Computers and Electronics in Agriculture 252 (2026) 112125

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

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

# Occlusion-aware pinching point positioning using shoot morphology for robotic summer pruning of kiwifruit

Bryan Gilbert Murengami [a], Leilei He [a], Shuolin Kong [a], Lamin L. Janneh [a], Xiaojuan Liu [a], Wenhao Guo [a], Rui Li [a,*], Jinyong Chen [d], Spyros Fountas [e], Hin Lim Shen [f], Longsheng Fu [a,b,c,**]

[a] College of Mechanical and Electronic Engineering, Northwest A&F University, Yangling, Shaanxi 712100, China
[b] Key Laboratory of Agricultural Internet of Things, Ministry of Agriculture and Rural Affairs, Yangling, Shaanxi 712100, China
[c] Shaanxi Key Laboratory of Agricultural Information Perception and Intelligent Service, Yangling, Shaanxi 712100, China
[d] Zhengzhou Fruit Research Institute, Chinese Academy of Agricultural Sciences, Zhengzhou, Henan 450009, China
[e] Agricultural University of Athens, Athens 11855, Greece
[f] School of Engineering, University of Waikato, Hamilton 3240, New Zealand

ARTICLE INFO

*Keywords:*
Summer pruning
Pinching
Instance segmentation
Shoot morphology
Point positioning

ABSTRACT

Pinching is a summer pruning practice in kiwifruit cultivation that involves permanently removing the shoot's growing tip after the distal bud. Automating this practice requires accurate positioning of pinching points, which remains challenging due to dense canopies and variability in shoot type. This study proposed a novel automated method for positioning pinching points by combining YOLO11s-seg for segmenting branches and shoots with a morphology-guided kiwifruit pinching point positioning algorithm (KPPPA). KPPPA integrated two sequential sub-algorithms: the direction-aware mask mending (MaskMend) reconstructed the occlusion-induced fragmented mask of shoots by estimating the local search direction for candidate reconnection extreme endpoints; then, proportion-based point locating (PPL) utilized the reconstructed mask for computing shoot geometric features to distinguish non-terminating from terminating shoots and ultimately position optimal pinching points. Experimental results show that YOLO11s-seg model achieved a mean average precision of 91.7% for detection and 61.1% for mask segmentation, while KPPPA positioned pinching points obtaining success rate of 87.0% and Average Euclidean error of 5.8 pixels with Mean absolute errors of 3.6 and 2.7 pixels along x, and y-axis, respectively. The findings of the study show that the proposed method can effectively position pinching points for kiwifruit in summer at a minimal computation speed of 136.6 ms per frame, making it feasible for real-time robotic pinching operation in kiwifruit orchards.

## 1. Introduction

Summer pruning in kiwifruit cultivation plays a pivotal role in enhancing fruit size and quality. Due to the vigorous growth of kiwifruit vines, intense competition arises among vegetative organs for resources (Patiyal et al., 2024). To manage this competition, summer pruning is essential (Liao et al., 2020; Zhang et al., 2024). Among summer pruning practices, pinching involves permanent removal of the shoot's growing tip. This practice effectively redirects nutrients towards fruit development, improving dry matter content, sugar levels, and firmness (Mhetre et al., 2025). Traditionally, skilled technicians manually determine optimal pinching point by cognitively applying a fruit-to-leaf ratio beyond the distal bud (Akbaş and Özcan, 2020). However, this manual process is labor-intensive, inconsistent due to operator skill variability, and further constrained by ongoing rural labor shortages (Li et al., 2025a; Zhang et al., 2025). Therefore, the development of automated pinching robots is a promising solution. Consequently, the design of an automated pinching point positioning method is crucial to realize robotic summer pruning.

An accurate analysis of kiwifruit vine architecture enables precise annotation of key morphological structures, forming the basis for identifying pinching points. Mature vines consist of a trunk, two leaders

* Corresponding author.
** Corresponding author at: College of Mechanical and Electronic Engineering, Northwest A&F University, Yangling, Shaanxi 712100, China.
*E-mail addresses:* ruili1216@nwafu.edu.cn (R. Li), fulsh@nwafu.edu.cn (L. Fu).

https://doi.org/10.1016/j.compag.2026.112125
Received 12 September 2025; Received in revised form 11 May 2026; Accepted 26 June 2026
Available online 2 July 2026
0168-1699/© 2026 Elsevier B.V. All rights are reserved, including those for text and data mining, AI training, and similar technologies.

trained along a trellis, various lateral branches with shoots (Tian et al., 2025). In spring, shoots emerge from axillary buds and are categorized as either terminating (fewer than 9 nodes) or non-terminating (more than 18 nodes) each with distinct growth patterns (Liao et al., 2020; Richardson et al., 2021). This distinction is critical for pruning decisions. Terminating shoots naturally cease growth and are generally retained as future fruiting branches. In contrast, non-terminating shoots grow vigorously, increasing canopy density, reducing light penetration, and intensifying resource competition among fruits and shoots (Salinero et al., 2009; Zhang et al., 2024). As a result, they are the target for summer pinching, where pinching points are identified beyond the last fruiting node using a fruit-to-leaf ratio of 1: 3 to 1: 6 (Akbaş and Özcan, 2020). However, manual identification of these precise pinching locations is difficult and time-consuming, especially within the complex architecture of a dense canopy. Despite the complexity of summer canopies, deep learning networks can effectively identify key morphological structures for accurate positioning of pinching points.

Deep learning convolutional neural networks have proven effective for automatic plant structure identification in complex orchards. DeepLabV3+ network was used to segment kiwifruit branches, calyxes and wires, achieving a mean intersection over union (mIoU) of 0.69 in kiwifruit harvesting (Song et al., 2021). In robotic pollination, Li et al. (2022) used YOLOv3 and YOLOv4 for kiwifruit flower and bud detection, achieving mAP of 80.9% and 91.4%, respectively. Building on the deep learning networks, some studies focused on positioning pruning points. Tong et al. (2023) achieved a positioning accuracy of 87.2% for apple dormant pruning points using SOLOv2, while Chen et al. (2024b) used with PSPNet and YOLOv5 to position pruning points in dormant grapevines, achieving an accuracy of 82.3%. These studies largely focused on positioning pruning points for winter pruning, where minimal foliage and simpler branch structures make target identification less complex. In contrast, kiwifruit in summer exhibit dense canopies with occluded overlapping branches and shoots, limiting automated pruning. Therefore, summer pruning is a promising research direction.

Non-terminating shoots of kiwifruits grow vigorously, curling and twisting, creating occlusion that hinders visibility of the entire shoot. Previous methods of handling occlusion such as amodal segmentation (Gené-Mola et al., 2023; Kim et al., 2023), active sensing (Sun et al., 2023, 2024), shape fitting (Kok and Chen, 2024; Liang et al., 2024) and regression-based approaches (Chen et al., 2023, 2024a), are ineffective for pinching point positioning because of the vigorous growth and the curling, twisting nature of the target shoot within the canopy. To overcome this challenge, recent work has combined deep learning with positioning algorithms that leverage branch and shoot morphologies to identify target points. Among the research, Wang et al. (2024a) used Litchi-YOSO and a branch morphology reconstruction algorithm to position picking points in litchi, achieving a success rate of 91.5%. Similarly, a morphology-guided grape picking point positioning algorithm reached an accuracy of 91.0% (Wang et al., 2024b), while Li et al. (2024) used YOLOv8 and mango morphology traits for picking point positioning, achieving success rate of 92.0%. Despite these theoretical foundations, kiwifruit shoots curl and twist during vigorous growth presenting a unique challenge that has not been investigated for summer pruning.

The overall goal of this study was to develop a novel automated pinching point positioning method by combining YOLO11 with a morphology-guided positioning algorithm. The lightweight module of YOLO11 model was employed to segment branches and shoots from RGB images of kiwifruit canopy. The segmentation output was passed to a kiwifruit pinching point positioning algorithm (KPPPA) developed to overcome occlusion and position pinching points. KPPPA leveraged two sub-algorithms: direction-aware mask mending (MaskMend), which reconstructed fragmented shoot masks and proportion-based point locating (PPL) which identified the non-terminating shoots and positioned the optimal pinching point. The remainder of this paper is organized as follows: Section 2 describes the image processing and specific method. Section 3 presents and discusses the results, and Section 4 outlines the conclusions and future research.

## 2. Materials and methods

This study presents a method for positioning pinching points in kiwifruit canopies by combining deep learning with shoot morphology-guided image analysis. The two-stage approach shown in Fig. 1 comprises model training and a positioning algorithm. A kiwifruit dataset was annotated and split for training and evaluation. Then, an instance segmentation model was trained in the first stage. The outputs of the model were used in the second stage by the positioning algorithm, which integrates two sub-algorithms to identify pinching points. Further details are provided in subsequent sections.

### 2.1. Kiwifruit canopy image acquisition

The canopy images of the 'Xuxiang' kiwifruit variety were collected during early summer seasons of 2023, 2024, and 2025. Images were acquired at the International Kiwifruit Innovation Orchard in Yangling, Shaanxi Province, China (34°17′N, 108°2′E; approximately 504 m above sea level). During this period, the kiwifruit vines were in full foliage which frequently resulted in occlusion of branches and shoots. The ZED 2i binocular and RealSense D435 RGB-D cameras were used for acquiring RGB images of the kiwifruit canopy. Both cameras were positioned parallel to the canopy facing upward at distances of 110 and 140 cm below the canopy. This setup was crucial as it enabled capturing far-view and close-view images. Far-view images ensured that the entire canopy was within the field of view, while the close-view images provided high resolution necessary for effective model training. Under varying lighting conditions during different times of the day, the binocular camera captured left and right images totaling 74 close-view and 96 far-view groups; however, only the left images were retained. The RealSense D435 captured an additional 187 close-view and 49 far-view images. In total, 406 RGB images (1920 × 1080 pixels) were acquired in 2023 (162 images), 2024 (99 images), and 2025 (145 images). The sample images are shown in Fig. 2. The acquired kiwifruit canopy images are publicly available at https://github.com/Smart-Ag/Kiwifruit_pinching_images.

