Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Real-time Detection and Characterization of Trunks and Upright Branches of Pear Trees for Automatic Dormant Pruning

Hao Sun
nanjing agricultural university
Gengchen Wu
nanjing agricultural university
Hu Xu
nanjing agricultural university
Jiaqi Li
nanjing agricultural university
Zhidi Zhou
nanjing agricultural university
Shutian Tao
nanjing agricultural university
Wei Guo
The University of Tokyo
Kaijie Qi
nanjing agricultural university
Hao Yin
nanjing agricultural university
Shaoling Zhang
nanjing agricultural university
Seishi Ninomiya
nanjing agricultural university
Yue Mu
yuemu@njau.edu.cn

nanjing agricultural university

Research Article

Keywords: Pear trees, Dormant pruning, Trunk detection, Upright branches, Characterization

Posted Date: November 7th, 2025

DOI: https://doi.org/10.21203/rs.3.rs-7675229/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Version of Record: A version of this preprint was published at Precision Agriculture on February 9th, 2026. See the published version at https://doi.org/10.1007/s11119-025-10310-9.

Real-time Detection and Characterization of Trunks and Upright Branches of Pear Trees for Automatic Dormant Pruning

Hao Sun [1] • Gengchen Wu [1] • Hu Xu [1] • Jiaqi Li [1] • Zhidi Zhou [1] • Shutian Tao [2,3] • Wei Guo [4] • Kaijie Qi [2] • Hao Yin [2] • Shaoling Zhang [2] • Seishi Ninomiya [1,4] • Yue Mu [1*]

Abstract

Purpose Dormant pruning is critical for fruit tree management and maintaining fruit quality. Traditional manual pruning is labor-intensive, driving interest in automated robotic dormant pruning. However, automated robotic dormant pruning meets significant challenges in trunk and branches detection, due to the complexity of the orchard environment and the interlacing of the branches. This paper proposes an automatic method for real-time detection and pruning of pear tree trunks and upright branches using an RGB-D camera.

Methods Pear trunk detection was conducted by enhancing the You Only Look Once version 5 Nano (YOLOv5n) model with Squeeze-and-Excitation Networks (SENet) and optimizing the anchor boxes. For branch segmentation, YOLOv8n-seg integrated with Dynamic Snake Convolution (DSConv) and Focal Scale Intersection over Union (Focal_SIoU) was employed. The length and angle of the branches were calculated from the generated mask images, and the pruning position was determined. A PRUNING_ROS package was developed for real-time orchard applications.

Results The evaluation demonstrated 96.7% mean average precision (mAP) for trunk detection and 82.6% mAP for branch segmentation in test dataset. Field test results showed that the mean absolute error (MAE) of trunk distance localization compared to manual measurements was 3.71 cm, with the root mean square error (RMSE) of 3.84 cm, and the frames per second (FPS) of 31.6. The MAE was 2.3 cm (RMSE: 2.6 cm) for pruning points in depth direction and 2.34° (RMSE: 2.71°) for upright branches angle, with the FPS of 36.2. The field pruning experiment showed a pruning success rate of 47.6%.

Conclusion This method provides technical support for the operation of fruit tree pruning robots, representing a step toward the full automation of fruit tree management.

Keywords Pear trees • Dormant pruning • Trunk detection • Upright branches • Characterization

Introduction

Fruit tree pruning is an important technique in modern fruit tree cultivation and management, and plays a crucial role in promoting tree healthy growth and improving fruit yield and quality (Davenport, 2006). However, fruit tree pruning is a time-consuming and labor-intensive process that requires significant human resources (Ma & Jiang, 2024). The most labor-intensive fruit tree pruning is the dormant pruning. According to statistics, the labor hours spent on dormant tree pruning account for about 20% of the total labor hours cost in the entire fruit tree production. As a major pear producer, China has the largest area of pear fruit tree cultivation in the world (Zhang et al., 2024). It is urgent to develop the techniques and robots for pear tree dormant pruning.

Accurate detection of the tree is a fundamental requirement for pruning robot localization and operation (Huang, et al., 2023). In fruit tree detection research, two tasks were typically involved: canopy segmentation or trunk detection. Methods such as threshold segmentation (Mu et al., 2018; Shen et al., 2022) and support vector machine (Chen, et al., 2020; Chen et al. 2018) were employed to identify and segment fruit trees. Liao et al. (2021) employed a Gaussian high-pass filtering algorithm to highlight the edge information of the tree trunks, and then detects the trunks and non-trunks based on thresholding and random forest classification methods. Shen et al. (2022) proposed a fast trunk recognition method using color, depth, width, and parallel edge features. They converted the color image to the Hue, Saturation and Value (HSV) color space and then performed super-pixel segmentation on the S-component of the HSV color space. The results showed a 2.52% improvement in accuracy under strong lighting conditions compared to weak lighting conditions. However, the robustness of detection in the complex terrain still needs to be improved, and the detection is not suitable for real-time robot operation.

For accurate dormant pruning of fruit trees, accurate branch detection of the tree is the crucial technique (Cuevas-Velasquez, et al., 2020). Gao et al. (2006) used a 2D color camera and image processing algorithms to capture images indoors. By using a white cloth to remove the noise from backgrounds, they detected the branches to be pruned of grapevines and fitted the outlines of pruned grapevine branches. Dey et al. (2012) reconstructed 3D point clouds of grapevines from images, and classified the structure of grapevines into leaves, branches, and fruit by using a combination of shape and color features. Medeiros et al. (2017) effectively segmented the trunk and branch point clouds by modelling each segment as a cylinder. The average segmentation accuracy of the primary branches reached 98%, but these branches were along the trunk. Overall, image analysis is hard to detect branch in real orchard environment, and 3D segmentation is effective but may cost a lot time in 3D point cloud reconstruction and processing.

In recent years, the emergence of deep learning-based computer vision algorithms has provided innovative solutions. The application of deep learning has overcome the limitations of traditional algorithms that rely on finite features such as colors and textures, which are highly sensitive to environmental changes. This has significantly improved the accuracy of object detection under varying lighting conditions and complex backgrounds (Zhang et al., 2018). At present, these algorithms have been widely applied in various agricultural detection fields, including fruit detection (Kim et al., 2022), weed identification (Rai et al., 2024), pest and disease detection (Xue et al., 2023), and fruit tree parameter acquisition (Wu et al., 2020).

In the studies of fruit tree detection, Simoes et al. (2022) developed a peach tree trunk detection algorithm based on the Single Shot MultiBox Detector algorithm, which achieved a precision of 92.3% for trunk detection. Ye et al. (2023) proposed an improved YOLOv5 method for tree detection in densely planted areas by incorporating the Convolutional Block Attention Module attention mechanism and adaptively adjusting the receptive field based on the size and shape of fruit tree canopies, which significantly improved the accuracy of tree detection. Cao et al. (2024) proposed a vine trunk detection model based on YOLOv8 by introducing an Efficient Multi-scale Attention mechanism and using the Mix-Shelter method to augment the dataset, which improved the robustness of the model in complex environments. Accurately detecting trees in orchards and measuring their distances to robot is a critical component of fruit tree pruning. However, most existing trunk detection methods neglects their spatial positions. In addition, uneven terrain, which leads to sensor tilt, increases the morphological discrepancy of trunks, demanding higher robust on trunk detection performance.

In the field of branch detection, Chen et al. (2021) compared different segmentation models for segmenting partially occluded V-shaped branches of an apple tree. The results showed that Convolutional Networks for Biomedical Image Segmentation (U-Net) outperformed other models in accuracy. Borrenpohl et al. (2023) performed instance segmentation of branches in dormant cherry trees, and demonstrated that Mask Region-based convolutional neural network (Mask R-CNN) under active illumination, achieved the higher accuracy compared to natural illumination conditions. The above studies focus primarily on the identification of branches to be pruned. However, there is still a gap to combine the detection in practical pruning operations. It is necessary to characterize these branches in the robot view, i.e., measure the branch angle to control the cutting for effective pruning, and obtain the actual coordinate positions of the pruning points.

In addition, the integration of branch detection algorithms with hardware control remains limited. In other agricultural applications, successful integration of algorithms and hardware had been achieved, such as combining vision systems with robotic arm control in fruit-picking systems (Wang et al., 2023). However, in the field of dormant fruit tree pruning, despite the high accuracy of existing branch detection technologies, the comprehensive fusion of algorithms and hardware systems still needs further improvement. In particular, the application of real-time pruning object identification and precise pruning position prediction remains underdeveloped. Therefore, achieving tight integration between algorithms and hardware in branch pruning remains one of the key challenges in current research.

The pear tree architecture involved in this study is similar to the Upright Fruiting Offshoots (UFO) architecture (Whiting, 2018), with the branches to be pruned referred to as upright branches on the side branches. This study focuses on the practical application scenarios of dormant fruit tree pruning, and aimed to improve the precision and efficiency of pruning upright branches of pear trees during dormancy. It realizes real-time identification and localization of pear tree trunks, as well as real-time detection and pruning of upright branches. The contributions include:

(1) Proposed a real-time trunk detection model by improved YOLOv5n with SENet modules, which increased the detection capability and speed of trunks.

(2) Integrated DSConv into YOLOv8n-seg for the accurate identification of upright branches, and enabling precise measurement of inclination angle and pruning position prediction of upright branches.

(3) Developed a PRUNING_ROS package for real-time robot operation in the field, including trunk detection, segmentation, characterization and pruning of upright branches.

Materials and methods

The framework of pruning system

This study proposes a dormant pruning system of upright branches (Fig. 1). The system consists of three main components: data collection and model training (Fig. 1(a)), trunk localization and parameter measurement of upright branch (Fig. 1(b)), and automated upright branches pruning (Fig. 1(c)). First, images of tree trunks and upright branches from dormant pear orchard were collected, and improved YOLOv5 and YOLOv8 algorithms were used for trunk detection and branch recognition. Next, the trained model was integrated into the Robot Operating System (ROS) to achieve real-time trunk localization and upright branch recognition. After locating the target tree, the robotic arm is elevated to 60° for pose initialization. Finally, to avoid misidentifying the upright scaffold branch and to accurately identify the upright branches on the side branches, the base was first controlled to rotate 10° clockwise to obtain the coordinates of the pruning point and the angle parameters of the upright branches. After completing the pruning operation, the base was then rotated 20° counterclockwise, and the operation was repeated to complete the pruning of the upright branches.

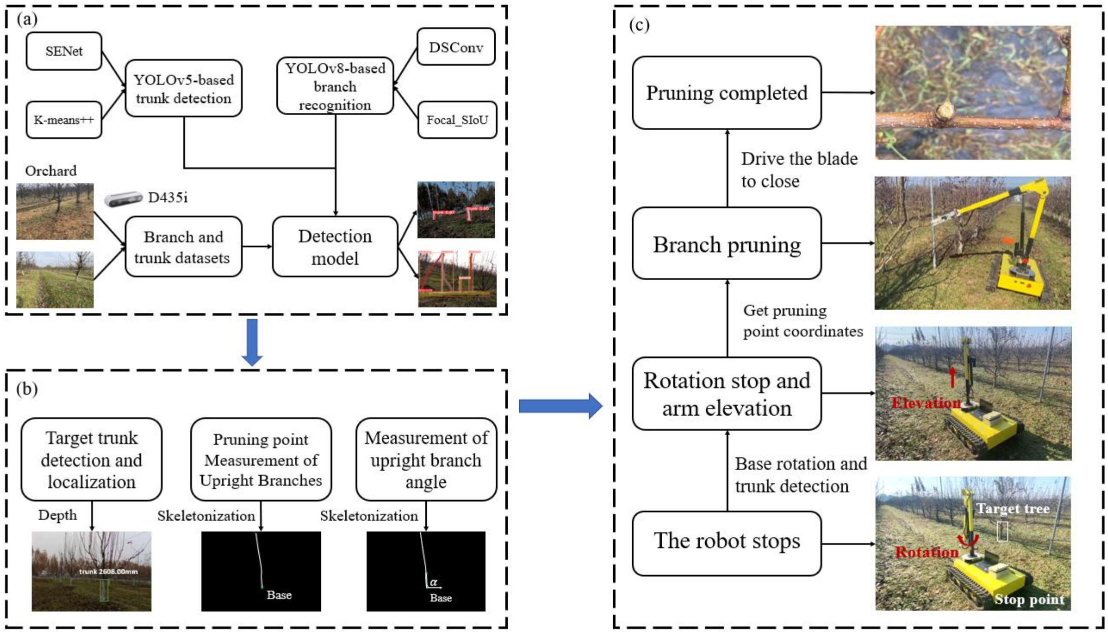

Fig. 1 The workflow of dormant pruning system of upright branches. **(a)** Data collection and model training. **(b)** Trunk localization and parameter measurement of upright branch. **(c)** Automated upright branches pruning

Data collection

In this study, we used an Intel RealSense D435i depth camera (Intel Corporation, USA) for detecting and segmenting trunks and upright branches. The dataset was collected from the pear orchard at the Baima Experimental Station of Nanjing Agricultural University and the Jinling Pear Orchard (a commercial orchard in Nanjing), with pear tree cultivation conducted in 2017 and 2015, respectively. Two pear tree cultivars, 'Cuiguan' and 'Hosui', with the training systems of “2+1” and “3+1” architecture, were selected for testing (Fig. 2(b)). The row spacing and column spacing of the pear trees were 4 m and 3.5 m, respectively.

