020. Evaluation of vineyard cropping systems using on-board rgb-depth perception. Sensors 20 (23), 6912.

Ning, Z., Luo, L., Ding, X., Dong, Z., Yang, B., Cai, J., Chen, W., Lu, Q., 2022. Recognition of sweet peppers and planning the robotic picking sequence in high-density orchards. Comput. Electron. Agric. 196, 106878.

Ning, Z., Luo, L., Liao, J., Wen, H., Wei, H., Lu, Q., 2021. Recognition and the optimal picking point location of grape stems based on deep learning. Trans. Chin. Soc. Agric. Eng. 37 (9), 222–229.

Pan, S.J., Yang, Q., 2009. A survey on transfer learning. IEEE Trans. Knowl. Data Eng. 22 (10), 1345–1359.

Pérez-Zavala, R., Torres-Torriti, M., Cheein, F.A., Troni, G., 2018. A pattern recognition strategy for visual grape bunch detection in vineyards. Comput. Electron. Agric. 151, 136–149.

Qi, C.R., Su, H., Mo, K., Guibas, L.J., 2017a. Pointnet: Deep learning on point sets for 3d classification and segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 652–660.

Qi, C.R., Yi, L., Su, H., Guibas, L.J., 2017b. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. Adv. Neural Inf. Process. Syst. 30.

Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You only look once: Unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 779–788.

Romeo, L., Devanna, R., Marani, R., Matranga, G., Biddoccu, M., Milella, A., 2023. Scale-invariant semantic segmentation of natural RGB-d images combining decision tree and deep learning models. In: Multimodal Sensing and Artificial Intelligence: Technologies and Applications III. Vol. 12621, SPIE, pp. 257–260.

Sa, I., Lehnert, C., English, A., McCool, C., Dayoub, F., Upcroft, B., Perez, T., 2017. Peduncle detection of sweet pepper for autonomous crop harvesting—combined color and 3-D information. IEEE Robot. Autom. Lett. 2 (2), 765–772.

Santos, T.T., de Souza, L.L., dos Santos, A.A., Avila, S., 2020. Grape detection, segmentation, and tracking using deep neural networks and three-dimensional association. Comput. Electron. Agric. 170, 105247.

Shamshiri, R.R., Hameed, I.A., Karkee, M., Weltzien, C., 2018. Robotic harvesting of fruiting vegetables: A simulation approach in V-REP, ROS and MATLAB. In: Proceedings in Automation in Agriculture-Securing Food Supplies for Future Generations. Vol. 126, pp. 81–105.

Sozzi, M., Cantalamessa, S., Cogato, A., Kayad, A., Marinello, F., 2022. Automatic bunch detection in white grape varieties using YOLOv3, YOLOv4, and YOLOv5 deep learning algorithms. Agronomy 12 (2), 319.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., 2015. Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–9.

Thomas, H., Qi, C.R., Deschaud, J.-E., Marcotegui, B., Goulette, F., Guibas, L.J., 2019. Kpconv: Flexible and deformable convolution for point clouds. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6411–6420.

Wang, X., Singh, D., Marla, S., Morris, G., Poland, J., 2018. Field-based high-throughput phenotyping of plant height in sorghum using different sensing technologies. Plant Methods 14, 1–16.

Williams, H., Ting, C., Nejati, M., Jones, M.H., Penhall, N., Lim, J., Seabright, M., Bell, J., Ahn, H.S., Scarfe, A., et al., 2020. Improvements to and large-scale evaluation of a robotic kiwifruit harvester. J. Field Robotics 37 (2), 187–201.

Yan, B., Fan, P., Lei, X., Liu, Z., Yang, F., 2021. A real-time apple targets detection method for picking robot based on improved YOLOv5. Remote Sens. 13 (9), 1619.

Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J., 2017. Pyramid scene parsing network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2881–2890.

Zhou, H., Wang, X., Au, W., Kang, H., Chen, C., 2022. Intelligent robots for fruit harvesting: Recent developments and future challenges. Precis. Agric. 23 (5), 1856–1907.