frontiers | Frontiers in Sustainable Food Systems

TYPE Original Research
PUBLISHED 06 June 2024
DOI 10.3389/fsufs.2024.1403872

OPEN ACCESS

EDITED BY
Raul Avila-Sosa,
Benemérita Universidad Autónoma de Puebla, Mexico

REVIEWED BY
Jieli Duan,
South China Agricultural University, China
Zhenguo Zhang,
Xinjiang Agricultural University, China

*CORRESPONDENCE
Zilin Xia
✉ xiazilin@stmail.ujs.edu.cn

RECEIVED 20 March 2024
ACCEPTED 13 May 2024
PUBLISHED 06 June 2024

CITATION
Tang S, Xia Z, Gu J, Wang W, Huang Z and Zhang W (2024) High-precision apple recognition and localization method based on RGB-D and improved SOLOv2 instance segmentation.
Front. Sustain. Food Syst. 8:1403872.
doi: 10.3389/fsufs.2024.1403872

High-precision apple recognition and localization method based on RGB-D and improved SOLOv2 instance segmentation

Shixi Tang[1,2], Zilin Xia[3]*, Jinan Gu[3], Wenbo Wang[3], Zedong Huang[3] and Wenhao Zhang[3]

[1]School of Information Engineering, Yancheng Teachers University, Yancheng, China, [2]Jiangsu Engineering Laboratory of Cyberspace Security, Suzhou, China, [3]School of Mechanical Engineering, Jiangsu University, Zhenjiang, China

Intelligent apple-picking robots can significantly improve the efficiency of apple picking, and the realization of fast and accurate recognition and localization of apples is the prerequisite and foundation for the operation of picking robots. Existing apple recognition and localization methods primarily focus on object detection and semantic segmentation techniques. However, these methods often suffer from localization errors when facing occlusion and overlapping issues. Furthermore, the few instance segmentation methods are also inefficient and heavily dependent on detection results. Therefore, this paper proposes an apple recognition and localization method based on RGB-D and an improved SOLOv2 instance segmentation approach. To improve the efficiency of the instance segmentation network, the EfficientNetV2 is employed as the feature extraction network, known for its high parameter efficiency. To enhance segmentation accuracy when apples are occluded or overlapping, a lightweight spatial attention module is proposed. This module improves the model position sensitivity so that positional features can differentiate between overlapping objects when their semantic features are similar. To accurately determine the apple-picking points, an RGB-D-based apple localization method is introduced. Through comparative experimental analysis, the improved SOLOv2 instance segmentation method has demonstrated remarkable performance. Compared to SOLOv2, the F1 score, mAP, and mIoU on the apple instance segmentation dataset have increased by 2.4, 3.6, and 3.8%, respectively. Additionally, the model's Params and FLOPs have decreased by 1.94M and 31 GFLOPs, respectively. A total of 60 samples were gathered for the analysis of localization errors. The findings indicate that the proposed method achieves high precision in localization, with errors in the *X*, *Y*, and *Z* axes ranging from 0 to 3.95mm, 0 to 5.16mm, and 0 to 1mm, respectively.

KEYWORDS

apple instance segmentation, lightweight spatial attention, EfficientNetv2, improved SOLOv2, RGB-D

1 Introduction

Currently, apple picking relies largely on manual labor, which is time-consuming and labor-intensive, resulting in high harvesting costs and low efficiency. With the rapid development of artificial intelligence and robotics, the realization of automated apple picking has become an inevitable trend (Wang et al., 2022, 2023). Achieving rapid and accurate apple identification and localization in complex orchard environments is the key to realizing

automated apple harvesting. However, the complex environment of orchards is influenced by shading, overlapping, and camera angles, making fast and accurate apple identification and positioning a greater challenge.

In recent years, along with the development of machine vision, the recognition and localization of apples have been extensively researched (Xia et al., 2022, 2023; Gai et al., 2023). Specific methods can be classified into three main categories: object detection-based (Huang et al., 2017; Hu et al., 2023), semantic segmentation-based (Jia et al., 2022b), and instance segmentation-based (Wang and He, 2022a). Object detection involves identifying and localizing objects in images and marking them with bounding boxes. Jia et al. (2022a) proposed an improved FoveaBox (Kong et al., 2020) for green apple object detection. This approach utilizes EfficientNetV2s as the feature extraction network and employs the BiFPN (Bidirectional Feature Pyramid Network) (Tan et al., 2020) for feature fusion. It utilizes the ATSS (Adaptive Training Sample Selection) (Zhang et al., 2020) technique to match positive and negative samples. The overall model achieves a high recall but with reduced speed. Wu et al. (2021) presented an enhanced YOLOV4 method for complex scene apple detection. They replaced YOLOV4's backbone feature extraction network with EfficientNet and achieved a 96.54% F1 score on the constructed dataset. Chen et al. (2021) proposed a Des-YOLOV4 detection model tailored for apples. This method introduces the DenseNet dense residual structure into YOLOV4 and employs Soft-NMS in the post-processing phase to enhance the recall rate of overlapping apples. The overall model has fewer parameters compared to YOLOV4. Apple recognition and localization based on object detection methods are faster. But when the apples in the detection bounding box are obscured or overlapped, it will hinder the acquisition of their depth information and lead to picking failure.