The collected RGB images and depth images of trunks and upright branches had a resolution of 848 × 480. The depth images were aligned with the RGB images using the camera's built-in software (https://github.com/IntelRealSense/librealsense) to ensuring that the depth value of each pixel corresponded to that in the RGB image.

A total of 800 RGB images and 800 depth images of trunks, as well as 1620 RGB images and 1620 depth images of upright branches, were collected during the experiment. During trunk image collection, the camera was positioned 2 m away from the trunk under both sunny and cloudy conditions, at a height of 0.6 m above the ground and oriented horizontally. For branch image collection, the camera was placed 0.6–0.8 m from the trunk, with the height ranging from 1.0 to 1.5 m above the ground (Fig. 2(a)). To enhance the diversity and robustness of the dataset, data augmentation techniques were applied, including brightness adjustment, translation, and mirroring, doubling the dataset in size. The dataset was then randomly divided into training, validation, and testing sets at a ratio of 8:1:1. Among them, the training set was used for model training, while the validation set and test set were employed for model ablation experiments and evaluation. The trunks in foreground were annotated using LabelImg (https://github.com/HumanSignal/labelImg), while upright branches and primary branches were annotated using Labelme (https://github.com/wkentaro/labelme), as shown in Fig. 2(c).

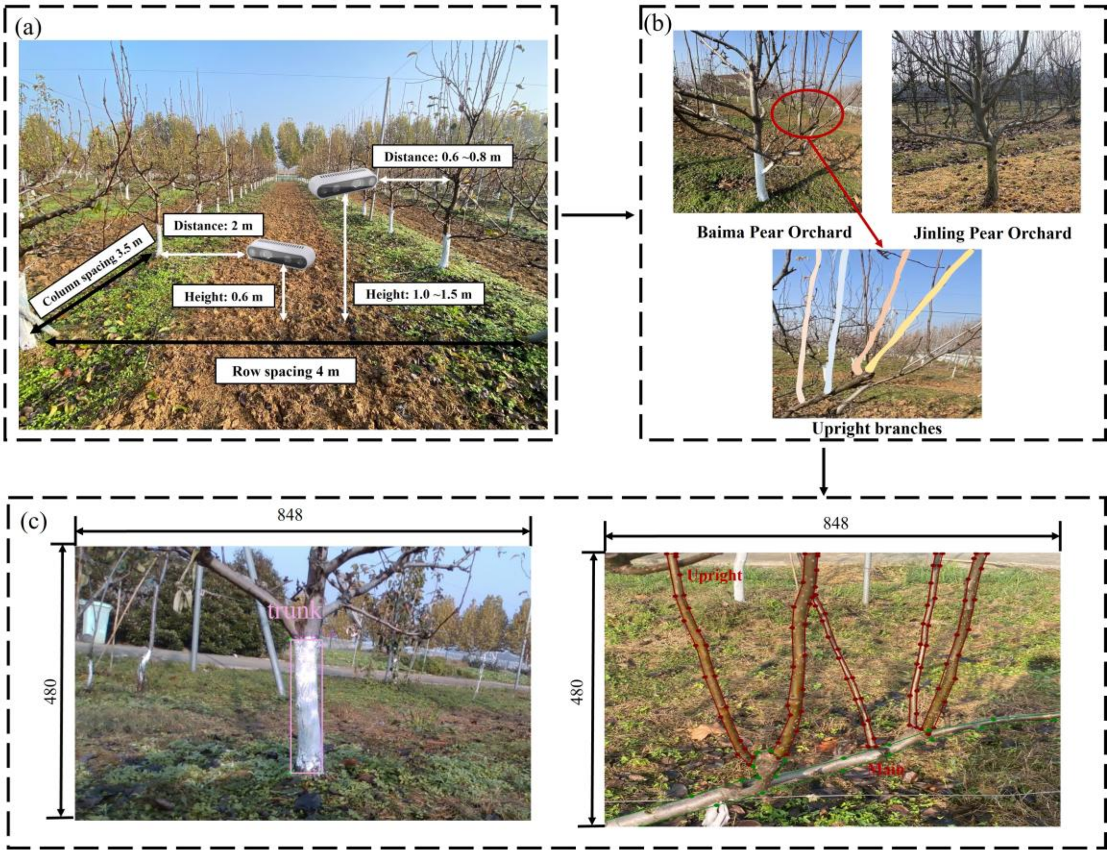

Fig. 2 Schematic diagram of data acquisition for trunks and upright branches on the side branches. **(a)** Data collection. **(b)** Types of branches to be pruned. **(c)** Data annotation

Trunk detection and localization

Model construction

In this study, the lightweight YOLOv5n object detection model was used to detect trunks, offering relatively fewer parameters and faster detection speed compared to other YOLO model versions (v8, v11, etc.). However, in real orchards, factors such as uneven terrain and the inclination of trunks increased the complexity of the object morphology. As a lightweight model, YOLOv5n had significant advantages in computation-constrained scenarios, with fewer parameters and relatively limited feature extraction capabilities (Ding et al., 2023).

To increase its adaptability and accuracy in detecting trunks in orchard, this paper had made two improvements to the YOLOv5n model. The improvements were as follows: (1) in the detection head, the original C3 module was replaced with an improved C3_SE module that incorporated the SENet attention mechanism, which strengthened the focus on key features and thereby improved the extraction of trunk features; (2) the K-means++ algorithm was used to cluster the dimensions (width and height) of the bounding boxes in the annotated dataset, which optimized the initial anchor box settings and improved the robustness of the model to changes in object scale. The model structure was shown in Fig. 3.

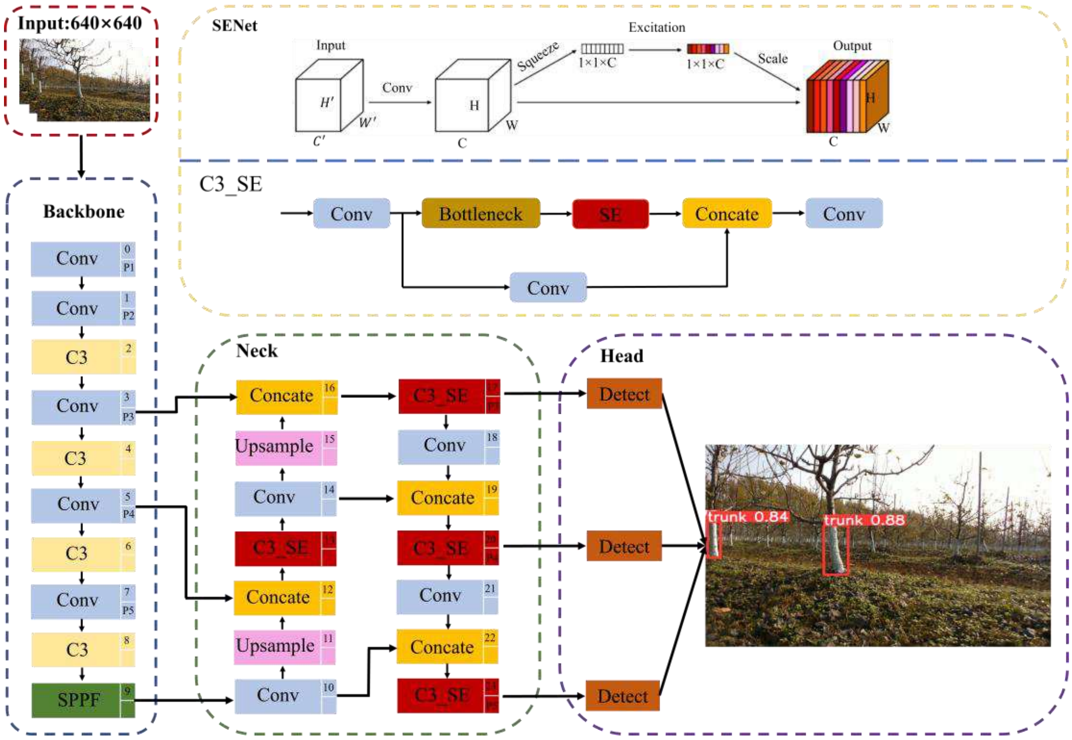

Fig. 3 The architecture of the improved YOLOv5n model. C3_SE: the C3 module for the introduction of SENet

Attention mechanism

The SENet attention mechanism module dynamically adjusted the weight of each channel, allowing the network to adaptively enhance critical channels (Qiu et al., 2022). At the same time, it focused more attention on features related to trunks, thereby improving the detection capability in varied light conditions. The SENet module consisted of three main stages: the squeeze layer, the excitation layer, and the scale layer.

Suppose the input feature map was X, with dimensions $H' \times W' \times C'$. After the conventional convolution operation, intermediate features U changed to $H \times W \times C$ after the convolution operation. The squeeze layer was responsible for global average pooling, compressing the input feature map into a real vector Z, and establishing connections for each channel (calculation formula in Appendix, Eq. (1)).

The excitation layer transformed the real number vector Z obtained from the squeeze layer through a fully connected network with nonlinear activation, producing a real vector S representing the weights for each channel (calculation formula in Appendix, Eq. (2)).

The scale layer applied the channel weights S, obtained from the excitation layer, to the corresponding channels of the feature map through element-wise multiplication, thus completing the feature recalibration (calculation formula in Appendix, Eq. (3)).

K-means++ clustering of anchor boxes

YOLOv5 used the K-means algorithm to determine anchor boxes, but the K-means clustering process heavily relied on the random initialization of cluster centers, which may lead to local optimal solutions. This study used the K-means++ clustering algorithm to optimize the width and height of the detection boxes in the annotated dataset. The setting of anchor boxes was as follows: (9, 18), (15, 23), (12, 36), (18, 31), (21, 45), (33, 55), (24, 76), (33, 89), and (46, 101). These updated anchor boxes were better suited to the diverse scale distribution of trunks in orchard scenarios, enabling more efficient and precise trunk detection and localization in complex environments.

Target trunk detection and localization

After the robot moved to the location near the target tree that to be pruned, the target trunk which was the nearest trunk, was detected by depth to the robot. The depth images were collected with the RGB images at the same time, and each pair was automatically aligned. We filtered out the other tree trunks behind the target trunk, by using the average depth of each detected trunk, and integrated the trunk detection model into the fruit tree pruning system. Through controlling the rotation of the base joint of the robotic arm, real-time detection and localization of the target trunk were achieved (Fig. 4). When the horizontal coordinate of the bounding box of the target trunk was located at the center of the image, the robotic base stopped rotating.

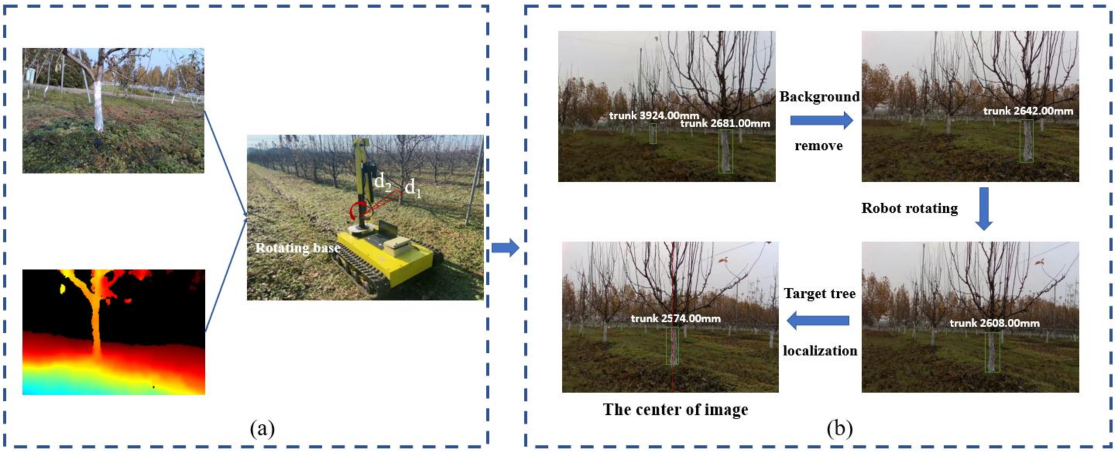

Fig. 4 The schematic diagram of target trunk detection and localization. **(a)** Image acquisition and model application. **(b)** Target trunk acquisition. The numbers in the image represent the detection distance of the trunk

Recognition and parameter measurement of upright branches

Recognition of upright branches

This study selected YOLOv8 as the framework for branch segmentation, primarily because it was more widely applied on the Robot Operating System (ROS) compared to other YOLO models, making it better suited for robot application needs. YOLOv8 introduced an additional segmentation branch on top of its single-stage detector. The segmentation component was inspired by the You Only Look at Coefficients (YOLACT) framework, which divided object detection and segmentation into two parallel tasks (Li et al., 2023). Given the differences between the upright branches and other branches mainly on morphological and structural features, we modified the YOLOv8 network to improve its segmentation performance by introducing Dynamic Snake Convolutions, as shown in Fig. 5.