### 2.2. Image annotation and data preprocessing

The initial dataset constituted the close-view and far-view images captured by ZED 2i and RealSense D435. Then it was split into two sets with 328 images used for training and evaluating the YOLO11 model, while an additional 89 images reserved exclusively for evaluating the pinching point positioning algorithm. The evaluation dataset was further partitioned into multi-scenario subsets based on three key conditions, canopy density, weather, and camera-to-canopy distance comprising, complex (57 images), simple (32 images), cloudy (55 images), sunny (34 images), far (35 images), and near (54 images) subsets. This partitioning was essential for evaluating the method's performance and feasibility in outdoor orchard environments. The images within the network training dataset were manually annotated by drawing polygons while the positioning dataset was annotated with points on target structures using the Labelme tool (http://labelme.csail.mit.edu). In order to easily position the pinching points, there was a need to precisely annotate relevant structures. Based on this requirement, branches (LB) and shoots (SH) classes were labeled. Since summer pinching targets non-terminating shoots, to distinguish them from terminating ones, extraction of geometric features like shoot length which is defined by the distance between the shoot basal and apex locations was required. Therefore, shoot polygons were allowed to overlap with branch polygons during annotation reflecting their natural growth from branches as shown by the black solid box in Fig. 3. Furthermore, in densely occluded canopies spatial context was preserved by annotating over the occluding

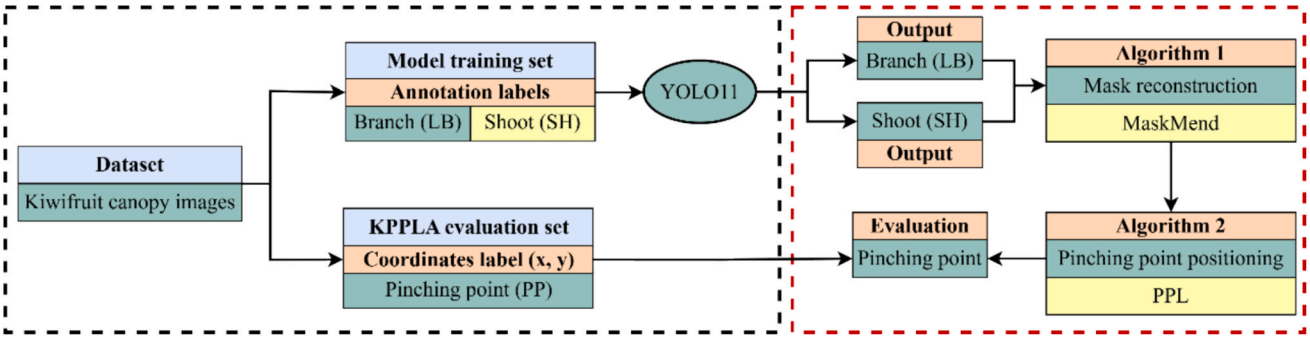



**Fig. 1.** Overview of the proposed method. Model training stage is in the black dotted box, while positioning stage is in the red dotted box.



(a) (b)

**Fig. 2.** The acquired kiwifruit canopy image samples with (a) far-view and (b) close-view.

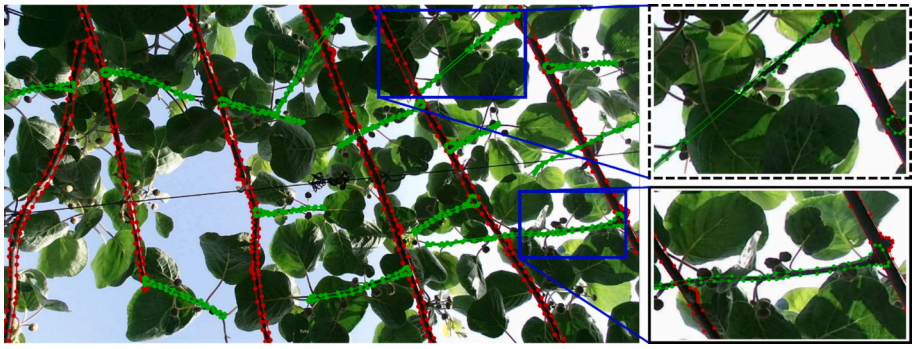


**Fig. 3.** Annotation examples showing canopy-level polygons, branch (red) and shoot (green). The black dotted box shows polygons drawn over the occluding leaf, while the black solid box denotes the branch-shoot overlap annotation relationship.

objects, as shown by the black dotted box in Fig. 3. For the evaluation set, ground truth pinching points coordinates ($x$, $y$) were annotated on non-terminating shoots based on the leaf-to-fruit ratio, thus leaving 3 ~ 6 leaves after the last fruiting node. To assess annotation consistency, a random subset of 34 images was independently re-annotated by a second annotator following the same technique. Inter-annotator agreement, measured as mean pixel-level intersection over union (Kok et al., 2023) between matched polygon sets, was 97.4% for branches and 91.6% for shoots. The lower agreement on shoots reflects the difficulty of outlining occluded, curling and twisting morphologies.

After the initial annotation, the datasets were enhanced to improve model robustness and prevent overfitting. Data augmentation techniques were applied, such as translation, rotation, brightness and contrast adjustments, local mosaic generation, Gaussian blur, and filtering. This expanded the network training dataset to 3,621 images and then it was split into training, validation, and testing sets using a ratio of 8: 1: 1. The pinching point evaluation dataset was also increased to 468 images for comprehensive assessment of the positioning algorithm performance.

### 2.3. Network selection

The YOLO (You Only Look Once) algorithm is widely recognized as one of the leading methods in computer vision due to its balance of high speed and accuracy (Zhang et al., 2025). In recent years, its adoption has expanded into agricultural domain, where rapid and accurate detection is essential for tasks such as monitoring, harvesting and yield estimation (Shi et al., 2025; Tian et al., 2025). Underpinning this expansion is the innovation behind architectural refinements. Among the iterations, YOLOv8, introduced a decoupled detection head separating classification and regression branches, mitigating gradient interference between localization and recognition tasks (Sapkota and Karkee, 2025). This was aided by an anchor-free prediction strategy which eliminated predefined anchor clustering improving generalization. The architecture's C2f backbone with SPPF further reduced parameter counts while preserving receptive-field richness, and instance segmentation was added (Yu et al., 2024). YOLOv9-GELAN introduced generalized efficient layer aggregation network (GELAN) and programmable gradient information (PGI). GELAN replaced conventional CSP-based aggregation with a flexible multi-branch topology that preserves gradient flow across arbitrary

depths without incurring proportional parameter overhead (Sapkota et al., 2025). PGI addressed the information bottleneck problem in deep networks by maintaining an auxiliary reversible branch during training, ensuring that complete gradient signals reach all layers and are not diluted by successive transformations.

The YOLO11 iteration replaced C2f blocks with compact C3k2 CSP and introduced the C2PSA spatial attention module, yielding improved feature selectivity. Its hybrid task-aware label assignment jointly tuned loss weighting across classification, localization, and auxiliary tasks, improving recognition in crowded scenes and small-objects (Li et al., 2025b; Sapkota and Karkee, 2025). The most recent iteration YOLO26 removed distribution focal loss (DFL) for lighter bounding-box parameterization and native non-maximum suppression (NMS) end-to-end inference, eliminating post-processing latency entirely. This was complemented by progressive loss balancing (ProgLoss), small-target-aware label assignment (STAL), and the MuSGD optimizer further addressing limitations around small and occluded instances (Sapkota and Karkee, 2025).

Unlike convolution-based YOLO models, the real-time detection transformer (RT-DETR) adopts a transformer-based encoder-decoder architecture. It has a hybrid encoder that combines intra-scale feature interaction via attention mechanisms with cross-scale feature fusion, enabling rich contextual modelling without the anchor-based priors that constrain YOLO-family models (Zhou and Wei, 2025). Coupled with an IoU-aware query selection mechanism that initializes decoder object queries from the most discriminative encoder features. These four deep learning architectures YOLOv9-GELAN, YOLO11, YOLO26, and RT-DETR were compared to obtain the optimal model configuration for summer canopy instance segmentation in this study.

The experiments were conducted on a desktop computer running a 64-bit Windows 10 system, equipped with an Intel Xeon Gold 5128 32-core processor (2.3 GHz), dual Nvidia GeForce GTX 4090 24G GPUs, and 128 GB of memory. The software environment was built on the PyTorch framework (version 2.11), along with CUDA 12.1, cuDNN 8.9.7, Python 3.10, and OpenCV 4.13. The deep learning models were employed to train kiwifruit branch and shoot instance segmentation. Training used an input size of 640 × 640 pixels and a batch size of 4, with momentum set to 0.8 and weight decay to 0.0005. The initial learning rate was set to 0.01, and the training epochs were set at 300 for analyzing the model training process. For RT-DETR a batch size of 1, a learning rate of 0.0001, an image input size of 648 × 648 and an epoch size of 50 were used during training.