Semantic segmentation can segment each pixel in an image into corresponding categories, yielding more refined object segmentation results. Ahmad et al. (2018) proposed a method based on a fuzzy inference system and fuzzy c-means to achieve the segmentation of apples with different colors during the growth process. Zou et al. (2022) introduced a color-index-based apple segmentation method that enables rapid segmentation of orchard apples, with an average segmentation time of 20 ms. While these traditional segmentation methods offer faster speed, their robustness is compromised when facing complex orchard environments. Kang and Chen (2019) introduced the DasNet, a deep learning-based semantic segmentation network, to achieve the segmentation of apples and tree branches. Li et al. (2021) proposed an improved U-Net (Ronneberger et al., 2015) method for segmenting green apples. It incorporated dilated convolutions and the ASPP (Atrous Spatial Pyramid Pooling) (Chen et al., 2017) structure into U-Net, which enlarged the receptive field and enhanced segmentation accuracy. Using semantic segmentation methods for apple segmentation can provide more detailed contours. However, in cases of overlapping apples, distinguishing between them becomes challenging with semantic segmentation, which, in turn, impacts the acquisition of depth information for each apple.

Instance segmentation enables the classification of each pixel's category in an image while distinguishing different instances of the same category. Kang and Chen (2020) proposed the DaSNet-V2 method for apple instance segmentation, using ResNet101 and ResNet18 as backbone feature extraction networks, achieving segmentation accuracies of 87.3 and 86.6%, respectively. Wang and He (2022b) introduced an improved Mask R-CNN method for apple instance segmentation. By incorporating attention mechanisms in the feature extraction module, this approach enhances apple segmentation accuracy, but at a slower speed. Jia et al. (2020) presented an enhanced Mask R-CNN method for apple instance segmentation. They combined the DenseNet dense connection structure into the ResNet backbone feature extraction network, thus improving segmentation accuracy and enabling recognition and segmentation of overlapping apples. Jia et al. (2021) proposed an anchor-free instance segmentation method tailored for green apples. This method adds an instance branch to FoveaBox, conducting apple detection before segmentation. Nevertheless, it exhibits subpar performance in segmenting apple edge contours. Instance segmentation-based methods can achieve apple recognition, precise localization, and mask generation. However, the majority of current research focuses on detection-based instance segmentation approaches. In these methods, the instance branch often lacks consideration of global context, resulting in suboptimal performance in edge segmentation and slower segmentation speeds.

Acquiring depth information for apples is a critical factor in achieving accurate picking. Specific means of obtaining this information include stereo cameras, structured light cameras, TOF (time-of-flight) cameras, and laser radar. Tian et al. (2019) proposed a fruit localization technique based on Kinect V2, utilizing depth images to determine the apple's center and combining RGB data to estimate the apple's radius. But, in cases of overlap and occlusion, the depth image may not fully represent the apple's true depth information, leading to ambiguous localization. Kang et al. (2020) implemented apple localization using an Intel D-435 camera. They employed RGB images for fruit detection and instance segmentation, combining depth information to fit apple's point cloud, thus localizing it. However, this method suffers from lower efficiency. Gené-Mola et al. (2019) utilized laser radar and object detection for apple localization, achieving a success rate of 87.5%. Kang et al. (2022) fused radar with the camera as input and then used instance segmentation to achieve apple localization, but this method incurs higher costs.

So far, the recognition and localization of apples have predominantly relied on object detection and semantic segmentation methods. However, these methods often lead to positioning errors when facing challenges such as occlusion and overlapping. While a few studies have explored detection-based instance segmentation methods for apple recognition and localization, these methods usually come with high parameter and computational complexity, are susceptible to the influence of detection results, and lack consideration of global information. SOLOv2 (Wang et al., 2020b) is a one-stage instance segmentation method that introduces an efficient instance mask representation scheme based on the foundation of SOLO (Wang et al., 2020a). It improves the efficiency of the overall method by decoupling instance mask generation into mask kernel and mask feature learning and utilizing convolutional operations to generate instance masks. Compared to two-stage instance segmentation models like MaskRCNN, SOLOv2 eliminates the need for anchor boxes, does not rely on detection results, occupies less memory, and is more suitable for practical engineering applications. Therefore, this paper proposes an apple recognition and localization approach based on RGB-D and an improved SOLOV2 instance segmentation method. This method eliminates reliance on detection results and can achieve accurate apple positioning even in occlusion and overlapping

scenarios. Specifically, the main contributions of this paper are as follows:

1 Introducing an improved SOLOV2 instance segmentation method that achieves high-precision apple instance segmentation and is independent of detection results.
2 Introducing a lightweight spatial attention mechanism into the mask prediction head of SOLOV2 to enhance the segmentation accuracy for overlapping apples.
3 Introducing an RGB-D-based apple localization method that achieves accurate positioning in scenarios with occlusion and overlapping, thereby enhancing the success rate of apple picking.