Firstly, in the Neck section before each detection head, the C2f module was replaced with a C2f-DS structure that introduced DSConv. This improvement aimed to enhance the model's ability to capture local features, particularly boosting its perception of complex shapes, such as the serpentine-like upright branches. Additionally, the original loss function, Complete Intersection over Union (CIoU), was replaced with Focal_SIoU to increase the model's performance on objects of small sample size, further optimizing its performance in branch detection.

By using the improved YOLOv8n-seg model, we recognized the upright branches on pear trees in the collected images, and obtained the corresponding mask image of the branches.

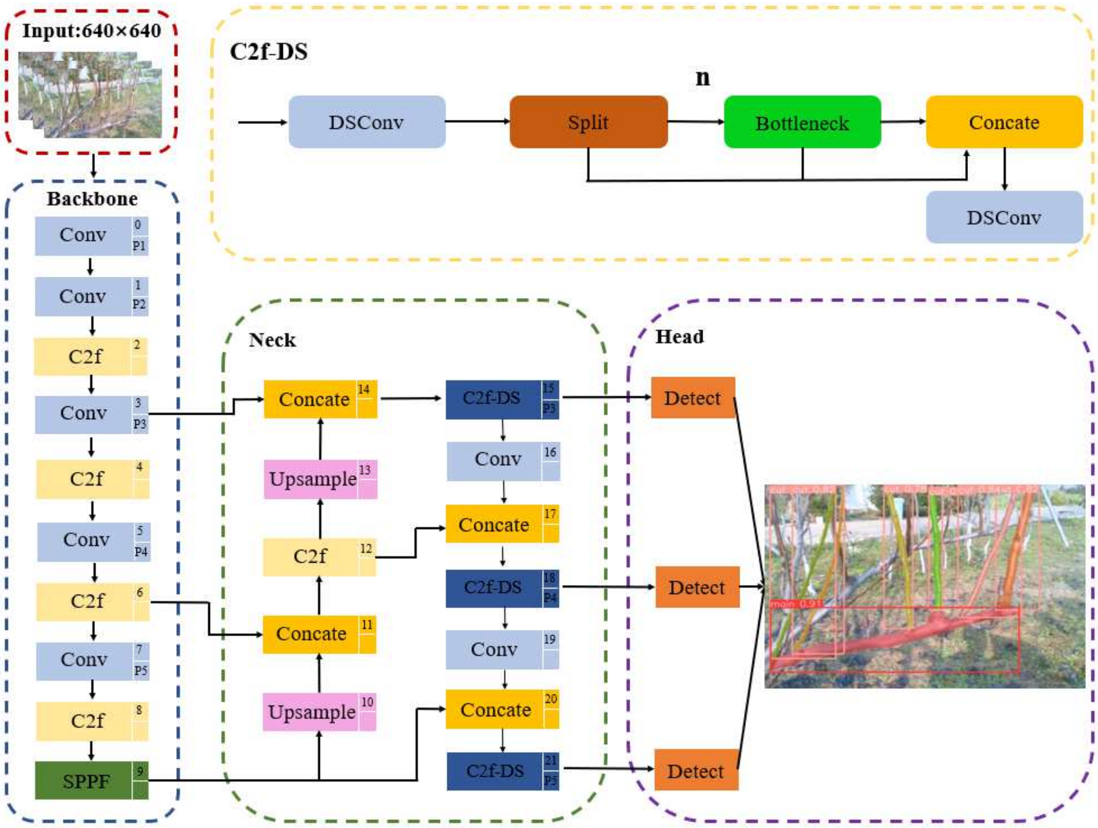

Fig. 5 The architecture of the improved YOLOv8n-seg model. C2f-DS: the C2f module for the introduction of DSConv

Dynamic Snake Convolutions

DSConv was effective at segmenting tubular and elongated structures, making it well suited for capturing the structural characteristics of branches (do Nascimento et al., 2019). By dynamically adjusting the parameters of convolution kernel through learning and adaptive mechanisms, DSConv could capture branch features of different scales and morphologies in input data. Therefore, introducing DSConv into the YOLOv8 model enhances its ability to detect and segment branches, thereby improving the stability and accuracy of branch detection.

This paper improved the C2f module in the Neck part by introducing DSConv into it. The convolution kernels were incorporated with continuity constraints, where the single convolution kernel was given in Eq. (4) in the appendix.

This convolution operation was performed on a two-dimensional image, where the center of the convolution kernel was located at the pixel (x, y), taking into account the coordinates of the eight surrounding neighboring pixels. To enhance the flexibility of the convolution kernel and enable it to more effectively capture the complex geometric features of the serpentine-like object, dynamic snake convolution, one variant of deformation convolution, was adopted. Deformation offsets were introduced to allow the convolution kernel to adapt to different shapes and structures, as shown in Fig. 6.

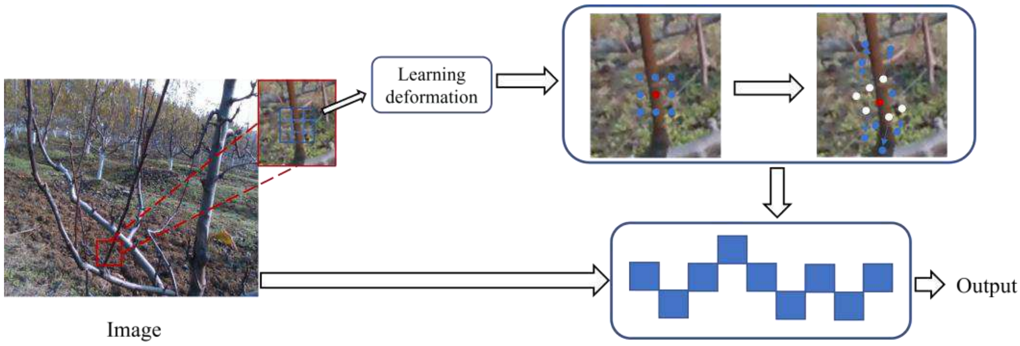

Fig. 6 Schematic diagram of dynamic snake convolution

Focal_SIoU Loss function

YOLOv8 used the CIoU loss function, which had the problem of slow convergence. In addition, the dataset had an imbalance between upright branches and primary branches, i.e., more upright branches and fewer primary branches. To address this issue, the Focal_SIoU loss function was introduced. This function optimized the bounding box regression process by introducing the vector angle between the ground truth box and the predicted box (Niu et al., 2024). Compared to CIoU, Focal_SIoU effectively reduced the regression degrees of freedom, which significantly improved the model's convergence speed. Additionally, it adopted a training strategy that reduced the weights of simple negative samples, enabling the detection module to better distinguish between upright branches and primary branches. Focal_SIoU was composed of the Focal Loss function and SIoU, and its calculation was defined by Eq. (5)-(6) in the appendix.

Measurement of Upright Branches

Pruning point Measurement of Upright Branches

In the pruning process of robot, pruning position determination was essential. Based on this, this study proposed a skeletonization-based pruning point measurement method to determine the pruning position. First, the improved YOLOv8n-seg model was applied to obtaining the corresponding mask image of the branches (Fig. 7(a)). Next, the corresponding mask image of the branches was labeled into single branches based on individual regions (Fig. 7(b)). Finally, the extracted mask of the single upright branches was then skeletonized by scikit-image (https://scikit-image.org), resulting in its skeletonized image (Fig. 7(c)).

During the thinning dominated pruning process of upright branches, most of the pruning were carried out at the base of the branches (Zhang et al., 2024). In this case, the pixel coordinates of the pruning point (u_{prune}, v_{prune}) were the same as the pixel coordinates of the base of the branch (u_1, v_1).

The pixel coordinates (u_{prune}, v_{prune}) of each pruning point obtained from the skeletonized image were used, along with the calibrated camera intrinsic matrix and the depth value D from the depth map (Fig. 7(d)), to calculate the corresponding coordinates of pruning point in the camera coordinate system (Fig. 7(e)). In the case of a depth omission at the pruning point, we used the nearest neighbor interpolation method to select the nearest valid depth data points from the skeletonized branch image to fill in the missing depth information. The calculation formula was given in Eq. (7) in the appendix.

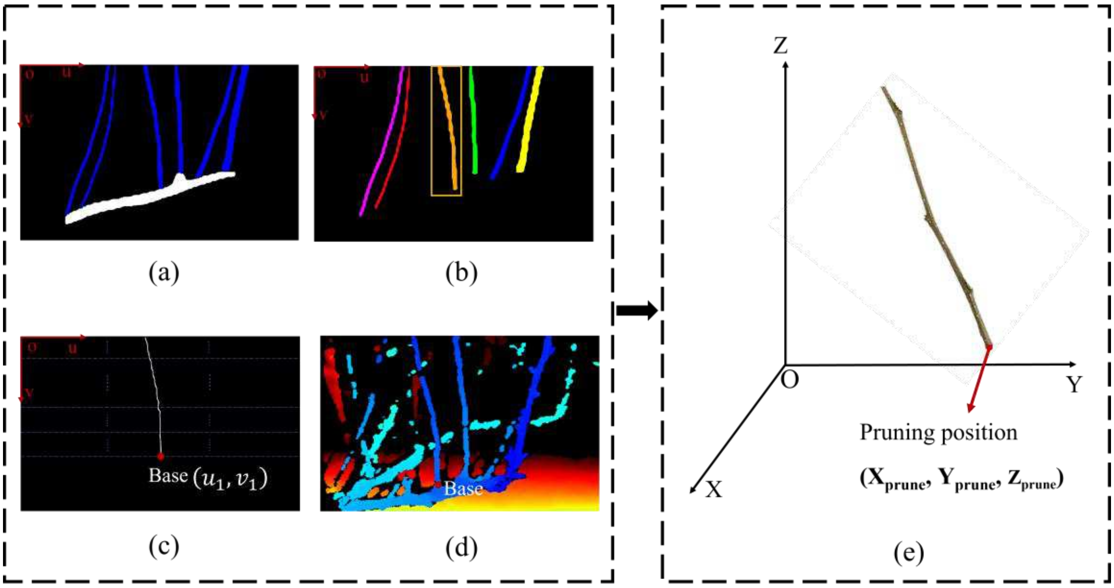

Fig. 7 Diagram of upright branch pruning position prediction. **(a)** Branch mask image. **(b)** Upright branch mask image. **(c)** Skeletonized upright branch. **(d)** Depth image. **(e)** Illustration of the pruning position of the upright branch with the yellow box

Measurement of upright branch angle

When pruning fruit trees, ensuring the correct orientation of the cutting surface for upright branches are critical (Wu et al., 2018). Therefore, accurately measuring the angle of the branch within the field of view and subsequently controlling the rotation angle of the pruning shears are important. Given the curved growth of the branches, the farther from the base, the greater the measurement error is when manually determining the angle. To ensure the accuracy of angle measurements in the field, this study used the base angle measurement of the branches as an example to evaluate the accuracy of the proposed method. We suppose in the small length of ε along the branch, the angle of branch doesn't change. In this study, the distance from the angle measurement point to the base position was set to ε=3 cm (Fig. 8(a)). We calculated the coordinates of the point located 3 cm from the base using Eq. (8) in the appendix. Subsequently, the coordinate of the measurement position on the v-axis, v_ε, was obtained.

Subsequently, the pixel coordinates of the measurement position, $(u_\varepsilon, v_\varepsilon)$ were obtained by traversing the pixel coordinates on the skeleton of the upright branch using v_ε. The base point of the skeleton was then connected to the measurement point, and the inclination angle in the pixel coordinate system was calculated (Fig. 8(b)). The calculation formula was given in Eq. (9)-(11) in the appendix.

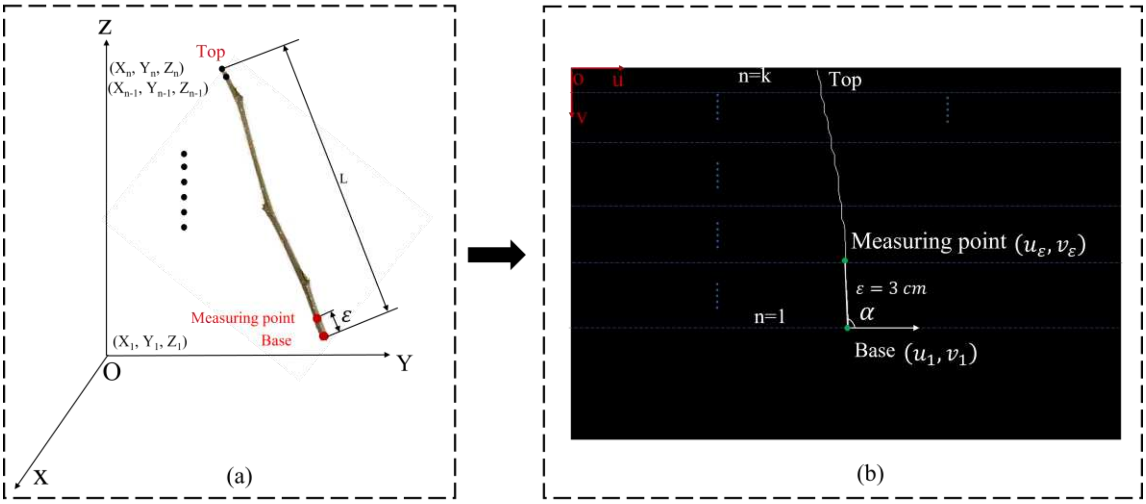

Fig. 8 Schematic diagram of the base angle calculation of the upright branch. **(a)** The measuring point and base point illustration of the upright branch. **(b)** Base angle α was calculated from the base (u_1, v_1) and measuring point $(u_\varepsilon,\ v_\varepsilon)$, where the length between them set to 3 cm

PRUNING_ROS development and field experiments

PRUNING_ROS development

We developed a PRUNING_ROS package, which runs on Ubuntu 20.04 with the Robot Operating System (ROS-Noetic), for real-time detection and characterization of pear tree trunks and upright branches, along with automated pruning of upright branches. The package primarily consisted of the following modules: Realsense_ROS, YOLO_ROS, and Arduino_ROS. The specific workflow was shown in Fig. 9.