### 2.4. Kiwifruit pinching point positioning algorithm

To obtain precise pinching points, a shoot morphology-guided

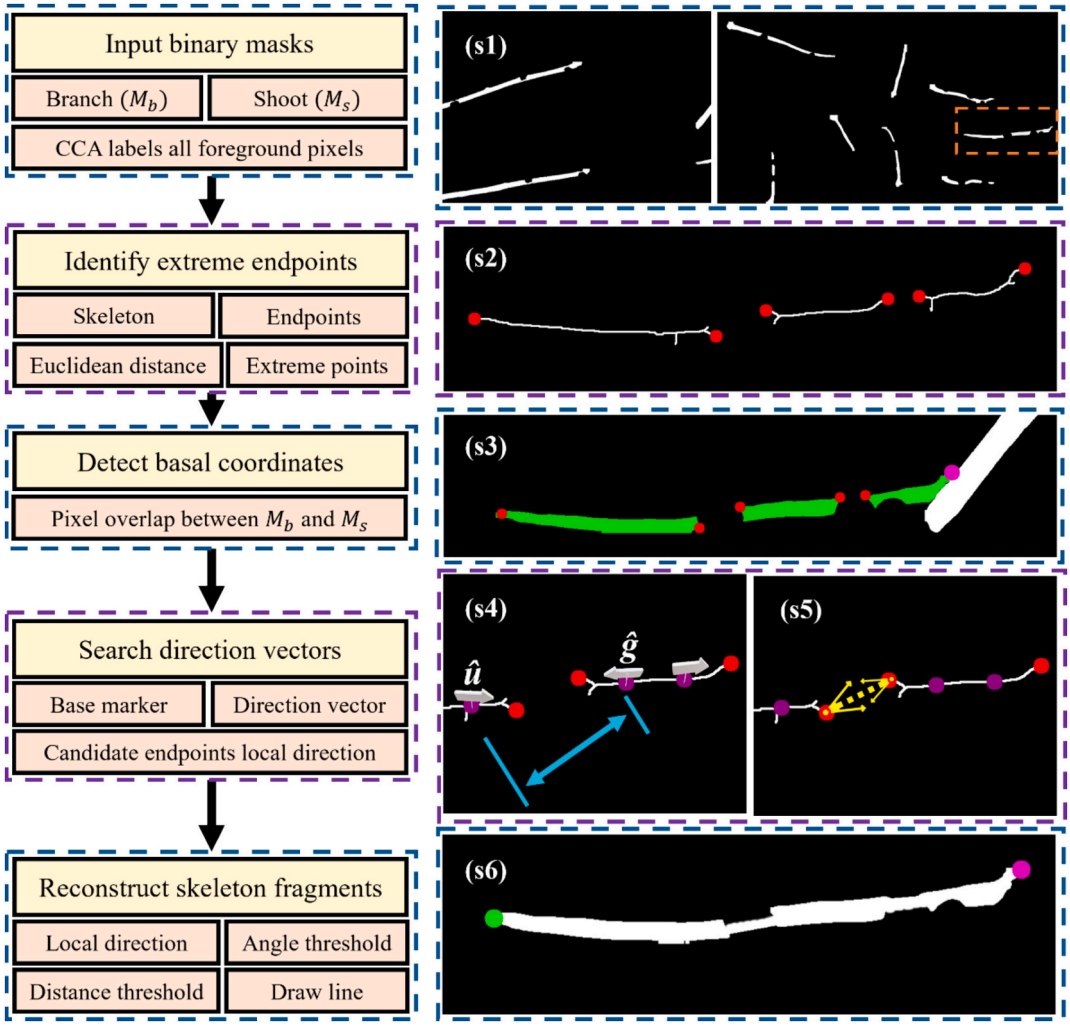



**Fig. 4.** Reconnection steps of fragmented shoots by the MaskMend algorithm. Extreme endpoints (red) define ends of shoot fragments. The basal coordinate (magenta) denotes shoot origin. Base coordinates (purple) help define local search direction vectors for reconnection candidates (white arrow). Angle between local base vector ($\hat{u}$) and candidate vector ($\hat{g}$) with distance threshold (blue arrow) set criteria for reconnections. Reconstructed shoot mask (white) with basal (magenta) and apex (green) locations are outputs.

kiwifruit pinching point positioning algorithm (KPPPA) was developed. The algorithm integrated two sub-algorithms, that sequentially processed the output of the instance segmentation model . First, the direction-aware mask mending (MaskMend) addressed occlusions by reconstructing shoot binary masks. Then the reconstructed masks were passed to proportion-based point locating (PPL) algorithm, which distinguished non-terminating from the terminating shoots and positioned the optimal pinching points.

#### 2.4.1. Direction-aware mask mending algorithm

Dense summer canopy growth in kiwifruit orchards complicates accurate shoot segmentation due to frequent occlusions, which break shoot continuity in segmented masks and can lead to incorrect positioning of pinching points. To address this, MaskMend was developed to reconnect broken shoot masks while preserving their structural orientation, this is achieved in five steps as shown in Fig. 4. Binarized branch (*Mb*) and shoot (*Ms*) masks from the segmentation model are passed as input to the algorithm as shown in Fig. 4(s1). Due to fragments, OpenCV's connected components analysis (CCA) (He et al., 2017), identified and labeled each group of foreground pixels and assigned a unique identifier to the fragments enabling pairwise analysis of branches and shoots. The labeled mask components are eroded, and skeletonized to a thin central line. This removed branching artifacts while improving computation speed for obtaining endpoints. The mask skeleton endpoints are detected using a $3 \times 3$ convolutional kernel that counted pixel connectivity within the neighborhood. However, due to shoot morphological complexity, a single skeleton often has multiple endpoints which affect reconnections. Therefore, the algorithm selects two extreme endpoints of each mask skeleton using Euclidean distance as shown in Fig. 4(s2). The basal points ($b_p$) are the extreme endpoints that represent the intersection point of branches and shoots. They are identified by searching for shared pixel ($x$, $y$) between $Ms_j$ and $Mb_i$ using the Eq. (1), where $Mb_i(x, y)$ is the $i^{th}$ branch mask at pixels ($x$, $y$), and $Ms_j(x, y)$ is the $j^{th}$ shoot mask at pixel ($x$, $y$). The basal coordinates (magenta) shown in Fig. 4(s3) are excluded to prevent erroneous reconnections.

The algorithm estimated the adjacent extreme endpoints as candidates for reconnection. The reconnection is achieved by first setting a base marker coordinate offset of 40 pixels inward along the skeleton from each extreme endpoint as shown in Fig. 4(s4). From this base marker, a normalized direction vector ($\hat{u}$) is computed using Eq. (2) toward the extreme point, defining a local search direction shown by white arrow in Fig. 4(s4). Extreme endpoints within this search direction are considered as reconnection candidates, while those on non-adjacent fragments are excluded. For each candidate within a set search distance threshold, its corresponding direction vector ($\hat{g}$) is calculated. The angle between the local direction vector ($\hat{u}$) of the base extreme endpoint and the local direction vector ($\hat{g}$) of each candidate is then calculated using the dot-product shown by Eq. (3). If this angle is below a set angle threshold, the two extreme endpoints are considered to share the same orientation (i.e., belonging to the same shoot as shown by yellow arrows in Fig. 4(s5)) and thus a line (yellow) is drawn to reconnect them. This reconnection is done for all the fragmented shoot masks. Finally, the farthest extreme endpoint (green) from the basal coordinate (magenta) is designated as the apex coordinate. The reconstructed shoot skeleton is dilated, and merged with the original fragment shoot mask. Ultimately, the output of MaskMend is the fully reconstructed shoot mask with basal and apex coordinates as shown in Fig. 4(s6).

$$b_p = \sum_{x,y} Mb_i(x,y) \cdot Ms_j(x,y) > 0 \tag{1}$$

$$\hat{u} = \frac{u}{\|u\|} \tag{2}$$

$$cos\theta = \hat{u} \cdot \hat{g} \tag{3}$$

#### 2.4.2. Proportion-based point locating algorithm

A crucial step in positioning pinching points is identifying non-terminating shoots within the complex kiwifruit canopy. Generally, the reconstructed shoot masks include both terminating and non-terminating shoots; however, only non-terminating shoots require pinching. The PPL algorithm was developed to identify non-terminating shoots from canopy images and position the pinching point in four steps, as shown in Fig. 5. To distinguish the non-terminating shoots from the terminating ones, the reconstructed shoot mask and its geometric features, namely the basal and apex points obtained from MaskMend, are used to compute the length of the shoot. Initially, the shoot mask shown in Fig. 5(t1) is reduced to a thin mask skeleton. Subsequently, the Dijkstra algorithm (Li et al., 2024) is used to determine the length ($L$) and the shortest path between the basal and apex points of the shoot skeleton as shown in Fig. 5(t2). If the length is above a set threshold ($T$) of 300 pixels, the shoot is classified as non-terminating (white), as shown by Eq. (4) and in Fig. 5(t3). The shortest path of the non-terminating shoot served as the basis for determining the pinching point. Specifically, a coordinate ($P$) corresponding to a fixed proportion ($\alpha$) of 82% of the total short path length ($\gamma$) from basal to apex of the shoot is identified as the pinching point as shown in Eq. (5) and in Fig. 5 (t4).

$$Class(S) = \begin{cases} \text{Teminating}, & L \leqslant T \\ \text{Non - terminating}, & L \geqslant T \end{cases} \tag{4}$$

$$P = \gamma(\alpha \cdot L),\ \alpha = 0.82 \tag{5}$$

### 2.5. Evaluation metrics

The instance segmentation model's performance was evaluated using standard metrics. The model's accuracy was assessed using *average precision* (*AP*) for each object class and *mean average precision* (*mAP*) across all classes. *AP* detailed by Eq. (6) was calculated based on *precision* and *recall* values, reflecting the area under the precision-recall curve. This value indicates how well YOLO11s-seg model can detect, localize and segment specific targets. *mAP* shown by Eq. (7) averages the *AP* scores of all ($k$) classes, which is a summary of the model's overall performance.

$$AP = \int_0^1 P(R)dR \tag{6}$$

$$mAP = \frac{1}{k}\sum_{i=1}^{k} AP \tag{7}$$

The performance of KPPPA was evaluated using average speed as well as spatial and directional accuracy metrics. Average speed reflects how quickly the algorithm can position pinching points per image, which is critical for real-time applications. Spatial and directional accuracy were assessed using Average Euclidean error ($E_{avg}$) and bidirectional Mean absolute errors along the x-axis ($MAE_x$) and y-axis ($MAE_y$). To accurately evaluate these metrics, the Hungarian algorithm (Keysers et al., 2004) was applied to match each predicted point with its closest ground truth counterpart. $E_{avg}$ calculated by Eq. (8) measures overall spatial accuracy by calculating average distance in pixels between predicted point ($x_{pi}$, $y_{pi}$) and its matched ground truth ($x_{gti}$, $y_{gti}$), over $N$ matched pairs. $MAE_x$ and $MAE_y$ calculated by Eqs. (9) and (10) respectively provide insight into the model's directional performance, highlighting any horizontal or vertical biases. This directional insight is critical in the context of robotic pinching, where axis-specific precision significantly impacts success in orchard operations.

$$E_{avg} = \frac{1}{N}\sum_{i=1}^{N}\sqrt{(x_{pi} - x_{gti})^2 + (y_{pi} - y_{gti})^2} \tag{8}$$