The sections of this paper are organized as follows: section 2 introduces the improved SOLOv2 instance segmentation method and the RGB-D-based apple-picking point localization method. Section 3 conducts comparative experiments and analyzes the experimental results. Section 4 summarizes the entire paper and outlines future research directions.

2 The proposed method

2.1 Apple instance segmentation method based on improved SOLOv2

In this paper, we further enhance the segmentation accuracy based on SOLOv2 without introducing excessive parameters. Specifically, we integrate the proposed lightweight spatial attention module into the mask kernel and mask feature branches and adopt a more efficient feature extraction network, EfficientNetV2. After these improvements, the improved SOLOv2 significantly boosts instance segmentation accuracy while maintaining efficiency and avoiding the introduction of redundant parameters. Figure 1 illustrates the enhanced SOLOv2 instance segmentation method, and detailed descriptions of each module will be elaborated in the subsequent sections.

2.1.1 Backbone feature extraction network

The feature extraction network, as a crucial component of the instance segmentation method, significantly influences the performance of the whole model. In this study, EfficientNetV2 (Tan and Le, 2021) was adopted as the backbone feature extraction network, building upon the improvements made in EfficientNetV1 (Tan and Le, 2019). The network's optimal width, height, and other design parameters were determined using NAS (Neural Architecture Search) techniques. To address the slow training speed of EfficientNetV1, the shallow MBConv modules were substituted with Fused-MBConv modules, with the specific MBConv and Fused-MBConv modules illustrated in Figure 1.

As depicted in Figure 1, the MBConv module employs a 1×1 convolutional layer to increase feature dimensionality, followed by a 3×3 depthwise separable convolutional layer for feature extraction. In contrast, the Fused-MBConv module directly utilizes a 3×3 convolutional layer to perform feature extraction and dimensionality expansion, improving feature extraction speed. EfficientNetV2 demonstrates exceptional accuracy on the ImageNet dataset while enhancing training speed and parameter efficiency. Compared to ResNet50, EfficientNetV2 exhibits higher efficiency, achieving greater precision with equivalent parameters and computation. Additionally, EfficientNetV2 is well-suited for mobile and embedded device deployment for tasks such as apple harvesting.

2.1.2 Instance mask generation module

SOLOv2 decouples the instance mask generation into mask kernels and mask features. Then, it utilizes convolution between the mask kernels and mask features to obtain the final instance mask. The parameters of the mask kernels and mask features are generated separately through the mask kernel branch and the mask feature branch.

As depicted in Figure 1, the first step is utilizing the FPN (Lin et al., 2017) to perform multi-scale feature fusion, aiming to achieve multi-scale segmentation. This process is detailed in the following Eq. 1.

$$P_2,P_3,P_4,P_5,P_6 = \mathrm{FPN}\left(C_2,C_3,C_4,C_5\right) \tag{1}$$

where C_2,C_3,C_4,C_5 are the effective feature layers output by EfficientNetV2, and P_2,P_3,P_4,P_5,P_6 are the feature layers output after feature fusion.

In the mask kernel branch, each feature layer P_i is sampled to a grid of size S_i, and if the center of the GT falls into this grid, it indicates that this grid is responsible for predicting its instances. Specifically as shown in Eq. 2.

$$K_i = f_{\text{Kernel Branch}}\left(P_i\right), i = 2,3,4,5,6 \tag{2}$$

K_i is the mask kernel parameter generated by the corresponding feature layer with size $S_i \times S_i \times C$.

In the mask feature branch, FPN output features are used to create shared mask features across different levels. This approach allows different levels to share the same mask features, reducing parameters and improving efficiency. The process is detailed in Eq. 3 below.

$$F = f_{\text{Feature Branch}}\left(P_2, P_3, P_4, P_5\right) \tag{3}$$

where F denotes the shared mask features, with sizes of $H\times W\times C$. H and W are one-fourth the size of the input height and width, respectively.