The system captured RGB and depth images of trunks and upright branches in real time, as well as removed image distortion using the Realsense_ROS module. Image distortion was calibrated by obtaining the intrinsic matrix of the RealSense D435i camera using the chessboard calibration method. Real-time trunk detection and upright branch segmentation were performed using the developed YOLO_ROS module, which includes Yolov5_ros and Yolov8_ros. Yolov5_ros was used for real-time detection of trunks, providing the position information of target trunk. Yolov8_ros was responsible for branch segmentation tasks, generating branch masks, identifying pruning points for upright branches, and calculating their angles. Arduino_ROS module was the control system for the pruning robot, which received positional information from Yolov5_ros module and Yolov8_ros module and controlled the movement of the robotic arm. Specifically, Arduino_ROS module send pulse signals to control the rotation of the robotic base and used PWM signals to adjust the electric actuator and end effector, enabling the robotic arm to lift (to the appropriate working height), extend (adjust the arm length), and rotate (adjust the angle of the end effector).

In the field pruning experiments, Yolov5_ros first detected the trunk and transmitted its position information to Arduino_ROS module. Then, the electric actuator was controlled via PWM signals to lift the robotic arm to an appropriate angle for identifying the upright branches to be pruned. Yolov8_ros module performed upright branch segmentation, identified pruning points, and calculated angles. The system then passed this information to Arduino_ROS module. After receiving the pruning point coordinates and angles, Arduino_ROS module used inverse kinematics to convert the data into the required movement parameters for the robotic arm. Based on these parameters, Arduino_ROS module controlled the robotic base and electric actuator to accurately position the arm at the pruning point.

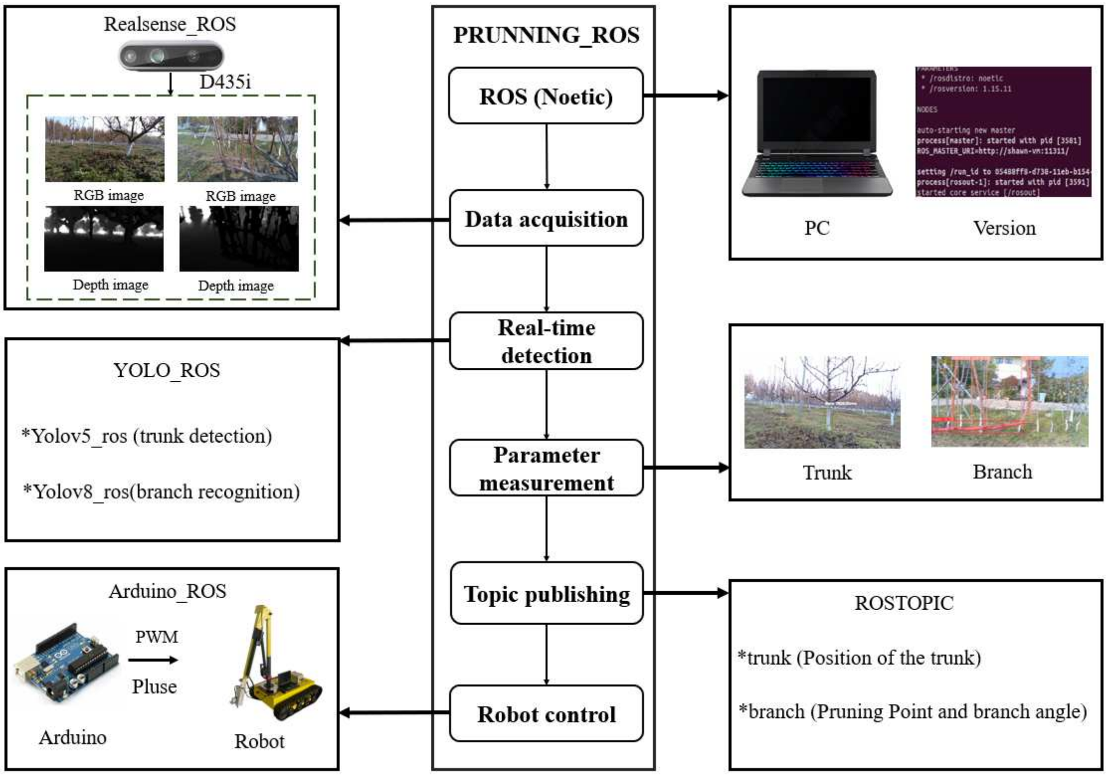

Fig. 9 Schematic diagram of the PRUNING_ROS package

Field experiments

To verify the actual accuracy of the pruning robot in trunk localization tasks, the localization accuracy tests were conducted on 10 trees in November 2024 and March 2025, respectively. During the tests, the robot stopped at designated points between adjacent trees along a preset path, as shown in Fig. 10(a). At stop point a, the robotic arm was first oriented toward the negative Y-axis, and the trunks on both sides were measured by rotating the arm clockwise and counterclockwise, with each direction measured three times and averaged. The robotic arm was then adjusted to face the positive Y-axis, and measurements were performed using the same procedure to detect the two trees nearby. All measurements from the robotic arm were compared with manual field measurements to calculate localization errors (Fig. 10(c)). After completing measurements at the current stop point, the robot moved to the next designated location, where only measurements with the robotic arm facing the positive Y-axis were conducted. In total, 60 trunk localization accuracy validations were completed during the entire experiment.

To evaluate the system's localization error for the pruning points in vision system, we used a measuring tape to measure the distance from the center of the base of the upright branches to the camera (Fig. 10(d)). In addition, in the direction of the camera's field of view, we used a thin line connected the base of the upright branches with to the point up 3 cm, and manually measured the angle between them using a protractor (Fig. 10(e)). Finally, we conducted field pruning experiments on the recognized upright branches to validate the feasibility of the pruning system.

To evaluate the success rate of upright branches pruning, the pruning robot was remotely controlled to navigate to the designated stop point and then automatically rotated horizontally. When the trunk was centered, the robotic base stopped rotating. Subsequently, the robotic arm was raised vertically to 60° through the Arduino_ROS module to establish the initial posture. To accurately identify the upright branches, the robotic base was first controlled to rotate 10° clockwise, an angle that helps avoid misidentifying the upright scaffold branch. In this position, the robot identified all upright branches within its field of view and recorded the distance from the base of each branch to the camera. It then began automatically pruning the nearest branch. Considering the thickness of the pruning blade, to avoid cutting the primary branch, we raised the vertical height by an additional 1 cm when the robotic arm reached its final position. After the end effector of the robotic arm performed the pruning action, the arm returned to the initial position and proceeded to prune the next nearest target branch. This process continued until all identified upright branches with a confidence score greater than 0.7 had been pruned. The camera monitored in real-time whether there were any remaining branches to prune. If no branches were available, the robotic base was controlled to rotate 20° counterclockwise, and the operation was repeated, as shown in Fig. 11.

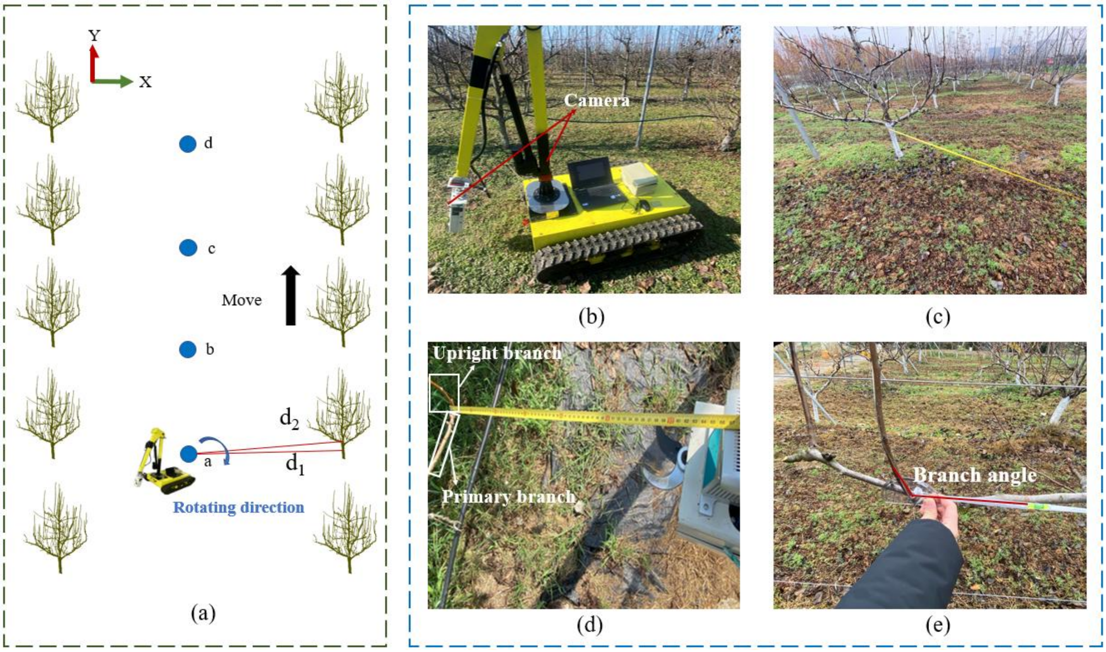

Fig. 10 Schematic diagram of the field experiment. **(a)** Schematic diagram of the test layout, a, b, c and d were the stop point of the implement. **(b)** Schematic diagram of the machine. **(c)** The trunk-to-sensor distance is measured manually. **(d)** The distance from the base of the branch to the camera was measured manually. **(e)** The angle of the upright branches was measured manually. d_1 was the manually measured horizontal distance from the trunk to the camera. d_2 was the horizontal distance from the trunk to the camera as measured by the camera

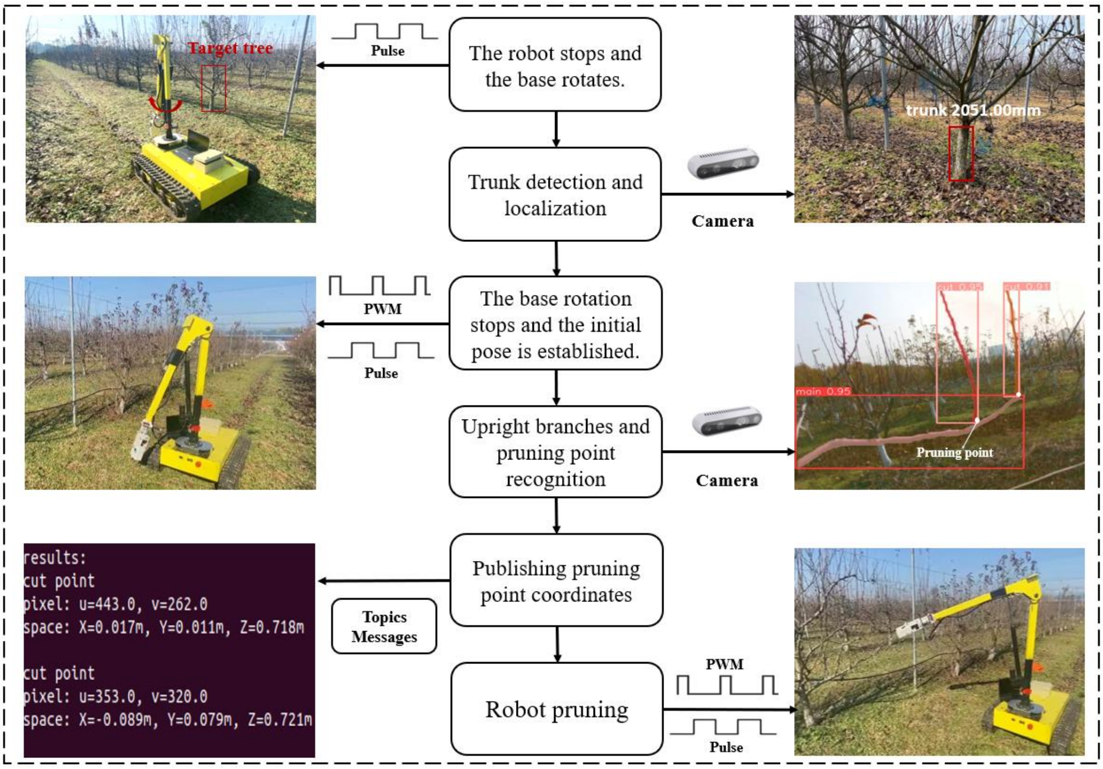

Fig. 11 Schematic diagram of the pruning experiment

Model running and evaluation

Model development and deployment

The deep learning models developed in this study were trained and tested on an PC consisting of an NVIDIA GeForce RTX 4060 GPU and an Intel Core i5-13600KF CPU, with the operating system being Windows 10 and CUDA 11.8. For the trunk detection and the identification of upright branches, transfer learning was applied using pre-trained weights from the COCO dataset. The models were trained with 300 epochs, a patience of 20, and a batch size of 4, with other parameters set to their default values.