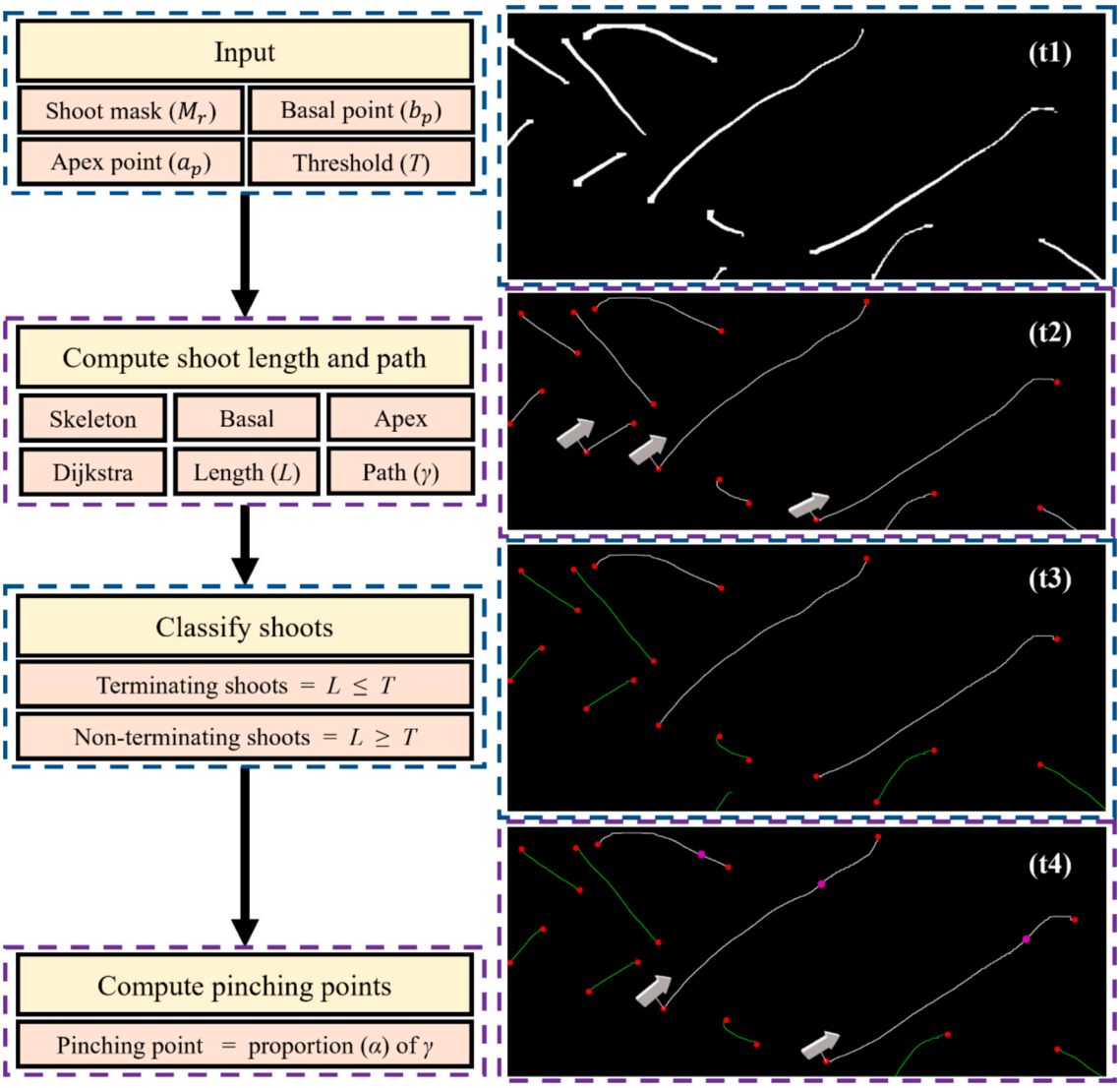



**Fig. 5.** Processing steps of the PPL algorithm for shoot classification and pinching point positioning. Reconstructed shoot mask (white) with basal and apex coordinates (red), are inputs. Green skeletons represent terminating shoots while pinching points (magenta) are placed on white skeletons which denotes non-terminating shoots. White arrows indicate direction of computation.

$$MAE_x = \frac{1}{N}\sum_{i=1}^{N}\left|x_{pi} - x_{gti}\right| \tag{9}$$

$$MAE_y = \frac{1}{N}\sum_{i=1}^{N}\left|y_{pi} - y_{gti}\right| \tag{10}$$

## 3. Results and discussion

### 3.1. Selection of optimal instance segmentation model configuration

YOLO11 and YOLO26 are suitable for kiwifruit canopy instance segmentation, as shown in Table 1. Overall, YOLO-based models consistently outperform RT-DETR variants in both accuracy and inference efficiency. YOLO26 achieves the highest mask *mAP*@50–95 of 72.3% (x variant), followed by YOLOv9-GELAN and YOLO11, while YOLOv9-GELAN attains the highest detection *mAP* of 96.1%. In contrast, RT-DETR yields lower mask and box *mAP* values of 56.4% and 88.7% respectively, alongside significantly higher inference times. Across all YOLO models, accuracy improved with scale; however, gains diminished beyond medium-sized variants, with mask *mAP* plateauing around 67.3 ~ 70.7%, while inference time continued to increase.

These trends reflect key architectural differences. YOLO-based models employ convolutional backbones with efficient feature pyramid representations, enabling strong local feature extraction and rapid inference. The improved performance of YOLO26 over YOLO11 suggests that refinements in feature fusion and segmentation heads contribute meaningfully to mask prediction quality (Sapkota and Karkee, 2025). YOLOv9-GELAN similarly benefits from enhanced gradient flow through its layer aggregation design, supporting competitive detection accuracy at moderate computational cost (Sapkota et al., 2025). RT-DETR, by contrast, employs a transformer-based architecture that models global context through attention mechanisms. While effective in scenes requiring long-range dependencies, this introduces substantial computational cost that is less suited to kiwifruit summer canopy. Its reduced training duration of 50 epochs likely further limited convergence, compounding the inference inefficiencies inherent to its design.

The results indicate that medium-scale YOLO models offer the most favorable balance between accuracy and efficiency. However, smaller variants, offer further reductions in computational demand with only modest decreases in accuracy, and are therefore preferable for outdoor orchard deployment. Although YOLO26s-seg achieves a 4.7% higher mask *mAP*, YOLO11s-seg is smaller by 2.83 MB and 1.4 ms faster. Given

**Table 1**
Instance segmentation results of different model architectures. Bold values depict best performance.

| Model | Scale | *mAP*@50–95 (%) | | *Inference* (ms) |
|---|---|---|---|---|
| | | Box | Mask | |
| YOLOv9-GELAN | c (56.28 MB) | 95.3 | 67.9 | 35.7 |
| | e (121.95 MB) | **96.1** | 69.9 | 53.1 |
| YOLO11 | n (6.05 MB) | 84.7 | 51.3 | **25.3** |
| | s (20.56 MB) | 91.5 | 61.1 | 26.8 |
| | m (45.20 MB) | 94.6 | 67.3 | 30.4 |
| | l (55.88 MB) | 94.8 | 67.6 | 39.9 |
| | x (124.82 MB) | 95.4 | 68.5 | 40.3 |
| YOLO26 | n (6.59 MB) | 83.6 | 56.0 | 28.6 |
| | s (23.39 MB) | 91.7 | 65.8 | 28.2 |
| | m (54.52 MB) | 94.5 | 70.1 | 32.1 |
| | l (63.49 MB) | 95.1 | 70.7 | 37.3 |
| | x (141.77 MB) | 95.7 | **72.3** | 37.7 |
| RT-DETR | n (128.09 MB) | 80.5 | 51.4 | 52.9 |
| | s (128.60 MB) | 82.4 | 51.4 | 64.5 |
| | m (136.16 MB) | 83.5 | 52.5 | 57.3 |
| | l (136.85 MB) | 85.0 | 54.2 | 69.9 |
| | x (145.39 MB) | 88.7 | 56.4 | 70.7 |

the full pipeline incorporates two additional processing modules MaskMend and PPL the marginal gain in segmentation accuracy was considered insufficient to justify the additional model size and computational cost. Thus, YOLO11s-seg was selected as the instance segmentation model for further experiments.

#### 3.1.1. Evaluation of the instance segmentation model for summer kiwifruit canopy

YOLO11s-seg achieved a decent performance in segmenting branches and shoots under full foliage of summer. For branches and shoots, the *AP* exceeded 90% for detection bounding boxes and were 63.9% and 58.2% for instance masks, respectively as shown in Fig. 6. Across all classes, the *mAP* was above 90% for bounding boxes and close to 65% for masks.

The consistently high *AP* and *mAP* values for bounding boxes demonstrate the model's capacity to detect and localize branches and shoots with high reliability, even in the full foliage of summer canopy. This indicates robust model generalization across multi-season kiwifruit canopy image data, which contain branches and shoots of different lengths and orientations. On the other hand, the lower mask *AP* and *mAP* values particularly for shoots, highlight limitations of the model's ability

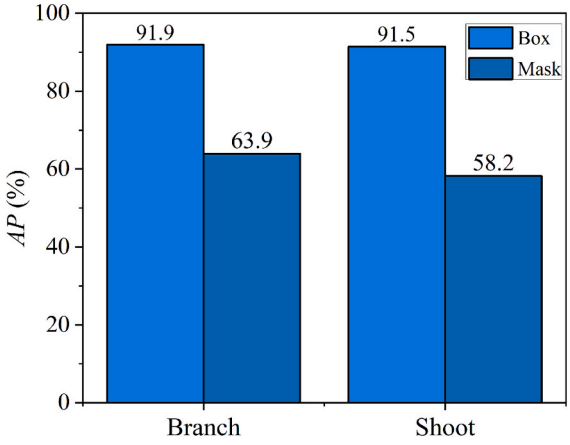



**Fig. 6.** Instance segmentation *AP* values of YOLO11s-seg on branches and shoot classes.

to precisely segment the branches and shoots. This reduction in segmentation performance is likely attributed to the difficulty of outlining non-terminating shoots that curl, twist and overlap within the dense foliage as shown by the red and light blue spheres in Fig. 7, respectively. Moreover, vigorous shoot growth increases occlusion as shown by yellow spheres in Fig. 7, which limits model learning by preventing complete feature extraction of the branches and shoots. Nevertheless, the instance segmentation model yields acceptable results for summer pinching, and this performance may have been aided by the inclusion of both close-view and far-view images (Wang et al., 2024a).

The instance segmentation model metrics obtained in this study are competitive with previous works focused on winter pruning. For instance, Tong et al. (2022) obtained a *mAP* of 70.9% for masks and 78.1% for bounding boxes for apple tree branches, trunks, and support, instance segmentation using Cascade Mask R-CNN. The segmentation mask *mAP* was 9.8% higher compared to the present study, while the detection bounding box *mAP* in this study was 13.4% higher. In a subsequent work, Tong et al. (2023) employed SOLOv2 for apple branch segmentation and reported a mask *mAP* of 68.9%, a value slightly higher than that achieved in this study. These results suggest that, the present study achieved competitive, and in some cases superior, performance under the more challenging dense summer canopy of kiwifruit.

The generalization ability of the model enables it to accurately identify and outline branches and shoots, key structures for summer pinching as highlighted in Fig. 7. The model can effectively capture the morphological overlap between branches in pink bounding boxes and shoots in blue bounding boxes, which is essential for KPPPA in positioning pinching points.

### 3.2. Performance evaluation of kiwifruit pinching point positioning algorithm

The KPPPA demonstrates strong performance in positioning pinching points, exhibiting both high accuracy and efficiency. To evaluate its performance, the algorithm was tested on an evaluation dataset consisting of 468 images with a total of 1058 pinching points. The results are presented in Table 2. The algorithm achieved a high success rate, failing to identify only 137 pinching points. Among the successful instances, $E_{avg}$ between the predicted and ground-truth pinching points was 5.8 pixels, demonstrating strong positional accuracy. Additionally, the $MAE_x$ and $MAE_y$ differed by only 0.9 pixels, with a slightly higher precision along the y-axis. The algorithm also processed images at an average speed of 7 frames per second, showing the potential for real-time deployment in agricultural applications.