Finally, the mask kernel parameters corresponding to the grids containing objects, denoted as K_i^{pos}, are selected. These are the grids where the center of the GT falls during training and the grids where the predicted classification score is greater than the score threshold during inference. The selected mask kernel parameters K_i^{pos} are utilized to convolve with the shared mask features F to generate instance masks. As shown in the following Eq. 4.

$$M_{i,j} = K_i^{\text{pos}} \circledast F \tag{4}$$

K_i^{pos} is the mask kernel parameter obtained by filtering in K_i with size $n\times1\times1\times\mathrm{C}$, and $M_{i,j}$ is the instance prediction mask generated at the corresponding location. The overall instance mask generation module is illustrated in Figure 2.

2.1.3 Improved mask feature branch

In SOLOv2, the mask feature branch is composed solely of upsampling and convolution operations. However, instances with

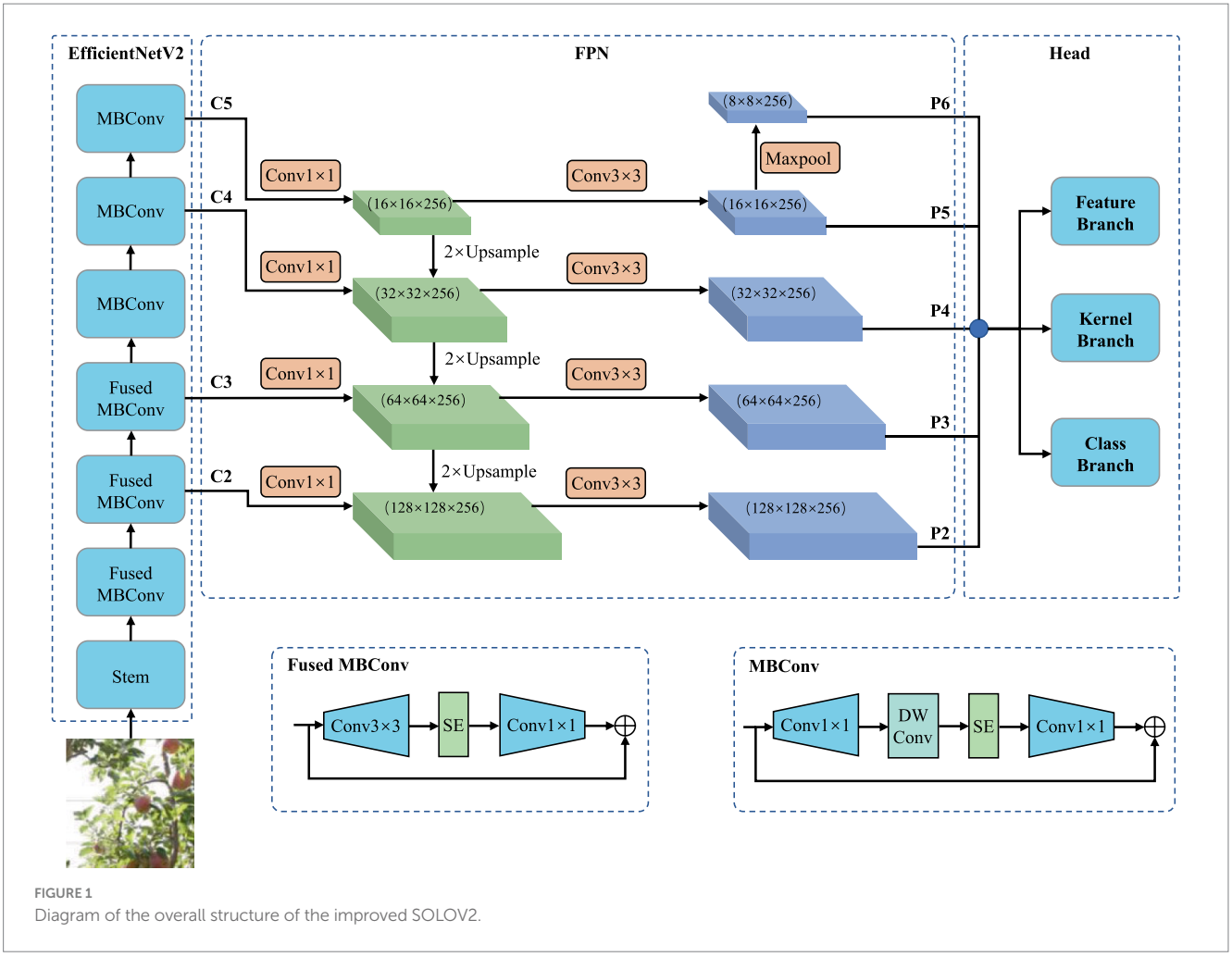

FIGURE 1
Diagram of the overall structure of the improved SOLOV2.