The trained model was deployed to the laptop for practical application. The laptop used was an HP Omen (i5-12500H, RTX 3050), with the operating system being Ubuntu 20.04, and the integration was carried out based on ROS Noetic.

Model evaluation

(1) Model evaluation

The trained model was evaluated based on several metrics: Precision (P), Recall (R), mean Average Precision (mAP), FLOPs, and the number of model parameters. The formulas for these metrics were given in Eq. (12)-(14) in the appendix. The Frames Per Second (FPS) was used to compare the detection speed of different models on the PC, and also to evaluate the real-time performance of the model on the laptop.

(2) Trunk and branch measurement accuracy evaluation

The evaluation was conducted using the Mean Absolute Error (MAE), the Root Mean Squared Error (RMSE), the Mean Standard Deviation (MSD) and the Coefficient of Determination (R^2), with the following calculation formulas were given in Eq. (15)-(17) in the appendix. Trunk localization was evaluated using the MAE, MSDand the RMSE., and the distance from the base of the upright branches and the angle of the upright branches were evaluated using MAE, RMSE, and R^2, respectively.

Results and analysis

Trunk detection evaluation

Ablation testing

To validate the effectiveness of the model improvements, ablation testing was conducted using the validation and test sets, comprising a total of 320 images. The improvements were gradually introduced based on the YOLOv5n model to assess their impact. The results were shown in Table 1. As seen in Table 1, after adding the C3 module with the attention mechanism, the precision and mAP improved by 1.7% and 1.4%, respectively, compared to the YOLOv5n model. This indicated that the C3_SE module enhanced the model's ability to perceive spatial information, particularly in detecting objects with varied light conditions. It enabled better focus on key regions, thereby improving the accuracy of trunk detection.

In addition, the improvement in recall indicates that the modified model not only reduces false negatives but also improved its ability to detect a wider range of objects. After introducing K-means++ clustering, the recall and mAP increased by 1.8% and 1.6%, respectively, compared to the YOLOv5n model. This showed that by optimizing the anchor boxes, the model was better able to adapt to the diverse scale distribution of the objects, allowing for more accurate matching of the trunk's scale and thus reducing missed detections. Furthermore, when combining the C3_SE module and K-means++ clustering, the model showed the highest values for detection precision, recall, and mAP. However, the FLOPs and FPS decreased by 15% and 27.8%, indicating that while the detection accuracy improved, the computational efficiency was significantly increased, enabling the model to perform real-time trunk detection tasks more efficiently.

Table 1 Ablation testing results

YOLOv5n	Improvement methods		Precision (%)	Recall (%)	mAP (%)	Parameters	FLOPs (G)	FPS
	C3_SE	K-means++						
√			88.4	91.5	94.1	1765270	4.2	185.2
√	√		90.1	92.2	95.5	1508118	4.0	222.2
√		√	89.9	93.3	95.7	1765270	4.2	232.6
√	√	√	90.5	93.8	96.7	1508118	4.0	256.4

Note: √ indicates that the module was used here.

Detections of different models

To evaluate the model's ability to detect trunks, this study compared the performance of several popular object detection models, including YOLOv5n, YOLOv8n, YOLOX, and YOLOv11n, on the test dataset. The results were shown in Table 2. The improved YOLOv5 model performed the best across all metrics, achieving a mAP of 96.7%. The trunk detection precision was improved by 2.1%, 0.7%, 2.3%, and 1.7% compared to YOLOv5n, YOLOv8n, YOLOX, and YOLOv11n, respectively.

Additionally, the improved model outperformed the others in terms of parameter amount, FLOPs, and FPS, making it more suitable for real-time trunk detection.

Table 2 Accuracy evaluation of different models

Models	Precision (%)	Recall (%)	mAP (%)	Parameters	FLOPs (G)	FPS
YOLOv5n	88.4	91.5	94.1	1872157	4.6	185.2
YOLOv8n	89.8	92.6	95.3	3157200	8.9	76.3
YOLOX	88.2	90.3	93.9	8940546	26.8	5.6
YOLOv11	88.8	91.4	95.8	2590035	6.4	54.6
Ours	90.5	93.8	96.7	1508118	4.0	256.4

To validate the effectiveness of the model in detecting trunks, we took photographs of trunks under different lighting conditions in the orchard and manually adjusted the camera's tilt angle to simulate detection conditions in uneven terrain. The detection results were shown in Fig. 12. It could be seen that when the camera was not tilt (Fig. 12(a) and (b)), all models performed well in detecting trunks. However, our model exhibited the highest confidence in both conditions of sufficient lighting and insufficient lighting.

In addition, in the case of sufficient lighting with camera tilt, YOLOX experienced missed detections (Fig. 12(c)). Both YOLOv5n and YOLOv11n also had missed detections under the conditions of insufficient lighting with camera tilt (Fig. 12(d)). The improved YOLOv5n and YOLOv8 successfully detected trunks under different conditions. The improved model outperformed YOLOv8n in terms of confidence, especially when the camera tilt angle was large (Fig. 12(c)). In this case, the confidence of the trunks detected by YOLOv8n was only 0.35, while the improved model achieved a confidence of 0.78. The experimental results indicate that the improved YOLOv5n model shows significant improvements in both detection accuracy and detection time.

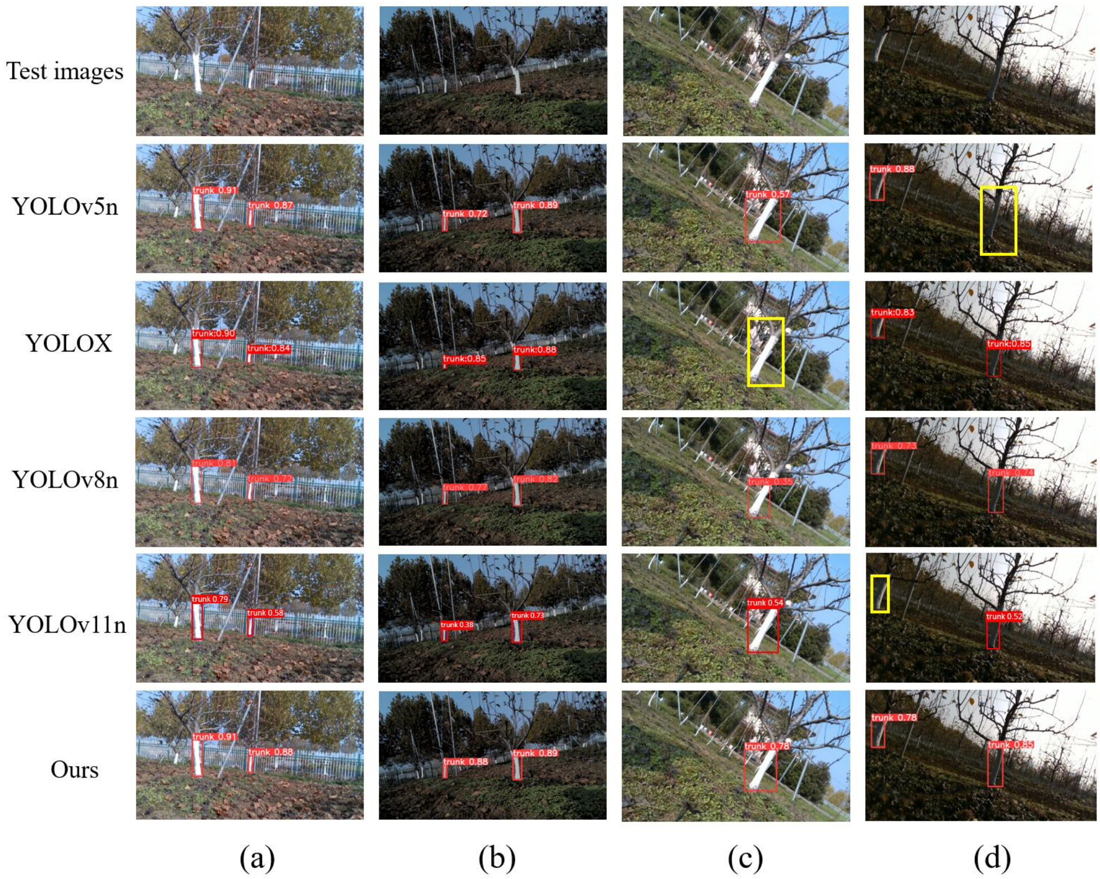

Fig. 12 Comparison of recognition results from different models. **(a)** Conditions of sufficient lighting. **(b)** Conditions of insufficient lighting. **(c)** Conditions of sufficient lighting with camera tilt. **(d)** Conditions of insufficient lighting with camera tilt. The numbers in the image represent the confidence of the trunk detection. The yellow boxes represent missed detections of the trunk

Upright branches recognition evaluation

Ablation testing

To validate the effectiveness of the improvements of the upright branch segmentation model, we conducted ablation testing on the validation and test sets, comprising a total of 648 images to evaluate the two modifications and compared them with the original model. The results are shown in Table 3.

The precision, recall, and mAP of the original YOLOv8-seg for primary branch segmentation were 86.2%, 88.9%, and 90.6%, respectively, while for upright branches segmentation, they were 80.9%, 77.5% and 79.8%. After changing C2f to C2f-DS, the precision for both the primary branch and upright branches increased by 2.3% and 1.8%, respectively. The results indicated that the introduction of DSConv could significantly improve the segmentation accuracy of the branches and increase the completeness of mask segmentation. After introducing Focal_SIoU, the recall of the primary branch and upright branches was improved, increasing by 2.2% and 1.7%, respectively. This indicated that modifying the loss function allowed the model to effectively address the imbalance issue in class sample size and improve the detection performance of the branch. When the model incorporated both DSConv and Focal_SIoU, the precision of upright branches segment increased by 2.6%, recall by 2.9%, and mAP by 2.8%. The experimental results demonstrated that DSConv and Focal_SIoU provide stronger representation capabilities for both the primary branch and upright branches, thereby improving the model's ability to segment the primary branches and upright branches of pear trees.

Due to the addition of DSConv, the model's parameters increased, and FLOPs rose by 0.9 G, leading to a 17.6% increase in detection time. Although the model size and detection time increased, these optimizations significantly improved detection accuracy. Considering both the improvement in model performance and the additional computational cost, this modification was very worthwhile as it greatly improved the segmentation of upright branches.

Table 3 Ablation testing results

YOLOv8n-seg	Improvement methods		Precision (%)		Recall (%)		mAP(%)		Parameters	FLOPs(G)	FPS
	C2f-DS	Focal_SIoU	primary	upright	primary	upright	primary	upright			
√			0.862	0.809	0.889	0.775	0.906	0.798	3264008	12.1	71.4
√	√		0.885	0.827	0.905	0.786	0.931	0.819	3762206	13.0	59.1
√		√	0.874	0.816	0.911	0.792	0.914	0.805	3264008	12.1	70.3

√	√	√	0.893	0.835	0.918	0.804	0.937	0.826	3762206	13.0	58.8

Note: √ indicates that the module was used here.

Branch segmentations

To analyze the improvements of the upright branch segmentation model, we visualized some photographs of branches of different conditions, i.e., single upright branch, multiple upright branches, and with different backgrounds. The detections were shown in Fig. 13. Given that one key objective of this study was to develop an automated pruning system capable of operating in real time under field conditions, the branch segmentation model was required to achieve high accuracy as well as fast inference speed to meet the real-time demands of the robotic platform. While two-stage networks (e.g., Mask R-CNN) typically deliver higher segmentation accuracy, their complex architectures lead to slower inference speeds, making them impractical for real-time applications. Based on these considerations, YOLOv8n-seg was selected as the baseline model and further improved to achieve real-time performance most suitable for in-field robotic pruning. Comparisons were conducted with the baseline model (YOLOv8n-seg) to validate the effectiveness of the proposed improvements.

Fig. 13 showed a comparison of the segmentation using different models. Fig. 13 (a) and (b) displayed the segmentation results of the primary branches and upright branches in the absence of interfering background branches, while Fig. 13 (c) and (d) showed the segmentation results when there were interfering background branches. As seen in Fig. 13 (a), both the original model and the improved YOLOv8n-seg model were able to segment the main and upright branches. In the case of multiple upright branches being segmented, the improved model demonstrated better segmentation performance compared to the original model. The original model had poor segmentation performance at the edges of some upright branches, as well as occurrences of incorrect segmentation. In contrast, the improved model, by introducing DSConv, significantly enhanced the completeness and accuracy of upright branch segmentation., as shown in Fig. 13 (b) and 13 (c).