To further analyze the positional accuracy of the algorithm, distribution of positioning errors in the evaluation dataset was examined. A histogram of the Average Euclidean error ($E_{avg}$) per image presented in Fig. 8(a) shows the majority of predictions exhibited low errors, predominantly within pixel range of 1 ~ 9. Only a small number of instances (50 points) exceeded an average error of 9.0 pixels, highlighting the algorithm's high spatial accuracy. The scatter plot in Fig. 8(b), visually compares predicted (red stars) and ground-truth (blue dots) pinching points. The visualization confirms that the predictions are tightly clustered around true points with minimal systematic bias. While small prediction errors were expected, the results demonstrate that KPPPA consistently approximates true pinching positions with high reliability.

An analysis of the directional error metrics revealed a minor symmetrical bias, which is not unusual in agricultural vision tasks where plant geometries have inherent directional characteristics. Both the x-axis ($MAE_x$) and y-axis ($MAE_y$) error distributions showed peak pixel error counts in the range of 0 ~ 5. However, as shown in Fig. 9(a), $MAE_x$ showed a slightly broader spread, with a few instances exceeding 10.0 pixels. In contrast, $MAE_y$ exhibited a narrower, sharper peak near 3.5 pixels and fewer extreme errors, as detailed in Fig. 9(b). The small pixel difference between the mean directional errors suggests only a minor



**Fig. 7.** Segmentation results of YOLO11s-seg with branches in pink and shoots in blue. The spheres indicate where the model struggled to segment curling (red), overlapped (yellow) and occluded shoots (light blue).

**Table 2**
Evaluation performance of KPPPA in positioning pinching points for kiwifruit.

| Metric | Success rate (%) | $E_{avg}$ (pixels) | $MAE_x$ (pixels) | $MAE_y$ (pixels) | Speed (ms) |
|---|---|---|---|---|---|
| Values | 87.0 | 5.8 | 3.6 | 2.7 | 136.6 |

directional bias. Similar trends have been observed in related work, such as Wang et al. (2024c), who reported concentrated error distributions within 0 ~ 3 pixels for wolfberry picking point positioning.

Deployability on embedded hardware was assessed by benchmarking KPPPA on an NVIDIA Jetson AGX Orin Developer Kit (MAXN power mode, JetPack R36.4.3) using 147 images randomly selected from the evaluation dataset. After 20 warmup iterations, end-to-end latency was 170.8 ± 42.7 ms per frame (median 165.1 ms, p95 248.5 ms), corresponding to a sustained throughput of 5.8 FPS. Instance segmentation using YOLO11s-seg was highly stable at 46.9 ± 1.3 ms, while MaskMend sub-algorithm accounted for 52% of total pipeline time (89.0 ± 31.5 ms), with variability scaling with canopy complexity and the number of fragmented shoot masks per image. Peak GPU memory consumption was 203.4 MB, well within the Jetson AGX Orin's 64 GB unified memory capacity. System on Chip (SoC) temperature remained stable between 51.9°C and 53.0°C, confirming the absence of thermal throttling during sustained operation. These results demonstrate feasibility for embedded orchard deployment, with MaskMend optimization identified as the primary target for future latency reduction.

### 3.3. Analysis of the positioning algorithm performance

#### 3.3.1. Positioning effect of the algorithm under different canopy conditions

KPPPA demonstrated strong ability for positioning pinching points in challenging outdoor environments. Canopy density represents the most challenging condition given that dense foliage increases the frequency and severity of shoot occlusions (Cardenas-gallegos et al., 2025). The effect of canopy density on KPPPA was assessed and the results are presented in Table 3. Under complex canopy conditions, the algorithm recorded the lowest $E_{avg}$ of 4.8 pixels across all tested conditions, while exhibiting the highest processing time of 167.3 ms and the second lowest success rate. Interestingly, simple canopy condition produced the highest $E_{avg}$ of 6.8 pixels, and a marginally better success rate. This result reflects two opposing mechanisms, in dense canopies, neighboring vines

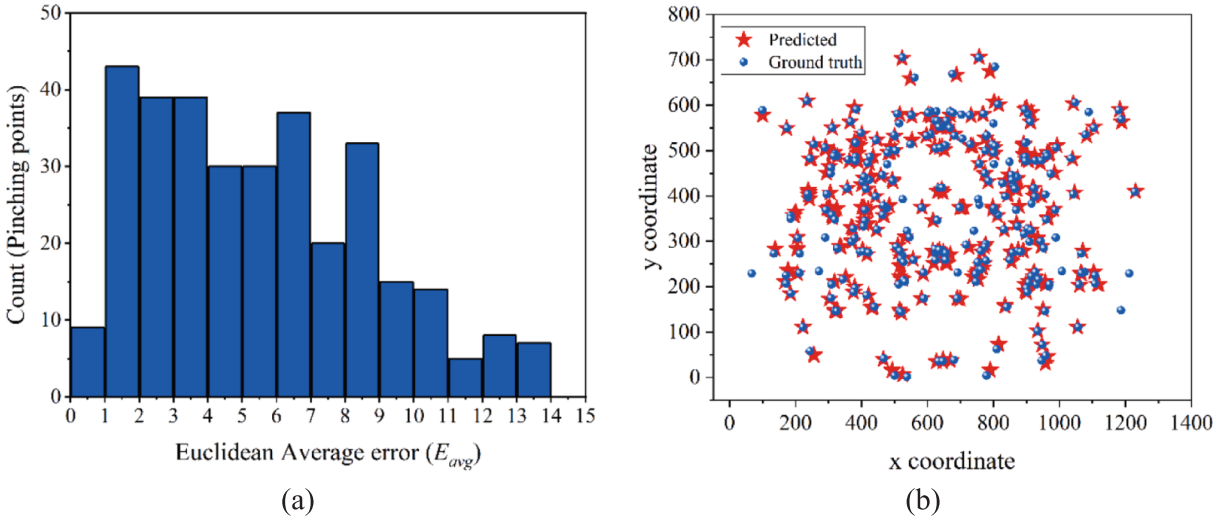



**Fig. 8.** KPPPA positioning results highlighting (a), distribution of Average Euclidean error ($E_{avg}$) and (b), the pinching points with predicted (red stars) and ground truth (blue dots).

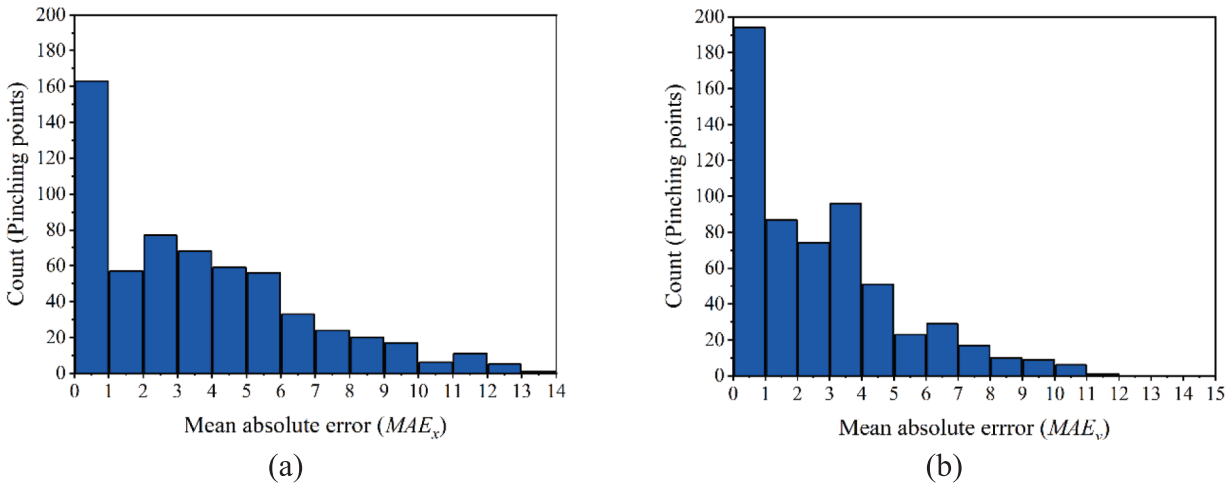



**Fig. 9.** KPPPA directional error distribution along the (a) x-axis and (b) y-axis.

**Table 3**
Evaluation performance of KPPPA under different canopy density conditions.

| Condition | Success rate (%) | $E_{avg}$ (pixels) | $MAE_x$ (pixels) | $MAE_y$ (pixels) | Speed (ms) |
|---|---|---|---|---|---|
| Complex | 86.3 | 4.3 | 3.2 | 2.7 | 167.3 |
| Simple | 88.7 | 6.8 | 4.1 | 3.4 | 110.4 |

physically constrain shoot growth, producing more linear shoot trajectories that are easier to skeletonize accurately. The frequent occlusions are largely resolved by MaskMend, as confirmed by the ablation in Table 6. In simple canopies, by contrast, non-terminating shoots grow unconstrained and exhibit pronounced curling and twisting. This morphological irregularity introduces curvature that the Dijkstra algorithm cannot fully capture, accumulating small positional errors along the shoot path. Success rate, however, remains higher in simple canopies, indicating that segmentation is easier, only the per-point spatial precision degrades. The ablation results corroborate this, since removing MaskMend caused a drop of 3.7% in success rate under high-density conditions compared to 2.5% in low density hence; mask reconstruction is critical in occluded, dense canopy environments.

In these conditions KPPPA effectively addressed the occlusions through leveraging the shoot's morphological traits. Specifically, shoot basal location and the local search direction of reconnection candidate extreme endpoints were utilized by MaskMend sub-algorithm to reconstruct fragmented shoot masks. As illustrated in Fig. 10(a), MaskMend successfully reconnected small gaps caused by buds, while in Fig. 10(b), it handled larger gaps caused by leaf coverage. The same Fig. 10(b) reveals how basal location was effective in dealing with unpredictable shoot orientation. The shoot fragment **f-1** is growing along the branch, while **f-2** is growing away from it. The two fragments meet the criteria for reconnection through their local orientations; however, since the basal coordinate of **f-2** is excluded from reconnections the two were not reconnected.