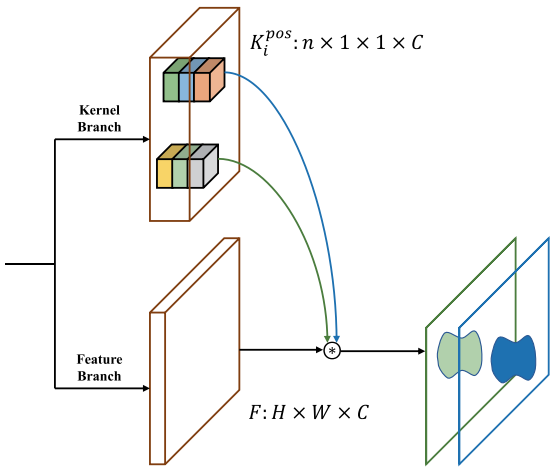

FIGURE 2
Instance mask generation module structure diagram.

similar semantic features heavily rely on positional features for differentiation. Relying solely on convolution-generated positional information is insufficient. Therefore, an Att-Block is utilized as a replacement for the convolution operation to construct mask features containing comprehensive positional information without introducing excessive parameters. In the Att-Block, the convolution is replaced with a depthwise separable convolution and a lightweight spatial attention mechanism is introduced to capture positional features between instances. The details are illustrated in Figure 3.

The lightweight spatial attention module is divided into two steps: (1) first, obtain the corresponding spatial position relationship in the vertical direction by using a $K \times 1$ convolutional kernel on the feature map. The computational complexity of this step is H^2W; (2) then obtain the corresponding spatial position relationship in the horizontal direction by using a $1 \times K$ convolutional kernel on the feature map generated in step (1), the computational complexity of this step is HW^2. Finally, use Sigmoid to generate the spatial attention map, the overall computational complexity is $H^2W + HW^2$, compared with directly using a fully connected layer to calculate the spatial attention map of the feature map, the lightweight attention module has lower computational complexity when the feature map width W and height H are large. This makes it particularly suitable for capturing feature map spatial relationships in lightweight networks.

2.1.4 Improved mask kernel branch

With the aim of enhancing the sensitivity of learned mask kernel parameters to positional information and improving instance segmentation accuracy, this paper introduces a modification to the mask kernel branch. The convolution operations in the mask kernel branch are

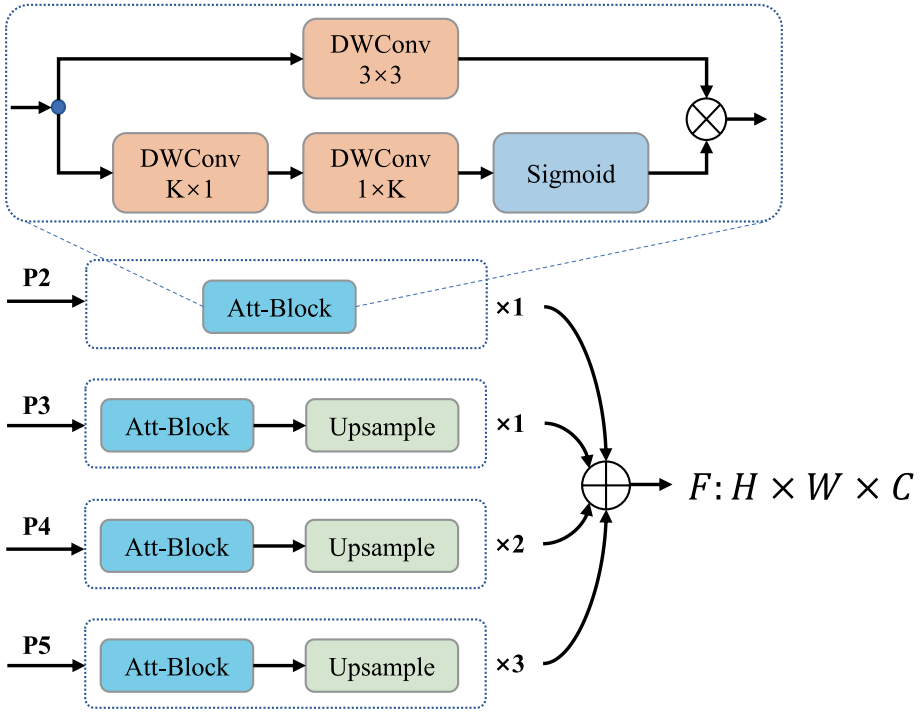

FIGURE 3
The structure diagram of the improved mask feature branch.

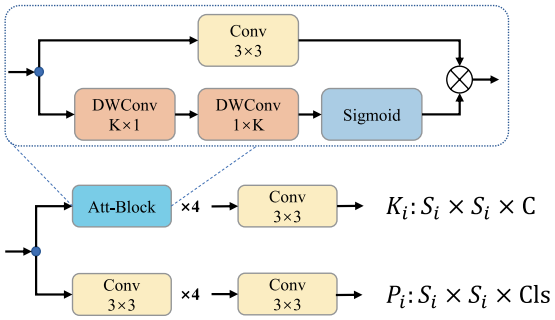

FIGURE 4
The structure diagram of the improved mask kernel branch.

replaced with Att-Block modules to capture feature spatial relationships. This alteration enables the learned mask kernel parameters to encompass richer positional information, as depicted in Figure 4. It is important to note that the Att-Block used in the improved mask kernel branch employs regular convolutional structures rather than depthwise separable convolutions. This choice aims to ensure that the encoded information within the mask kernel is more comprehensive.