It was also found that when there were a large number of interfering branches in the background, the original model failed to detect some of the upright branches, as shown in Fig. 13 (d). However, the improved model, by enhancing its feature extraction capabilities, successfully achieved complete segmentation of all the upright branches. The experimental results indicated that under complex background conditions, the mask segmentation performance of the improved model was superior to that of the original model.

However, we also observed some cases of branch segmentation failure in the field experiments, as shown in Fig. 14. When upright branches were intertwined (Fig. 14(a)) or when other branches were attached to the primary branch (Fig. 14(b)), the model was prone to missed detections. In addition, some upright branches masks showed incomplete segmentation (Fig. 14(c) and (d)), which could lead to misjudgments in pruning point localization within the field of view.

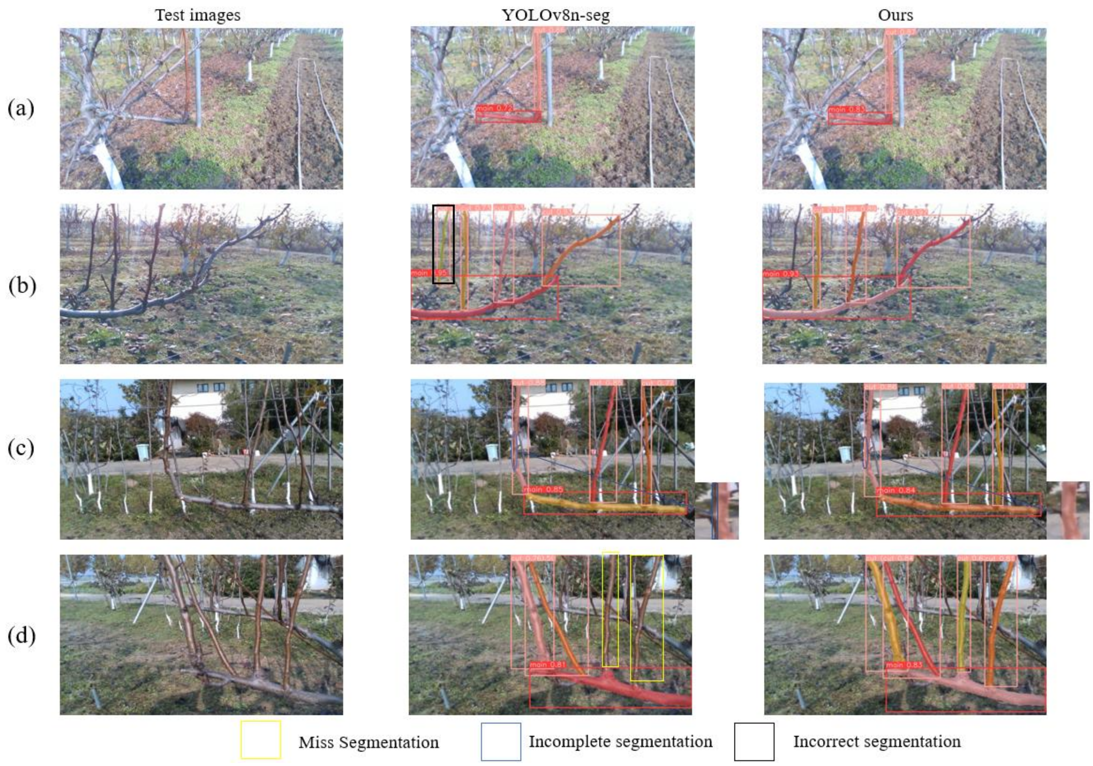

Fig. 13 Comparison of segmentation results. **(a)** Single upright branch. **(b)** Multiple upright branches with a background free of interference. **(c)** and **(d)** Multiple upright branches with interference. The numbers in the image represent the confidence of the branch segmentation

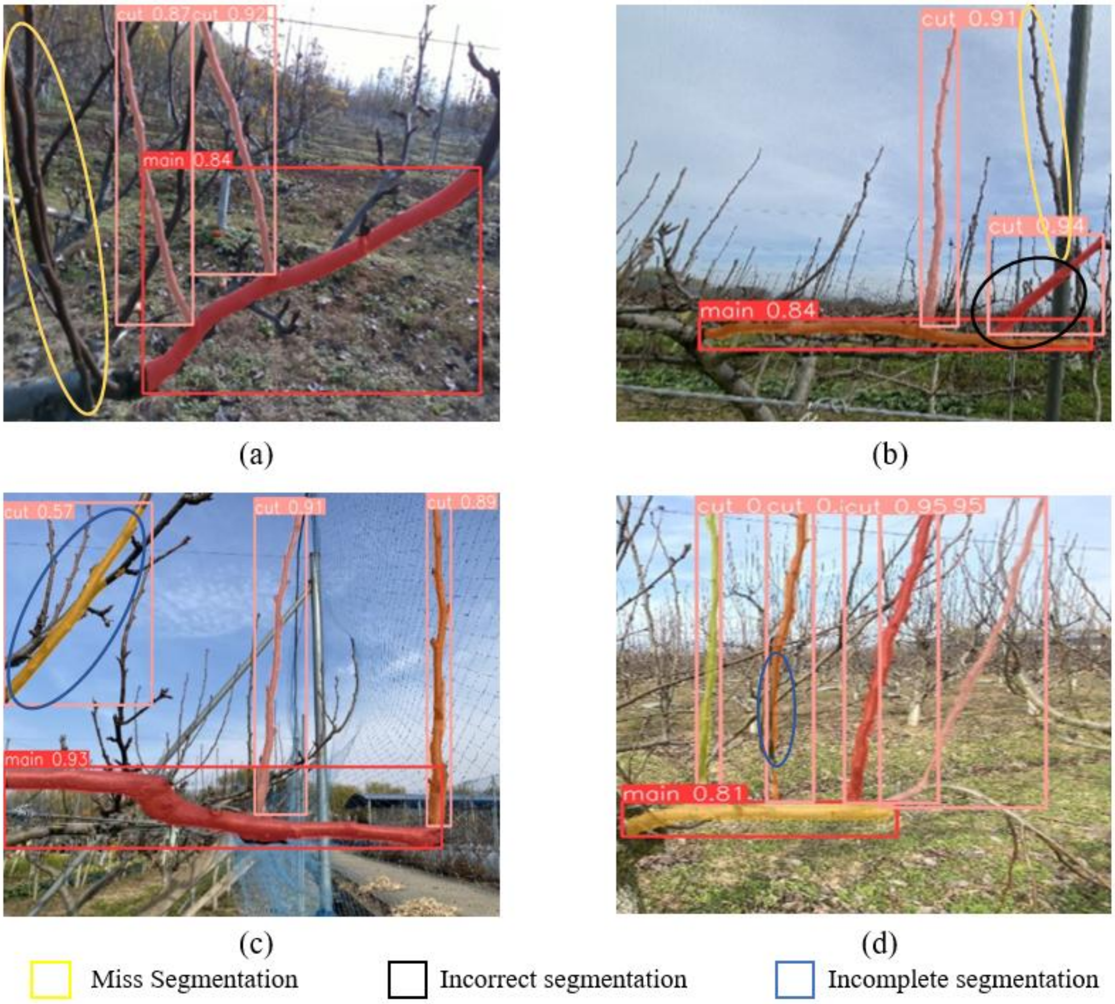

Fig. 14 Examples of unsuccessful branch segmentation. **(a)** and **(b)** Undetected uprights branches. **(c)** and **(d)** Incompletely segmented branches

Field experiments results

Real-time Trunk Localization

The experimental results of trunk detection deployed on the laptop were shown in Fig. 15. The improved model achieved an FPS of 31.6, namely the detection speed for one image was about 31.6 ms, which fully met the real-time detection requirements for trunks. We conducted field experiments and found that the system can effectively filter out the other trunks behind the target trunk in different lighting conditions.

Compared with manual measurements, the model's trunk distance localization results showed a MAE of 3.71 cm and RMSE of 3.84 cm, indicating high horizontal distance measurement accuracy, as shown in Fig. 16. The MSD of the algorism measurement of trunk center position was 1.04 cm, compared to 0.34 cm for manual measurement. The difference primarily originated from measurement errors generated by depth cameras in outdoor environments. The smaller variance observed in manual measurements (0.34 cm) demonstrated that the detected trunk center position remained relatively stable during each robotic arm rotation. These results indicated that the system could not only achieve precise detection of trunks but also enabled accurate rotation of the robotic arm.

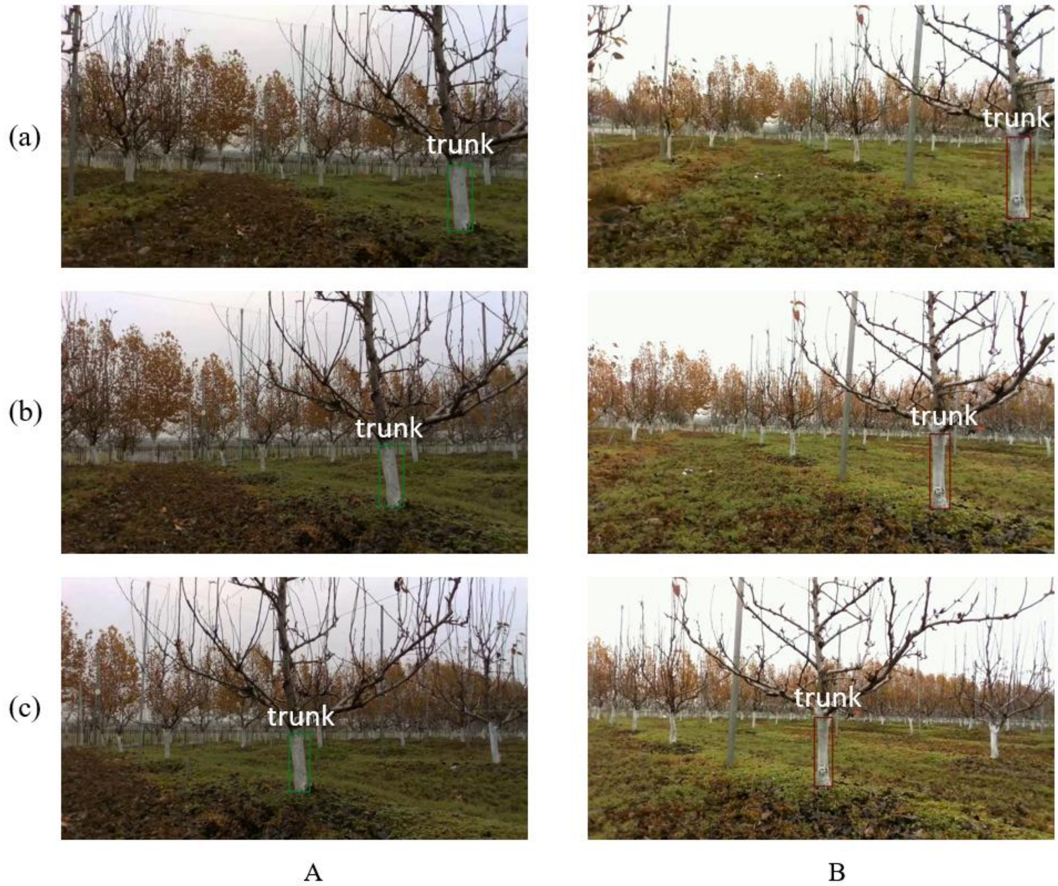

Fig. 15 Field test results of target trunk localization from different models. **A** Backlight condition. **B** Front lighting condition. (**a)** Robot initial Position. (**b)** Robot rotation. **(c)** Robot stopping position