### 3.3.2. *Effect of weather conditions on the algorithm*

KPPPA demonstrated reliable positioning capability across varying ambient lighting conditions. Cloudy conditions yielded a higher success rate of 88.5%, compared to 85.8% under sunny conditions shown in Table 4. This 2.7% improvement can be attributed to more diffuse and uniform illumination in cloudy conditions, reducing high-contrast shadows and specular reflections that fragment shoot masks leading to

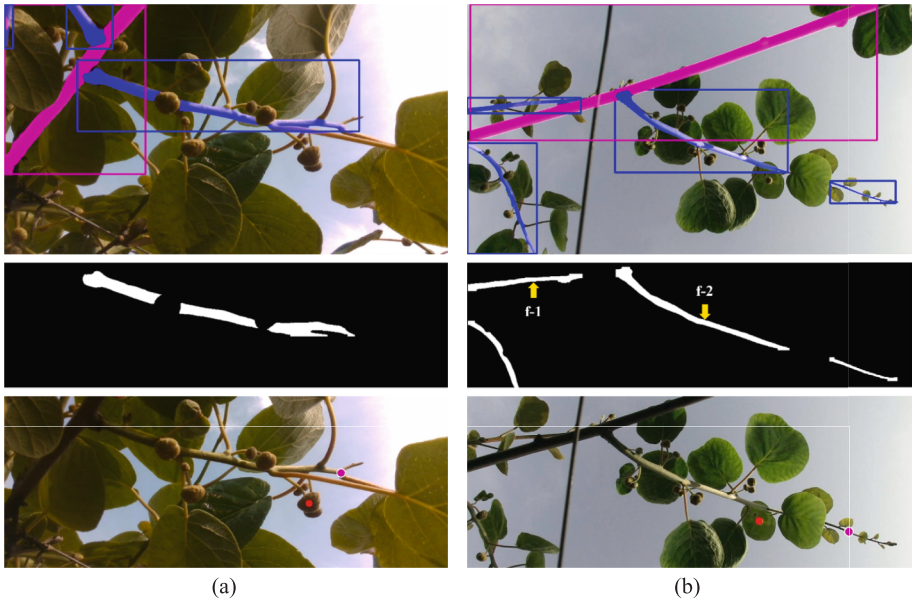



**Fig. 10.** Examples of KPPPA under occlusion caused by (a) buds and (b) leaves. The segmented branches are shown in pink, shoots in blue and the distal bud in red.

**Table 4**
Performance of KPPPA under different weather conditions in positioning pinching points.

| Condition | Success rate (%) | $E_{avg}$ (pixels) | $MAE_x$ (pixels) | $MAE_y$ (pixels) | Speed (ms) |
|---|---|---|---|---|---|
| Sunny | 85.8 | 5.6 | 3.1 | 2.7 | 137.7 |
| Cloudy | 88.5 | 5.8 | 4.2 | 3.9 | 132.6 |

underestimation of the shoot length as discussed in section 3.3.5. This contrasts with Li et al. (2024), who reported higher success in sunny conditions likely due to camera orientation differences, as facing the kiwifruit canopy places the sun within the camera's field of view, whereas their sideways camera orientation in mango picking point positioning is largely unaffected. The low $E_{avg}$ of 5.6 pixels under sunny conditions suggest adequate pinching point positioning accuracy when a shoot is successfully detected. However, the ablation study in Table 6 revealed that removing MaskMend sub-algorithm, caused success rates to drop by over 2%, confirming that the primary challenge under direct sunlight is the initial shoot segmentation not positioning accuracy. Additionally, the KPPPA required more processing time under sunny conditions (137.7 ms) compared to cloudy (132.6 ms), reflecting the additional computational effort needed to resolve ambiguous edge features. The results confirm that the proposed algorithm generalizes well across practical outdoor weather conditions without requiring condition-specific recalibration.

### 3.3.3. Camera-to-canopy distance effect on the performance of KPPPA

Camera-to-canopy distance introduces a distinct source of variability by altering the scale and resolution of shoot segments within the image. In far-view images KPPPA achieved a lower $E_{avg}$ of 1.4 pixels than in near-view as shown in Table 5, indicating that images captured at greater distance benefit from a wider field of view that captures more shoot geometry, which is crucial for positioning pinching points by KPPPA. However, this positioning accuracy is at the cost of success rate, which fell by 4.2% relative to near-view condition. At greater camera distances, smaller shoot sizes and low-resolution increase mask fragmentation and incomplete shoot segmentation, leading to inaccurate positioning of pinching points. The near-view condition, conversely, achieved the highest success rate of 89.3% across all subsets, benefiting from higher-resolution shoot representations that facilitate reliable detection, despite greater morphological complexity that increases positional error (Jin et al., 2022). Furthermore, the speed of the KPPPA increased in far-view images reaching 158.7 ms per image. This is likely due to shoots appearing smaller and more fragmented in far-view images, resulting in a larger number of mask instances per image for the algorithm to evaluate. The ablation study in Table 6 provides additional insight into the role of MaskMend across distance conditions, as its removal reduced the success rate by 2.5% for near-view and 3.4% for far-view, with a notably larger absolute impact in far-view images. Thus, MaskMend is valuable in far-view conditions, where partial occlusions produce disconnected mask fragments.

### 3.3.4. Effect of the proposed sub-algorithms

The success of KPPPA under different conditions can be attributed to the inclusion of the proposed sub-algorithms. MaskMend was effective in high density situations and far-view images as discussed above and as shown by the ablation in Table 6. Mask reconstruction was successful due to carefully selected optimal parameters used. The base marker distance of 40 pixels was fundamental in estimating the local direction to search for candidate extreme endpoints for reconnections. It was observed that larger base distances improved search direction estimation by offsetting the effect of extreme points lying off the central axis of the mask, but would fail on smaller mask fragments. Additionally, an angular threshold of 20° combined with a search distance of 150 pixels was essential in eliminating non-adjacent reconnection candidate extreme endpoints. These optimized parameters enabled MaskMend sub-algorithm to reconstruct fragmented shoot masks with high accuracy. The inclusion of basal point location and local search direction of candidate extreme endpoints was essential to occlusion handling, contributing significantly to KPPPA's effectiveness in complex outdoor orchard conditions.

**Table 5**
Performance of KPPPA on near-view and far-view conditions.

| Condition | Success rate (%) | $E_{avg}$ (pixels) | $MAE_x$ (pixels) | $MAE_y$ (pixels) | Speed (ms) |
|---|---|---|---|---|---|
| Near-view | 89.7 | 6.5 | 3.7 | 3.1 | 112.8 |
| Far-view | 85.5 | 5.1 | 3.4 | 2.8 | 158.7 |

**Table 6**
Performance of KPPPA under different conditions without MaskMend sub-algorithm.

| Condition | Success rate (%) | $E_{avg}$ (pixels) | $MAE_x$ (pixels) | $MAE_y$ (pixels) | Speed (ms) |
|---|---|---|---|---|---|
| Sunny | 83.7 | 4.1 | 2.3 | 1.7 | 119.4 |
| Cloudy | 84.2 | 4.5 | 2.5 | 2.1 | 118.2 |
| Complex | 82.6 | 3.9 | 2.5 | 2.0 | 141.5 |
| Simple | 86.2 | 4.7 | 2.3 | 1.8 | 96.2 |
| Near | 86.8 | 4.5 | 2.0 | 1.7 | 107.5 |
| Far | 82.1 | 4.0 | 2.6 | 2.1 | 141.1 |

KPPPA utilized shoot length to filter out terminating from non-terminating shoots. For example, among the ten segmented shoots highlighted in blue bounding boxes in Fig. 11(a), seven were filtered out as terminating while three were identified as non-terminating based on their length. The algorithm then identified the pinching points on the non-terminating shoots (magenta and white) shown in Fig. 11(b), positioning them just beyond the distal bud (red dot). To determine the optimal pinching point proportion, leaf counts beyond the distal bud were measured across 57 non-terminating shoots spanning two cultivars 'Xuxiang' and 'Nongda Yuxiang' at two growth stages: inflorescence emergence (Stage 5) and flowering (Stage 6) (Salinero et al., 2009). The results are shown in Fig. 12. At 82%, compliance with the horticultural guideline of 3 ~ 6 leaves were 79.2% for Xuxiang at Stage 5 (mean = 5.58 ± 1.18 leaves), 69.2% at Stage 6 (mean = 6.31 ± 0.75), and 65.0% for Nongda Yuxiang at Stage 5 (mean = 4.60 ± 1.79). While the 70% proportion achieved marginally higher overall compliance (89.5% combined), its mean leaf count (3.50 ~ 4.9) is near the lower boundary of the guideline, leaving minimal buffer against positioning errors during robotic pinching. The 82% proportion (magenta) places the pinching point closer to the upper end of the guideline, reducing the risk of insufficient leaf area for fruit development, consistent with the examples illustrated in Fig. 13, where lower proportions (50% ~ 70%) left insufficient foliage while higher proportions (90% ~ 95%) risks regrowth stimulation by placing the point too close to the shoot apex. Furthermore, the cultivar and stage differences particularly the shift from 5.58 to 6.31 mean leaves between Stage 5 and Stage 6 in Xuxiang confirm that the optimal proportion is growth-stage-dependent and should be recalibrated for other cultivars or later phenological stages. These results confirm that the proposed method can accurately identify suitable pinching locations during summer pruning representing a significant advance toward automated kiwifruit canopy management.

### 3.3.5. Error analysis of the kiwifruit pinching point positioning algorithm

The performance of the KPPPA is fundamentally constrained by the accuracy of its underlying instance segmentation model. Segmentation errors, particularly incomplete segmentations, were a primary source of positioning inaccuracies. This aligns with findings from Li et al. (2024), who identified similar challenges in dense agricultural environments. For example, lighting variation in the orchard such as shadows cast by




**Fig. 11.** KPPPA performance showing (a) the segmentation results with branches (pink) and shoots (blue); (b) the pinching points (magenta and white) positioned on the filtered non-terminating shoots beyond distal bud (red).

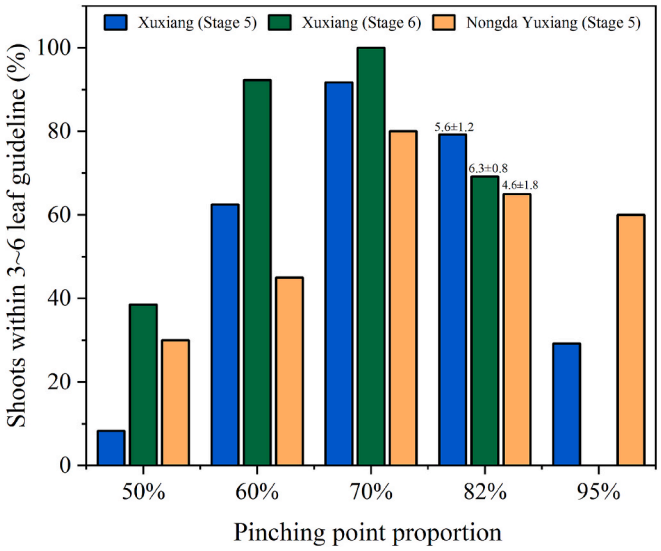



**Fig. 12.** Compliance with the 3 ~ 6 horticultural leaf guideline at each pinching point proportion, across two cultivars and growth stages. Mean leaf count ± SD shown above each bar group for the 82% proportion.

surrounding leaves occasionally led to incomplete segmentation of the shoot apex region such as shown in Fig. 14(a). This resulted in underestimation of the shoot total length and a subsequent misplacement of the pinching point (magenta and white), placed offset from the ground truth (light blue) shown in Fig. 14(b). This particular failure highlights the sensitivity of the segmentation model to uneven lighting, a persistent challenge in real-world agricultural settings (Cardenas-gallegos et al., 2025).