2.1.5 Label assignment method and loss calculation

SOLOV2 differs from detection-based instance segmentation methods in that it does not assign labels by IoU thresholding. It resizes different feature layers into $S \times S$ grids of different sizes, and each element in the grid is responsible for predicting one instance. Given an image where GT represents the ground truth labels, $\mathrm{GT}_{\mathrm{area}}$ denotes the area of the label, $\mathrm{GT}_{\mathrm{mask}}$ represents the mask of the label, and $\mathrm{GT}_{\mathrm{label}}$ indicates the category of the label. Firstly, the ground truth instances are categorized into different levels based on their area. Specifically as shown in Eq. 5.

$$lb_i \leq \mathrm{GT}_{\mathrm{area}} \leq up_i \tag{5}$$

where lb_i and up_i represent the lower and upper bounds of the object scale predicted by the current feature layer, if instances satisfy this condition, are considered as GT_i for the current layer. Subsequently, GT_i is scaled around its center, and the grid cells within the scaled GT_i are selected as positive samples, as shown in the following Eq. 6.

$$\mathrm{pos}_i^{\mathrm{index}} = \mathrm{GT}_i * \mathrm{pos}_{\mathrm{scale}} \tag{6}$$

where $\mathrm{pos}_i^{\mathrm{index}}$ represents the indices of grids within the scaled GT_i, which are the indexes of positive samples; $\mathrm{pos}_{\mathrm{scale}}$ is the scaling factor. Then, the mask kernel parameters corresponding to the positive samples are selected using these indices and denoted as K_i^{pos}. Specifically as shown in Eq. 7.

$$K_i^{\mathrm{pos}} = K_i\left[\mathrm{pos}_i^{\mathrm{index}}\right] \tag{7}$$

The mask kernel parameters corresponding to positive samples from all layers are collected and denoted as K_{pos}. Then, convolution is applied to obtain the predicted masks. Specifically as shown in Eq. 8.

ALGORITHM 1 The label assignment method and loss calculation in SOLOv2

Input: lb, up, GT, *G*, *F*, *K*, *P*, *L*, $\text{pos}_{\text{scale}}$

lb, up: The lower and upper bounds of the object scale predicted by each feature level are [(1, 96), (48, 192), (96, 384), (192, 768), (384, 2048)].

GT: Ground truth in images.

F: Mask feature with the size of $H \times W \times C$.

K: The set of mask kernel predictions with size $S \times S \times C$.

P: The set of classification scores, whose size is $S \times S \times$ cls.

L: The number of feature pyramid levels.

$\text{pos}_{\text{scale}}$: GT scaling factor, after scaling its interior is positive samples.

Output: L_{mask}, L_{cls}

L_{mask}: Mask loss.

L_{cls}: Classification loss

for each level: $i \in [1, L]$ do

$\text{GT}_i \leftarrow \text{lb}_i \leq \text{GT}_{\text{area}} \leq \text{up}_i$

$\text{pos}_i^{\text{index}} \leftarrow \text{GT}_i * \text{pos}_{\text{scale}}$ GT inner grid after scaling

$K_i^{\text{pos}} \leftarrow K_i[\text{pos}_i^{\text{index}}]$ Positive sample grid mask kernel parameters

$\text{Target}_{\text{label}}^i \leftarrow \text{pos}_{\text{index}}^i$ is at the corresponding GT_{label}, the others are 0.

$\text{Target}_{\text{mask}}^i \leftarrow$The GT_{mask} corresponding $\text{pos}_{\text{index}}^i$

$K_{\text{pos}} \leftarrow K_{\text{pos}} \cup K_i^{\text{pos}}$

$\text{Target}_{\text{label}} \leftarrow \text{Target}_{\text{label}} \cup \text{Target}_{\text{label}}^i$

$\text{Target}_{\text{mask}} \leftarrow \text{Target}_{\text{mask}} \cup \text{Target}_{\text{mask}}^i$

end for

$M \leftarrow K_{\text{pos}} \circledast F$

$L_{\text{mask}} = \text{Diceloss}(M, \text{Target}_{\text{mask}})$

$L_{\text{cls}} = \text{Focalloss}(P, \text{Target}_{\text{label}})$

return L_{mask}, L_{cls}

$$M = K_{\text{pos}} \circledast F \tag{8}$$

where F is the mask feature generated by the mask branch, and M is the prediction mask. Finally, the mask and classification losses are computed as follows in Eqs. 9, 10.