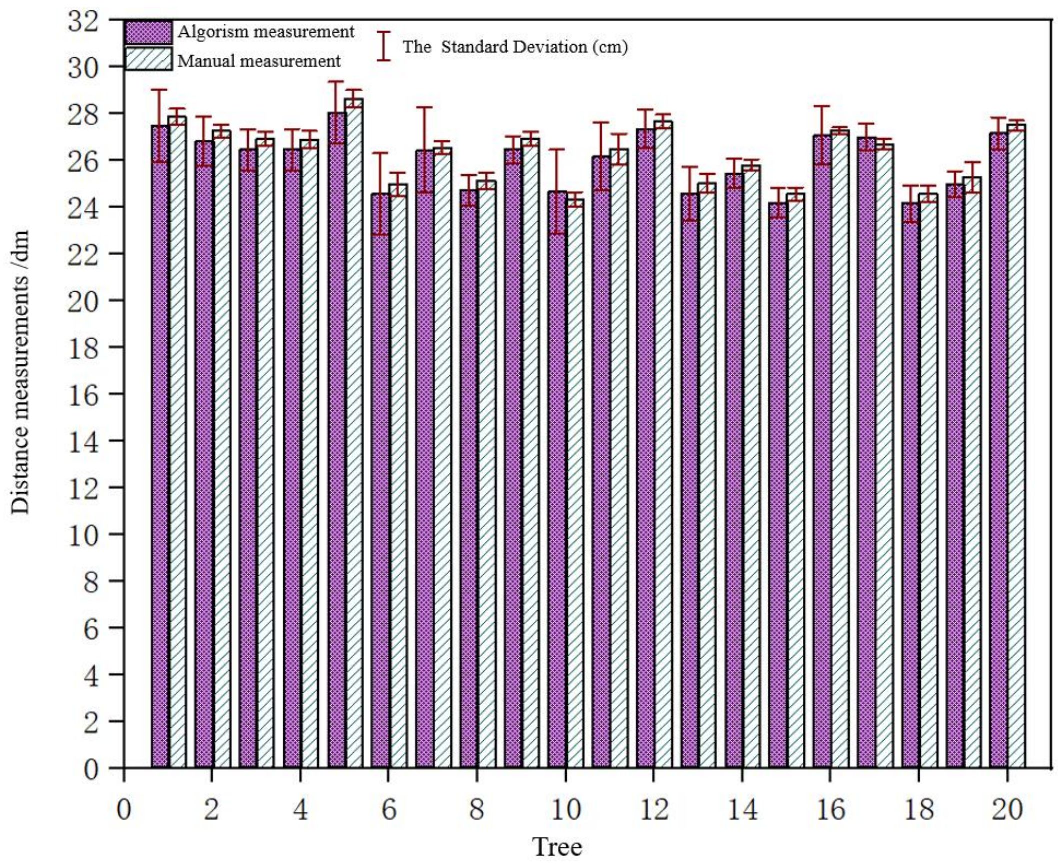

Fig. 16 Real-time trunk localization results

Characterization of upright branches

In the field experiment, we collected upright branches to be pruned within 14 fields of view and detected a total of 48 upright branches, of which 42 were correctly recognized as upright branches. The manually counted number of upright branches to be pruned was 51, resulting in a recognition accuracy of 82.3%. This indicated that the system produced 6 false positives (non-upright branches incorrectly recognized as upright branches, including cases of incomplete segmentation) and 9 false negatives (upright branches present but not detected), suggesting that the system was primarily prone to missed detections. We further evaluated the accuracy of branch parameter measurements. The algorism measurement of the pruning points and angle of upright branches were deployed on the laptop, and evaluated with manual measurement, as shown in Fig. 17. The MAE for distance from the base of the branch to the camera was 2.3 cm, and the MAE for branch angle was 2.34°. The RMSE for distance from the base of the upright branches to the camera was 2.6 cm, and for branch angle, it was 2.71°. Considering the pruning blade opening is 6 cm, a depth error within 3 cm (half of the opening) was acceptable. The R^2 values between the algorithm's measurements and the manual measurements for the distance from the base of the branch to the camera and upright brancher angle were 0.960 and 0.955, respectively. The detection FPS of the upright branches was 36.2, namely the detection speed for one image was 27.6 ms, which met the requirements for real-time branch segmentation. The experimental results demonstrated that the improved detection model was capable of performing real-time branch characterization at a high frame rate, while also providing high consistency in distance from the base of the branch to the camera and angle measurements. Accurately measuring the distance from the base of the upright branches to the camera and angle of upright branches enabled the precise determination of pruning positions and angles during the pear tree dormant pruning process, thereby meeting the requirements of the robot pruning operations.

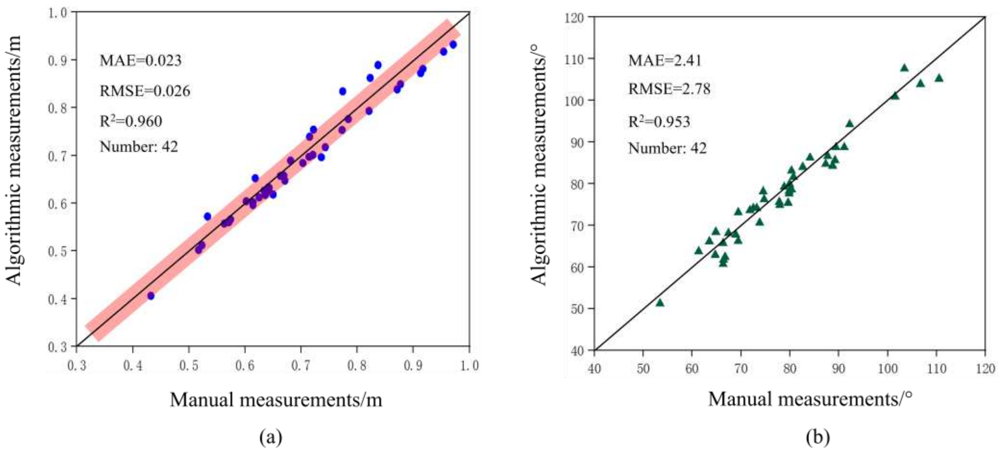

Fig. 17 Schematic diagram of distance from the base of the branch to the camera and upright branches angle measurement results. **(a)** The distance from the base of the branch measurement results. **(b)** Branch angle measurement results. The shadow area in Figure (a) represents the acceptable error range (±0.03 m)

Field pruning results

Field pruning experiments were conducted on these upright branches, as shown in Fig. 18. For each upright branch, we allowed 2 attempts to cut the branch before moving on to the next branch. For the pruning of 42 upright branches, 20 upright branches were successfully pruned (Fig. 18(a)), resulting in a success rate of 47.6%, with an average pruning time of 38 seconds per branch. However, the pruning efficiency requires further improvement. There were also cases of failure during pruning. For 15 upright branches, depth acquisition had large errors (exceeding 3 cm), which could be the main reason for the pruning failure of the robotic arm (Fig. 18(b)). For 7 upright branches, their pruning points were difficult to reach due to obstruction by the main branch or other branches. (Fig. 18(c)). In addition, there were motion errors during each operation of the robotic arm, which also had a certain impact on the pruning process. To address this issue, future work will focus on optimizing the system's localization accuracy to improve the precision and efficiency of pruning operations.

(a) (b) (c)

Fig. 18 The field pruning results. **(a)** Successful pruning. **(b)** and **(c)** Failed pruning

Discussion

The automatic pruning of dormant fruit trees has long been recognized as a key technique and significant challenge in smart orchard systems. Efficient pruning relies heavily on the accurate characterization of trees that are to be pruned (Akbar et al., 2016). To address this challenge, we developed a real-time detection and characterization system for the trunks and upright branches of pear trees oriented for automatic dormant pruning. This system optimized YOLO algorithms, improving the detection accuracy of trunks and the segmentation accuracy of upright branches in real pruning process.

In terms of trunk detection, our model achieved a detection accuracy of 90.5%, with a model size of only 4.2 MB. Field experiments revealed that, compared to the original model, the improved model increased the FPS from 22.1 to 31.6, significantly enhancing the processing efficiency and real-time performance during the automated pruning process. The MAE for trunk localization was 3.71 cm, and the RMSE was 3.84 cm, fully demonstrating the superiority of our method in trunk detection and localization. In previous studies, some relatively accurate results have been achieved in trunk detection. Su et al. (2022) proposed an improved YOLOv5s-based trunk detection method for semi-structured apple orchards. However, the trunk detection accuracy was only 86.2%, and the model size was 13.6 MB. Liu et al. (2024) proposed a trunk detection method based on LiDAR and camera fusion, achieving a trunk detection accuracy of 93.1%, but the trunk localization error was 7.5 cm. In contrast, our method not only ensures high accuracy but also significantly reduces the model size and improves trunk localization precision.

In branch segmentation, our model size was 8.3 MB. Although it was 1.7 MB larger than the original segmentation model, it improved segmentation accuracy and achieved precise measurement of the distance from the base of the branch and angle of upright branches, which was critical for pruning operations. While existing studies, such as the research of Borrenpohl et al. (2023), achieved 89% segmentation precision using Mask-RCNN, and Majeed et al. (2018) obtained similarly high segmentation precision (93%) through Convolutional Neural Networks (CNN) and Segmentation Network (SegNet), these methods rely on computationally intensive two-stage segmentation networks. In contrast, we adopted the single-stage segmentation network YOLOv8n, which maintains high segmentation precision (83.5%) while significantly improving processing speed and reducing the computational complexity of the model. In addition, simple branch segmentation alone is insufficient to meet the requirements of practical applications. Ma et al. (2021) conducted a 3D reconstruction of dormant jujube trees and used the Segmentation Prediction and Guidance Network (SPGNet) for branch segmentation. This enabled the extraction of parameters such as branch length and diameter, providing a basis for automated dormant pruning of jujube trees. Zhang et al. (2020) utilized LiDAR for 3D reconstruction of fruit trees and extracted the topology and structural information of branches to obtain their length and quantity. Although point cloud reconstruction can provide accurate 3D information of branches, it typically requires significant computational and processing time, which can affect the real-time decision-making and pruning efficiency of robots. Our study further provides precise branch parameter measurements based on segmentation. Field experiment results show that the image detection frame rate is 36.2, which meets real-time requirements well.

Despite certain advancements in the integration and application of ROS systems in fruit-picking robots, where many studies have achieved precise fruit recognition and harvesting through ROS, such as in tomato (Yung et al., 2019) and apple (Zhu et al., 2024) harvesting systems, the integration and application of fruit tree pruning robots still require further development. In contrast to fruit picking, the integration and application of pruning robots require further development. In this context, we have successfully integrated the trunk detection model and upright branch segmentation model into the ROS system, enabling precise detection and segmentation of trunks and branches. This development provides valuable technical support for the practical application of automated pear tree pruning.

Based on the previous analysis, we have further explored the sources of error in the pruning system and its applicability, as well as proposed directions for future optimization. We find that the measurement error in the depth direction of the depth camera is one of the main sources of error in the pruning experiments, with a pruning failure rate of 35.7%. This is because the accuracy of depth data is easily affected by factors such as lighting changes, camera position, and angle, which may lead to the robotic arm being unable to reach the pruning points accurately. The structure of tree branches is another important factor affecting the pruning accuracy of the robot. For example, when pruning upright branches are obstructed by the main trunk or other branches, the robot may be unable to reach the pruning points accurately. Furthermore, in orchards with high planting density, the current design and control system of the robotic arm may struggle to adapt to the narrow pruning space, thus affecting the pruning results. Additionally, this system currently focuses on pruning upright branches, and for other types of branches, the existing system may need to incorporate more training data to accommodate different branches. To improve pruning efficiency and accuracy, future work will focus on optimizing the localization accuracy of pruning points. We plan to reduce depth camera errors further through multi-sensor fusion technology. At the same time, in order to minimize the risk of improper pruning the upright scaffold branch, we will also incorporate branch diameter recognition and make judgments based on the branch diameter. In the future, we plan to test the system on more tree architectures suitable for mechanized pruning, such as UFO, espalier, and V-shaped structures.

Conclusions

The pruning of fruit trees during the dormant period requires an accurate machine vision system for efficient operation. Firstly, A dormant pear tree trunk detection and localization model based on an improved YOLOv5n algorithm was proposed in this study. The model achieved an accuracy of 90.5% in trunk detection, with an average localization error of 3.71 cm, significantly improving trunk detection precision and localization accuracy. Subsequently, an upright branch segmentation model based on the improved YOLOv8n-seg algorithm, along with a new method for measuring the pruning points and angle of upright branches, was proposed. The model achieved an accuracy of 83.5% in branch segmentation This method not only accurately captured branch parameters but also precisely located pruning position based on different pruning requirements, significantly enhancing the efficiency and specificity of pruning operations. Ultimately, the PRUNING_ROS package was developed, which integrates trunk detection and localization, upright branches segmentation, parameter measurement and pruning control in orchard environments. The field pruning success rate was 47.6%. Overall, the proposed method provided effective technical support for automatic pear tree pruning in orchard.

Acknowledgments This work was co-financed by the Major Science and Technology

Projects of Xinjiang Uygur Autonomous Region (2024A02006-3), the Jiangsu Agricultural Science and Technology Innovation Fund (No. CX(22)2025 and No. CX(23)1011), and the National Natural Science Foundation of China (No. 32001980).

Data availability The PRUNING_ROS package will be shared on GitHub after the paper is accepted. Trunk and branch data will be made available on request.

Declarations

Conflict of 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.

References

Akbar, S.A., Chattopadhyay, S., Elfiky, N.M., Kak, A., 2016. A Novel Benchmark RGBD Dataset for Dormant Apple Trees and its Application to Automatic Pruning. Proceedings of 29th ieee conference on computer vision and pattern recognition workshops, 347-354.

Borrenpohl, D., Daniel, T., 2023. Automated pruning decisions in dormant sweet cherry canopies using instance segmentation. Comput. Electron. Agric., 207, 107716.

Cuevas-Velasquez H, Gallego AJ, Tylecek R, 2020. Real-time Stereo Visual Servoing for Rose Pruning with Robotic Arm[J]. 2020 IEEE international conference on robotics and automation, 2020, 7050-7056.

Chen, Y.Y., Hou, C.J., Tang, Y., Zhuang, J.J., Lin, J.T., He, Y., Guo, Q.W., Zhong, Z.Y., Lei, H., Luo, S.M., 2020. Article Citrus Tree Segmentation from UAV Images Based on Monocular Machine Vision in a Natural Orchard Environment, Sensors, 19(24).