Another significant source of error was severe occlusion, where significant sections on non-terminating shoots were hidden within the dense canopy as shown in Fig. 15(a) by yellow circles. In such cases, the KPPPA utilized only the visible fragments to position pinching points (magenta and white) as shown in Fig. 15(b). However, these locations were unreliable because they compromise the optimal fruit-to-leaf ratio. The length discrimination threshold (T) in the PPL sub-algorithm serves as a partial safeguard for such cases; when occlusion reduces the visible reconstructed skeleton below T, the shoot is classified as terminating and excluded from pinching, rather than incorrectly processed. However, cases where the visible fragment clears T yet the true pinching region remains hidden represent a significant limitation for a real-world robotic pinching implementation, and cannot be overcome by current solutions (Chen et al., 2023, 2024a; Sun et al., 2023, 2024). A potentially viable solution would be modifying the method proposed by Sun et al. (2024), which involves estimating an occlusion rate then use it to produce a confidence score to judge whether to pinch, skip or flag the shoot for human intervention. Additionally, the confidence score can also be leveraged to adjust camera position which can mitigate incomplete segmentation caused by uneven lighting.

Recent works has largely focused on winter pruning in less visually complex orchards environments for fruit trees like grapevine (Gentilhomme et al., 2023; Oliveira et al., 2024), jujube (Fu et al.,

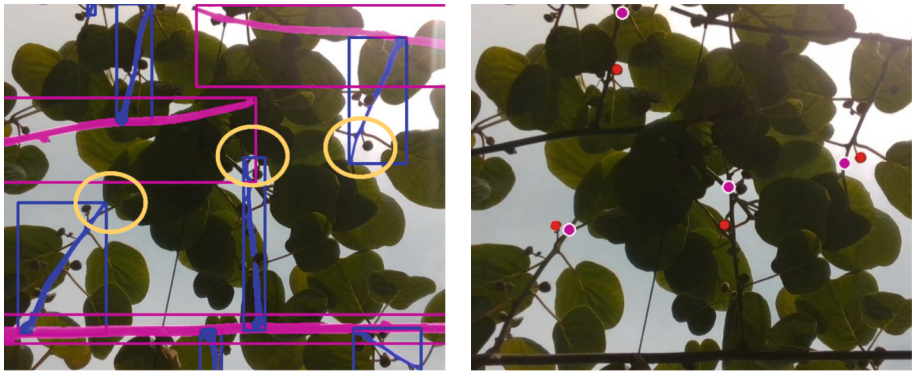



**Fig. 13.** Evaluation of pinching point candidates at varying proportions of total shoot length. Colored points indicate proportions of 95% (dark blue), 90% (light blue), 82% (magenta), 70% (yellow), 60% (black), and 50% (green). The red point marks the distal.



(a) (b)

**Fig. 14.** Example of incomplete segmentation highlighting (a) failure of mask to cover the entire shoot (blue), resulting in (b) inaccurate pinching points (magenta and white) positions. Segmented branches are depicted in pink, while the distal bud is shown in red.



(a) (b)

**Fig. 15.** Severe occlusion examples were (a) significant portion of the shoot (blue) is hidden by canopy (yellow circles), leading to (b) pinching points (magenta and white) placed at incorrect locations. Segmented branches are presented in pink, while the distal bud is depicted in red.

2023), cherry tree (You et al., 2023), and apple trees (Tong et al., 2022, 2023), summer pruning presents unique challenges due to the dense foliage. Hence, this has remained underexplored in agricultural robotics. This work has demonstrated that, it is possible to develop targeted algorithms capable of navigating these challenges by leveraging the morphological characteristics of kiwifruit shoots. The integrated YOLO11s-seg and KPPPA method successfully identified pinching points by utilizing the shoot morphology traits and apply horticultural pinching rules on the non-terminating shoots.

Future work will aim to improve the performance of the underlying segmentation model, specifically the inclusion modified YOLO or transformer-based models. Integrating these architectural advances in future iterations of the KPPPA could reduce inference below the current 136.6 ms per frame and improve segmentation of heavily occluded shoots. Complimenting strong segmentation model will be physical methods to improve visibility such as selective leaf removal (Poni et al., 2023), to enable safe and automated summer pinching in kiwifruit cultivation. Additionally, depth data from either ZED 2i or RealSense D435 cameras will be integrated into the pipeline to transition from 2D to 3D pinching point localization, improving spatial precision for robotic deployment. Specifically, the pinching point $(u, v)$ from the KPPPA will be back-projected to 3D using depth $Z$ sampled from the registered depth map at $(u, v)$; a $7 \times 7$ neighborhood will be employed to suppress sensor noise. Then the 3D point will be obtained in the camera frame via the pinhole model $X = (u - c_x)\, Z / f_x$, $Y = (u - c_y)\, Z / f_y$, where $f_x$, $f_y$ are the focal lengths and $c_x$, $c_y$ are the principal components of camera intrinsics. The pinching point will be subsequently transformed from the camera frame into the arm-base frame using pre-calibrated extrinsic parameters. This will be complemented by end-to-end actuation experiments with robotic pruning arms under real outdoor orchard conditions to validate system-level performance.

## 4. Conclusions

This study introduced a novel automated method for positioning pinching points on kiwifruit shoots, addressing a key challenge in summer pruning automation. The method combined YOLO11s-seg with a shoot morphology-guided algorithm. YOLO11s-seg accurately segmented branches and shoots with a *mAP* of 91.7% for bounding boxes and 61.1% for masks. Specifically, the model detected and segmented branches with *AP* values of 91.9% and 63.9% while for shoots the *AP* values were 91.5% and 58.2%%, respectively. The segmented output was then processed by KPPPA to position pinching points. KPPPA achieved a positioning success rate of 87.0%, with an Average Euclidean error ($E_{avg}$) of 5.8 pixels, demonstrating the robustness of the method despite the complex summer canopy conditions. Incorporating shoot basal location and local search direction of reconnection candidates was essential in reconstructing occlusion-induced fragmented shoot masks. The length of the shoot can be utilized to distinguish between terminating and non-terminating shoots. Additionally, a proportional constant of 82% closely aligned with the optimal fruit-to-leaf ratio, providing a simple yet effective geometry-based approach for positioning pinching points. These findings illustrate that the proposed method is effective in positioning pinching points in complex orchards, which is crucial to realize robotic pruning of kiwifruit in summer. Future work will focus on incorporating image depth information to enable outdoor orchard evaluation, as well as integrating active sensing and selective leaf thinning to address occasional segmentation failures due to uneven lighting, severe occlusion, and curling of shoots.

## CRediT authorship contribution statement

**Bryan Gilbert Murengami:** Writing – original draft, Software, Methodology, Data curation. **Leilei He:** Visualization, Methodology. **Shuolin Kong:** Investigation, Data curation. **Lamin L. Janneh:** Writing

– review & editing, Conceptualization. **Xiaojuan Liu:** Validation, Methodology. **Wenhao Guo:** Investigation, Data curation. **Rui Li:** Writing – review & editing, Supervision, Investigation. **Jinyong Chen:** Writing – review & editing, Methodology. **Spyros Fountas:** Writing – review & editing, Conceptualization. **Hin Lim Shen:** Writing – review & editing, Formal analysis. **Longsheng Fu:** Writing – review & editing, Supervision, Methodology, Data curation, Conceptualization.

## Declaration of generative AI and AI-assisted technologies in the writing process

During the preparation of this work, the authors used Claude for grammar checking to improve the readability of article. After using these servers, the authors reviewed and edited the content as needed and takes full responsibility for the content of the published article.

## 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.

## Acknowledgements

This work was supported by the National Natural Science Foundation of China (32371999, 32501786); Chinese Universities Scientific Fund (2452026021); National Foreign Expert Project, Ministry of Human Resources and Social Security, China (H20240238, Y20240046); Key Research and Development Program of Shaanxi, China (2024PT-ZCK-27); Key Research and Development Program of Shaanxi, China (2024NC-YBXM-195); Central Public-interest Scientific Institution Basal Research Fund, China (1610192023105); National Key Research and Development Program (2022YFD1600700).

## Data availability

Data is available at https://github.com/Smart-Ag/Kiwifruit_pinching_images.

## References

Akbaş, H., Özcan, M., 2020. Effects of fruit/leaf ratios on fruit characteristics in kiwifruit. Erwerbs-obstbau 62, 369–375. https://doi.org/10.1007/s10341-020-00511-z.

Cardenas-gallegos, J.S., Severns, P.M., Klime, P., Lacerda, L.N., Peduzzi, A., Ferrarezi, R. S., 2025. Reliable plant segmentation under variable greenhouse illumination conditions. Comput. Electron. Agric. 229, 109711. https://doi.org/10.1016/j.compag.2024.109711.

Chen, Z., Granland, K., Newbury, R., Chen, C., 2023. HOB-CNN: Hallucination of occluded branches with a convolutional neural network for 2D fruit trees. Smart Agric. Technol. 3, 100096. https://doi.org/10.1016/j.atech.2022.100096.

Chen, Z., Granland, K., Tang, Y., Chen, C., 2024a. HOB-CNNv2: deep learning based detection of extremely occluded tree branches and reference to the dominant tree image. Comput. Electron. Agric. 218, 108727. https://doi.org/10.1016/j.compag.2024.108727.

Chen, Z., Wang, Y., Tong, S., Chen, C., Kang, F., 2024b. Grapevine branch recognition and pruning point localization technology based on image processing. Appl. Sci. 14, 3327. https://doi.org/10.3390/app14083327.

Fu, Y., Xia, Y., Zhang, H., Fu, M., Wang, Y., Fu, W., Shen, C., 2023. Skeleton extraction and pruning point identification of jujube tree for dormant pruning using space colonization algorithm. Front. Plant Sci. 13, 1–11. https://doi.org/10.3389/fpls.2022.1103794.

Gené-Mola, J., Ferrer-Ferrer, M., Gregorio, E., Blok, P.M., Hemming, J., Morros, J.R., Rosell-Polo, J.R., Vilaplana, V., Ruiz-Hidalgo, J., 2023. Looking behind occlusions: a study on amodal segmentation for robust on-tree apple fruit size estimation. Comput. Electron. Agric. 209, 107854. https://doi.org/10.1016/j.compag.2023.107854.