$$L_{\text{mask}} = \text{Diceloss}\left(M, \text{Target}_{\text{mask}}\right) \tag{9}$$

$$L_{\text{cls}} = \text{Focalloss}\left(P, \text{Target}_{\text{label}}\right) \tag{10}$$

L_{mask} is the mask loss, specifically Diceloss, where $\text{Target}_{\text{mask}}$ means that the index of positive samples corresponds to GT_{mask}, and the negative samples do not participate in the calculation of the mask loss. L_{cls} is the classification loss, specifically Focalloss, where P is the classification prediction value, and $\text{Target}_{\text{label}}$ means that the positive samples correspond to GT_{label}, and the negative samples are 0. Both positive and negative samples contribute to the calculation of the classification loss. The overall loss function is formulated as shown in Eq. 11.

$$L_{\text{total}} = \lambda_1 L_{\text{cls}} + \lambda_2 L_{\text{mask}} \tag{11}$$

where L_{total} is the total loss, λ_1 and λ_2 are the weights of classification loss and mask loss, which take the values of 1.0 and 3.0 in this paper, respectively. The overall training label assignment method and loss calculation can be seen in Algorithm 1.

2.2 RGB-D-based apple localization method

To achieve precise apple localization, especially in scenarios with occlusion and overlapping, this paper proposes an RGB-D-based apple localization method. The method begins by employing the enhanced SOLOv2 apple instance segmentation method to obtain masks for apples in the images. Subsequently, these masks are combined with the depth maps generated by an RGB-D camera to accurately locate the points where apples can be picked. The overall workflow is depicted in Figure 5, with the following steps.

Step 1: Instance segmentation.

Perform segmentation on the RGB images to obtain apple masks.

Step 2: Finding the minimum enclosing circle of the mask.

Utilize OpenCV to compute the minimum enclosing circle of the segmented apple mask. This step aims to ensure a better fit of the mask to the apple, avoiding excessive inclusion of background information.

Step 3: Calculating mask and minimum enclosing circle IoU.

To ensure that the pixel information of the apple is as complete as possible, thereby enhancing the success rate of picking, compute the IoU to filter out apples that are viable for picking in the current view. A higher IoU value indicates fewer obscured parts of the apple. This paper adopts an IoU threshold of 0.5.

Step 4: Confirming if the central point of the minimum enclosing circle belongs to the apple.

Select the center point of the minimum enclosing circle of the apple mask as the picking point. To do so, verify whether the pixel coordinates of the circle's center point correspond to the apple. If leaves or branches potentially obstruct the point, picking is not viable from the current viewpoint.

Step 5: Calculate picking point coordinates.

If steps 3 and 4 are satisfied, it indicates that the viewpoint allows picking. Using pixel coordinates along with the corresponding depth information and camera intrinsic allows calculating the three-dimensional coordinates (x, y, z) of the picking point in the camera coordinate system. Specifically as shown in Eqs. 12,13.

$$x = z \times \frac{u - u_0}{f_x} \tag{12}$$

$$y = z \times \frac{v - v_0}{f_y} \tag{13}$$

where (u, v) represents the pixel coordinates of the center of the minimum enclosing circle in the X and Y directions, z indicates the depth information of the circle center, and u_0, v_0, f_x, and f_y are the camera intrinsic.

3 Experiments

3.1 Dataset

The apple instance segmentation dataset constructed in this paper consists of two parts. One part is the public dataset, which includes 3,925 apple images annotated with instance labels (Gené-Mola et al., 2023). This dataset covers two growth stages of apples, with approximately 70% at the growth stage where apples are primarily green, as shown in Figure 6A. The remaining approximately 30% are at the ripening stage, where apples are mostly light red, as shown in Figure 6B.

The other part of the dataset is collected from orchards, consisting of 300 apple images and annotated with instance labels using the Labelme tool. These images were captured during the ripe stage of apples, characterized by their red color, as illustrated in Figure 6C.

Lastly, an 8:2 data split ratio was employed to ensure the effective utilization of training data. It means that 80% of the data were used for training and validation, totaling 3,400 images, while the remaining 20% were reserved for testing, comprising 852 images. Such a division aims to avoid overfitting, thereby improving the generalization ability and robustness of the model.