Chen, Z.J., Ting, D.V., Newbury, R., Chen, C., 2021. Semantic segmentation for partially occluded apple trees based on deep learning. Comput. Electron. Agric., 181: 105952.

Cao, Z., Gong, C.Z., Meng, J.J., Liu, L., Rao, Y., Hou, W.H., 2024. Orchard Vision Navigation Line Extraction Based on YOLOv8-Trunk Detection. IEEE Access, 12: 104126-104137.

Davenport, T.L., 2006. Pruning strategies to maximize tropical mango production from the time of planting to restoration of old orchards. Hortscience, 41(3), 544-548.

Dey, D., Mummert, L., Sukthankar, R., 2012. Classification of plant structures from uncalibrated image sequences. Proceedings of IEEE Workshop on Applications of Computer Vision, 329-336.

Ding, Z.J., Liu, C., Li, D.W., Yi, G.R., 2023. Deep-Sea Biological Detection Method Based on Lightweight YOLOv5n. Sens., 23(20).

do Nascimento, M.G., Fawcett, R., Prisacariu, V.A., 2019. DSConv: Efficient Convolution Operator. 2019 IEEE/CVF International Conference on Computer Vision, 5147-5156.

Gao, M., Lu, T., 2006. Image processing and analysis for autonomous grapevine pruning. IEEE International Conference on Mechatronics and Automation, 922–927.

Huang, P.C., Huang, P.K., Wang, Z.H., Wu, X., Liu, J., Zhu, L.X., 2023. Deep-Learning-Based Trunk Perception with Depth Estimation and DWA for Robust Navigation of Robotics in Orchards. Agronomy, 13(4).

Kolmanic, S., Strnad, D., Kohek, S., Benes, B., Hirst, P., Zalik, B., 2021. An algorithm for automatic dormant tree pruning. Appl. Soft Comput., 2021, 99.

Kim, E., Hong, S.J., Kim, S.Y., Lee, C.H., Kim, S., Kim, H.J., Kim, G., 2022. CNN-based object detection and growth estimation of plum fruit (Prunus mume) using RGB and depth imaging techniques. Sci. Rep., 12(1).

Li, S.C., Huang, H.P., Meng, X.Y., Wang, M.S., Li, Y., Xie, L., Distante, C., 2023. A Glove-Wearing Detection Algorithm Based on Improved YOLOv8. Sens., 23(24).

Liu, Y., Ji, J., Zhao, L.J., Feng, W., He, Q., Wang, X.K., 2024. Trunk detection method based on fusion of lidar and camera data. Journal of Southwest University, 46(02): 183-196.

Liao, H.Y., Zhou, X.C., Huang, H.Y., 2021. Detection of Lodging Landscape Trees in Typhoon Disaster based on Unmanned Aerial Vehicle Remote Sensing. Remote Sensing Technology and Application, 36(3): 533-543.

Majeed, Y., Zhang, J., Zhang, X., Fu, L.S., 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(17): 75-80.

Ma, B.J., Jiang, H.Y., 2024. Research status on visual perception system of pruning robot for dormant fruit tree. Agricultural Engineering, 14(02), 29-33.

Ma, B.J., Du, J., Wang, L., Jiang, H.Y., Zhou, M.C., 2021. Automatic branch detection of jujube trees based on 3D reconstruction for dormant pruning using the deep learning-based method. Comput. Electron. Agric., 190.

Medeiros, H., Kim, D., Sun, J.X., Seshadri, H., Akbar, S.A., Elfiky, N.M., Park, J., 2017. Modeling Dormant Fruit Trees for Agricultural Automation. J. Field Robot., 34(7): 1203-1224.

Mu, Y., Fujii, Y., Takata, D., Zheng, B.Y, Noshita, K., Honda, K., Ninomiya, S., Guo, W., 2018. Characterization of peach tree crown by using high-resolution images from an unmanned aerial vehicle, Hor. Res., 5.

Niu, W.D., Lei, X.P., Li, H., Wu, H.Q., Hu, F.S., Wen, X.X., Zheng, D.C., Song, H.Y., 2024. YOLOv8-ECFS: A lightweight model for weed species detection in soybean fields. Crop Prot., 184.

Peeters, A., Zude, M., Kathner, J., Unlu, M., Kanber, R., Hetzroni, A., Gebbers, R., Ben-Gal, A., 2015. Getis-Ord's hot- and cold-spot statistics as a basis for multivariate spatial clustering of orchard tree data. Comput. Electron. Agric., 111: 140-150.

Qiu, C., Tian, G.Z., Zhao, J.W., Liu, Q., Xie, S.J., Zheng, K., 2022. Grape Maturity Detection and Visual Pre-Positioning Based on Improved YOLOv4. Electron., 11(17).

Rai, N., Zhang, Y., Villamil, M., Howatt, K., Ostlie, M., Sun, X., 2024. Agricultural weed identification in images and videos by integrating optimized deep learning architecture on an edge computing technology. Comput. Electron. Agric., 216.

Shen, Y., Zhuang, Z.Z., Liu, H., Jiang, J.B., Ou, M.X., 2022. Fast recognition method of multi-feature trunk based on RealSense depth camera. Transactions of the Chinese Society for Agricultural Machinery, 53(04): 304-312.

Simoes, J.P., Gaspar, P.D., Assuncao, E., Mesquita, R., Simoes, M.P., 2022. Navigation system of autonomous multitask robotic rover for agricultural activities on peach orchards based on computer vision through tree trunk detection. X International Peach Symposium, 1352: 593-599.

Su, F., Zhao, Y.P., Shi, Y.X., Zhao, D., Wang, G.H., Yan, Y.F., Zu, L.L., Chang, S.Y., 2022. Tree Trunk and Obstacle Detection in Apple Orchard Based on Improved YOLOv5s Model. Agronomy,12(10).

Wu, J.T., Yang, G.J., Yang, H., Zhu, Y.H., Li, Z.H., Lei, L., Zhao, C.J., 2020. Extracting apple tree crown information from remote imagery using deep learning. Comput. Electron. Agric., 174.

Wang, X.C., Li, W.M., Wang, L., Shi, Y.Y., Wu, Y., Wang, D.Z., 2023. Method of detection-discrimination-localization for mature asparagus based on improved YOLACT++ algorithm. Transactions of the Chinese Society for Agricultural Machinery, 54(07): 259-271.

Wu, G.R., Chen, J.X., 2018. Research on the key points of fruit tree pruning technology. Agriculture and Technology, 38(24): 222.

Whiting, M.D., 2018. Precision orchard systems. In: Q. Zhang (Ed.), Automation in Tree Fruit Production. CAB International, 75–93.

Xue, Z.Y., Xu, R.J., Bai, D., Lin, H.F., 2023. YOLO-Tea: A Tea Disease Detection Model Improved by YOLOv5. For., 14(2).

Ye, X.W., MAMAT, SAWUT., Li, R.P., He, X.G., 2023. Research on fruit tree detection method with improved YOLOv5s for UAV imagery. Science of Surveying and Mapping, 48(11): 200-210.

Yung, L., Maccarana, Y., Maroni, G., Previdi, F., 2019. Partially structured robotic picking for automation of tomato transplantation. 2019 IEEE International Conference on Mechatronics (ICM), 640-645.

Zhang, F.H., W, Q., Li, H.T., Zhou, Q.Y., Tan, Z.H., Zu, X.C., Yan, X., Zhang, S.L., Ninomiya, S., Mu, Y., Tao, S.T., 2024. Study on the Optimal Leaf Area-to-Fruit Ratio of Pear Trees on the Basis of Bearing Branch Girdling and Machine Learning. Plant Phenomics, 6.

Zhang, J.Y., Wu, S.J., Liu, Y., Song, J., Yang, G.F., Yang, G., 2024. Fruit tree shaping and pruning technology. Anhui Agricultural Science Bulletin, 30(03): 39-43.

Zhang, J., He, L., Karkee, M., Zhang, Q., Zhang, X., Gao, Z.M., 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.

Zhu, F.W., Zhang, W.J., Wang, S.Y., Jiang, B., Feng, X., Zhao, Q.L., 2024. Apple-Harvesting Robot Based on the YOLOv5-RACF Model. Biomimetics, 2024, 9(8).

Zhang, C.J., Yang, G.J., Jiang, Y.Y., Xu, B., Li, X., Zhu, Y.H., Lei, L., Chen, R.Q., Dong, Z., Yang, H., 2020. Apple Tree Branch Information Extraction from Terrestrial Laser Scanning and Backpack-LiDAR, Remote Sens., 12.

Authors and Affiliations

Hao Sun [1] • Gengchen Wu [1] • Hu Xu [1] • Jiaqi Li [1] • Zhidi Zhou [1] • Shutian Tao [2,3] • Wei Guo [4] • Kaijie Qi [2] • Hao Yin [2] • Shaoling Zhang [2] • Seishi Ninomiya [1,4] • Yue Mu [1*]

*Correspondence: Yue Mu

yuemu@njau.edu.cn

[1] Academy for Advanced Interdisciplinary Studies, Collaborative Innovation Center for Modern Crop Production Co-Sponsored by Province and Ministry, Nanjing Agricultural University, Nanjing 210095, China

[2] Sanya Institute, College of Horticulture, Nanjing Agricultural University, Nanjing, Jiangsu, 210095, China

[3] College of Horticulture, Xinjiang Agricultural University, Urumqi, China

[4] Graduate School of Agricultural and Life Sciences, The University of Tokyo, 1-1-1 Midori-cho, Tokyo 188-0002, Japan

Appendix

The equations involved in this paper were as follows:

$$Z = Squeeze(U) = \frac{1}{W \times H} \sum_{i=1}^{W} \sum_{j=1}^{H} U \tag{1}$$

$$S = Excitation(Z, W) = \partial[W_2 \delta(W_1 Z)] \tag{2}$$

Where W_1 and W_2 were the weight matrices of the two fully connected layers, ∂ was the Sigmoid activation function, δ was the ReLU activation function.

$$\tilde{X} = Scale(U, S) = U \times S \tag{3}$$

$$K = \{(x-1, y-1), (x-1, y), \cdots, (x+1, y+1)\} \tag{4}$$

Where x was the horizontal coordinate of the pixel (the horizontal direction of the image), y was the vertical coordinate of the pixel (the vertical direction of the image).

$$FocalLoss = -\alpha(1-\beta)^{\gamma} lg\beta \tag{5}$$

Where β was model prediction probability, α was balancing parameter, γ was adjustable focusing parameter.

$$L_{SIoU} = L_{IoU} - (\Delta + \Omega)/2 \tag{6}$$

Where L_{SIoU} was SIoU loss value, Δ was distance loss value, Ω was shape loss value, L_{IoU} was IoU loss value.

$$\begin{bmatrix} X_{prune} \\ Y_{prune} \\ Z_{prune} \end{bmatrix} = \begin{bmatrix} (u - c_x) \cdot \frac{D}{f_x} \\ (v - c_y) \cdot \frac{D}{f_y} \\ D \end{bmatrix} \tag{7}$$

Where (u, v) were the pixel coordinates in the pixel coordinate system, c_x and c_y were the principal point coordinates in the image coordinate system, f_x and f_y were the focal lengths in the image coordinate system, D was the depth value of the point from the depth map.

$$\varepsilon = \sum_{n=1}^{k} \sqrt{\left(X_{c(n)} - X_{c(n-1)}\right)^2 + \left(Y_{c(n)} - Y_{c(n-1)}\right)^2 + \left(Z_{c(n)} - Z_{c(n-1)}\right)^2} \tag{8}$$

Where ($X_{c(n)}$, $Y_{c(n)}$, $Z_{c(n)}$) were coordinates of each point in the camera coordinate system, k was number of points after skeletonization.

$$\alpha = arctan\left(\frac{\sqrt{(v_\varepsilon - v_1)^2}}{\sqrt{(u_\varepsilon - u_1)^2}}\right) \ (u_\varepsilon > u_1) \tag{9}$$

$$\alpha = 180^{\circ} - arctan\left(\frac{\sqrt{(v_{\varepsilon}-v_1)^2}}{\sqrt{(u_{\varepsilon}-u_1)^2}}\right)\ (u_{\varepsilon}<u_1) \tag{10}$$

$$\alpha = 90^{\circ}\ (u_{\varepsilon}=u_1) \tag{11}$$

Where (u_1, v_1) were the coordinates of the base point of the skeleton of the upright branches, $(u_{\varepsilon}, v_{\varepsilon})$ were the coordinates of the measurement point of the skeleton.

$$Precision = \frac{TP}{TP+FP} \times 100\% \tag{12}$$

$$Recall = \frac{TP}{TP+FN} \times 100\% \tag{13}$$

$$mAP = \frac{1}{C}\sum_{i=1}^{N} P(i)\Delta R(i) \tag{14}$$

Where TP was True Positive, FP was False Positive, FN was False Negative.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - y_i'| \tag{15}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - y_i')^2} \tag{16}$$

$$MSD = \frac{1}{N}\sum_{i=1}^{N}\sqrt{\frac{\sum_{j=1}^{3}\left(y_i^j - y_i^{j\prime}\right)^2}{3}} \tag{17}$$

Where y_i was manually measured values, y_i' was algorithmically measured values, N was number of measurements. j: Number of repetitions of the experiment.