Gentilhomme, T., Villamizar, M., Corre, J., Odobez, J.M., 2023. Towards smart pruning: ViNet, a deep-learning approach for grapevine structure estimation. Comput. Electron. Agric. 207, 107736. https://doi.org/10.1016/j.compag.2023.107736.

He, L., Ren, X., Gao, Q., Zhao, X., Yao, B., Chao, Y., 2017. The connected-component labeling problem: a review of state-of-the-art algorithms. Pattern Recognit. J. 70, 25–43. https://doi.org/10.1016/j.patcog.2017.04.018.

Jin, Y., Liu, J., Wang, J., Xu, Z., Yuan, Y., 2022. Far-near combined positioning of picking-point based on depth data features for horizontal-trellis cultivated grape. Comput. Electron. Agric. 194, 106791. https://doi.org/10.1016/j.compag.2022.106791.

Keysers, D., Deselaers, T., Ney, H., 2004. Pixel-to-pixel matching for image recognition using Hungarian graph matching. Lect. Notes Comput. Sci 3175, 154–162. https://doi.org/10.1007/978-3-540-28649-3_19.

Kim, S., Hong, S., Ryu, J., Kim, E., Lee, C., Kim, G., 2023. Application of amodal segmentation on cucumber segmentation and occlusion recovery. Comput. Electron. Agric. 210, 107847. https://doi.org/10.1016/j.compag.2023.107847.

Kok, E., Chen, C., 2024. Occluded apples orientation estimator based on deep learning model for robotic harvesting. Comput. Electron. Agric. 219, 108781. https://doi.org/10.1016/j.compag.2024.108781.

Kok, E., Wang, X., Chen, C., 2023. Obscured tree branches segmentation and 3D reconstruction using deep learning and geometrical constraints. Comput. Electron. Agric. 210, 107884. https://doi.org/10.1016/j.compag.2023.107884.

Li, G., Suo, R., Zhao, G., Gao, C., Fu, L., Shi, F., Dhupia, J., Li, R., Cui, Y., 2022. Real-time detection of kiwifruit flower and bud simultaneously in orchard using YOLOv4 for robotic pollination. Comput. Electron. Agric. 193, 106641. https://doi.org/10.1016/j.compag.2021.106641.

Li, H., Huang, J., Gu, Z., He, D., Huang, J., Wang, C., 2024. Positioning of mango picking point using an improved YOLOv8 architecture with object detection and instance segmentation. Biosyst. Eng. 247, 202–220. https://doi.org/10.1016/j.biosystemseng.2024.09.015.

Li, L., Li, K., He, Z., Li, H., Cui, Y., 2025a. Kiwifruit segmentation and identification of picking point on its stem in orchards. Comput. Electron. Agric. 229, 109748. https://doi.org/10.1016/j.compag.2024.109748.

Li, P., Chen, J., Chen, Q., Huang, L., Jiang, Z., Hua, W., Li, Y., 2025b. Detection and picking point localization of grape bunches and stems based on oriented bounding box. Comput. Electron. Agric. 233, 110168. https://doi.org/10.1016/j.compag.2025.110168.

Liang, J., Huang, K., Lei, H., Zhong, Z., Cai, Y., Jiao, Z., 2024. Occlusion-aware fruit segmentation in complex natural environments under shape prior. Comput. Electron. Agric. 217, 108620. https://doi.org/10.1016/j.compag.2024.108620.

Liao, G.L., Xu, X.B., Liu, Q., Zhong, M., Huang, C.H., Jia, D.F., Qu, X.Y., 2020. A special summer pruning method significantly increases fruit weight, ascorbic acid, and dry matter of Kiwifruit ("Jinyan", Actinidia eriantha 3 A.chinensis). HortSci. 55, 1698–1702. https://doi.org/10.21273/HORTSCI15158-20.

Mhetre, S.B.G.V.B., Verma, W.H.R.M.K., Kavitha, R.E.R., Prusty, R.R., 2025. Agronomic practices affecting the yield and quality of Kiwifruit: a comprehensive review. Appl. Fruit Sci. 67, 239. https://doi.org/10.1007/s10341-025-01471-y.

Oliveira, F., da Silva, D.Q., Filipe, V., Pinho, T.M., Cunha, M., Cunha, J.B., dos Santos, F. N., 2024. Enhancing grapevine node detection to support pruning automation: leveraging state-of-the-art YOLO detection models for 2D image analysis. Sensors 24, 6774. https://doi.org/10.3390/s24216774.

Patiyal, V., Rana, V.S., Rana, N., Hashem, A., Abd_Allah, E.F., Sharma, S., 2024. Appraisal of extended cane length and fruit thinning strategies on the performance of growth yield and quality of kiwifruit. Heliyon 10, e29546. https://doi.org/10.1016/j.heliyon.2024.e29546.

Poni, S., Frioni, T., Gatti, M., 2023. Summer pruning in mediterranean vineyards: is climate change affecting its perception, modalities, and effects? Front. Plant Sci. 14, 1–20. https://doi.org/10.3389/fpls.2023.1227628.

Richardson, A., Eyre, V., Kashuba, P., Ellingham, D., Jenkins, H., Nardozza, S., 2021. Early shoot development affects carbohydrate supply and fruit quality of red-fleshed actinidia chinensis var. Chinensis 'zes008. Agronomy 11, 66. https://doi.org/10.3390/agronomy11010066.

Salinero, M.C., Vela, P., Sainz, M.J., 2009. Phenological growth stages of kiwifruit (Actinidia deliciosa 'Hayward'). Sci. Hortic. (Amsterdam) 121, 27–31. https://doi.org/10.1016/j.scienta.2009.01.013.

Sapkota, R., Karkee, M., 2025. Ultralytics YOLO evolution: An overview of YOLO26, YOLO11, YOLOv8 and YOLOv5 object detectors for computer vision and pattern recognition. arXiv 26, 1–16. https://doi.org/10.48550/arXiv.2510.09653.

Sapkota, R., Rizwan, M.F., Chetan, Q., 2025. YOLO advances to its genesis: a decadal and comprehensive review of the you only look once (YOLO) series. Artif. Intell. Rev. 54, 274. https://doi.org/10.1007/s10462-025-11253-3.

Shi, X., Wang, S., Zhang, B., Ding, X., Qi, P., Qu, H., Li, N., Wu, J., Yang, H., 2025. Advances in object detection and localization techniques for fruit harvesting robots. Agronomy 15, 145. https://doi.org/10.3390/agronomy15010145.

Song, Z., Zhou, Z., Wang, W., Gao, F., Fu, L., Li, R., Cui, Y., 2021. Canopy segmentation and wire reconstruction for kiwifruit robotic harvesting. Comput. Electron. Agric. 181, 105933. https://doi.org/10.1016/j.compag.2020.105933.

Sun, T., Zhang, W., Miao, Z., Zhang, Z., Li, N., 2023. Object localization methodology in occluded agricultural environments through deep learning and active sensing. Comput. Electron. Agric. 212, 108141. https://doi.org/10.1016/j.compag.2023.108141.

Sun, T., Zhang, W., Gao, X., Zhang, W., Li, N., Miao, Z., 2024. Efficient occlusion avoidance based on active deep sensing for harvesting robots. Comput. Electron. Agric. 225, 109360. https://doi.org/10.1016/j.compag.2024.109360.

Tian, Z., Guo, X., Ma, W., Xue, X., 2025. Research on kiwifruit harvesting robot worldwide: a solution for sustainable development of kiwifruit industry. Smart Agric. Technol. 10, 100792. https://doi.org/10.1016/j.atech.2025.100792.

Tong, S., Yue, Y., Li, W., Wang, Y., Kang, F., Feng, C., 2022. Branch identification and junction points location for apple trees based on deep learning. Remote Sens. 14, 4495. https://doi.org/10.3390/rs14184495.

Tong, S., Zhang, J., Li, W., Wang, Y., Kang, F., 2023. An image-based system for locating pruning points in apple trees using instance segmentation and RGB-D images. Biosyst. Eng. 236, 277–286. https://doi.org/10.1016/j.biosystemseng.2023.11.006.

Wang, C., Han, Q., Zhang, T., Li, C., Sun, X., 2024a. Litchi picking points localization in natural environment based on the Litchi-YOSO model and branch morphology reconstruction algorithm. Comput. Electron. Agric. 226, 109473. https://doi.org/10.1016/j.compag.2024.109473.

Wang, J., Lin, X., Luo, L., Chen, M., Wei, H., Xu, L., Luo, S., 2024b. Cognition of grape cluster picking point based on visual knowledge distillation in complex vineyard environment. Comput. Electron. Agric. 225, 109216. https://doi.org/10.1016/j.compag.2024.109216.

Wang, J., Tan, D., Sui, L., Guo, J., Wang, R., 2024c. Wolfberry recognition and picking-point localization technology in natural environments based on improved Yolov8n-Pose-LBD. Comput. Electron. Agric. 227, 109551. https://doi.org/10.1016/j.compag.2024.109551.

You, A., Parayil, N., Krishna, J.G., Bhattarai, U., Sapkota, R., Ahmed, D., Whiting, M., Karkee, M., Grimm, C.M., Davidson, J.R., 2023. Semiautonomous precision pruning of upright fruiting offshoot orchard systems: an integrated approach. IEEE Rob. Autom. Mag. 30, 10–19. https://doi.org/10.1109/MRA.2023.3309098.

Yu, C.h., Shi, X., Luo, W., Feng, J., Zheng, Z., Yorozu, A., Hu, Y., Guo, J., 2024. MLG-YOLO: a model for real-time accurate detection and localization of winter jujube in complex structured orchard environments. Plant Phenomics 6, 0258. https://doi.org/10.34133/plantphenomics.0258.

Zhang, Q., Zhang, H., Gui, Y., Geng, S., Liu, Z., Liu, Y., Liu, Z., 2024. Phenological growth stages of a new kiwifruit cultivar (Actinidia deliciosa 'Jinfu'). Sci. Hortic. (Amsterdam) 327, 112795. https://doi.org/10.1016/j.scienta.2023.112795.

Zhang, Y., Li, N., Zhang, L., Lin, J., Gao, X., Chen, G., 2025. A review on the recent developments in vision-based apple-harvesting robots for recognizing fruit and picking pose. Comput. Electron. Agric. 231, 109968. https://doi.org/10.1016/j.compag.2025.109968.

Zhou, Y., Wei, Y., 2025. UAV-DETR: an enhanced RT-DETR architecture for efficient small object detection in UAV imagery. Sensors 25, 4582. https://doi.org/10.3390/s25154582.