3.2 Experimental setting

The hardware setup for the experiments in this study included an E5-2678 V3 CPU, 32GB of RAM, and an NVIDIA 3090 GPU with 24GB of VRAM. The software system used was Ubuntu 18.04, with Python version 3.8. The deep learning framework employed was PyTorch. Pretrained weights were utilized for the backbone feature extraction networks to expedite model convergence. The training configuration encompassed 40 epochs with a batch size of 4. The SGD optimizer was used with an initial learning rate of 0.01. Learning rate

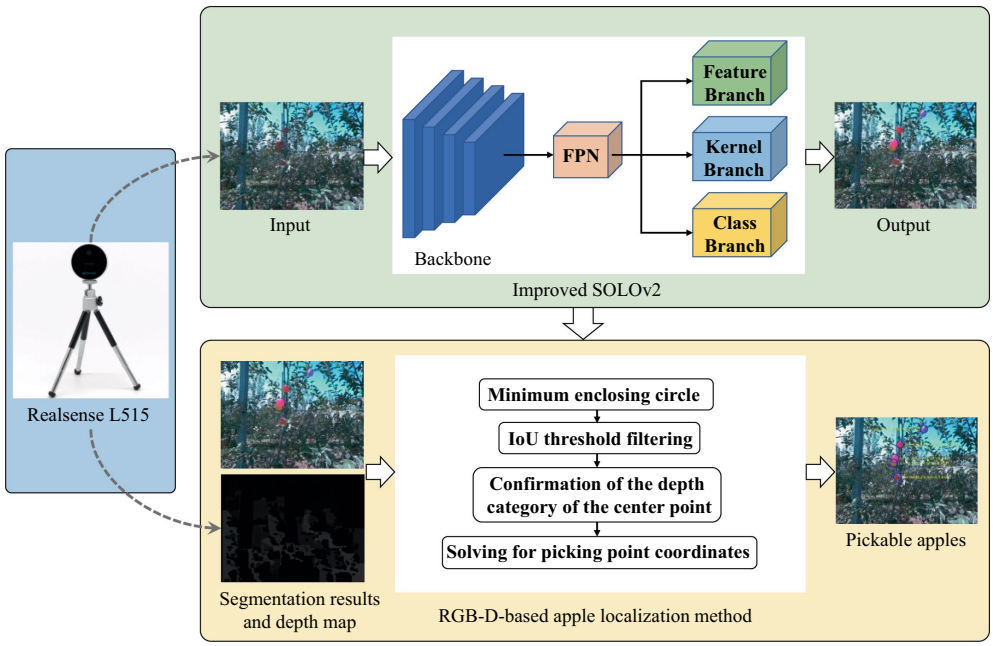

FIGURE 5
Flowchart of RGB-D based apple localization method.

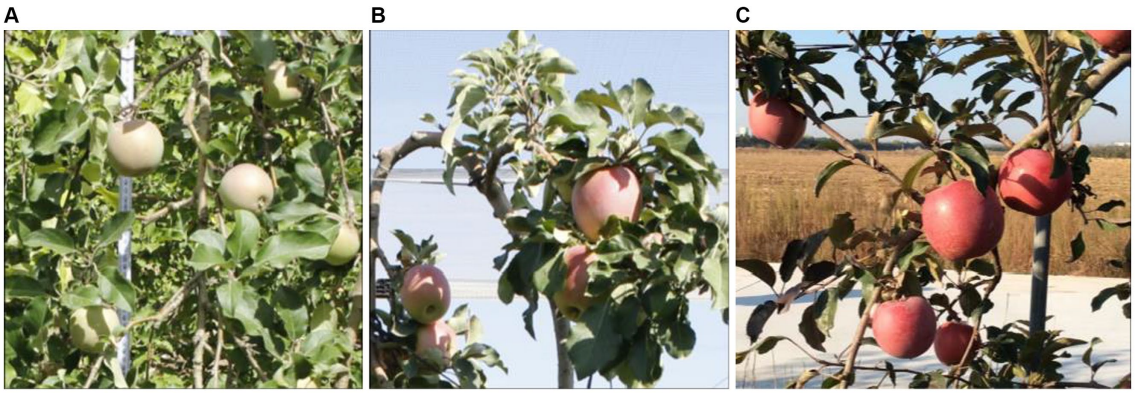

FIGURE 6
Samples of apple instance segmentation dataset.

adjustments were applied using the StepLR strategy, where the learning rate was reduced by 0.1 at the 16th and 32th epochs, respectively. To accelerate model convergence, the weights of the backbone for all models were initialized using pre-trained weights on ImageNet-1K. The specific experimental settings are shown in Table 1.

3.3 Evaluation metrics

In order to evaluate the performance of the proposed method, AP (average precision), mAP (mean average precision), mIoU (mean intersection over union), and F1 scores are used to measure the accuracy, and Params (parameters), FLOPs (floating-point operations), and FPS (frames per second) are used to measure the model complexity. The calculation formula is shown below.

$$\text{Precision} = \frac{\text{TP}}{\text{all detections}} \tag{14}$$

$$\text{Recall} = \frac{\text{TP}}{\text{all GTBox}} \tag{15}$$

$$\text{AP} = \int_{1}^{0} p(r)\,\mathrm{d}r \tag{16}$$

$$\text{mAP} = \frac{\sum \text{AP}}{N} \tag{17}$$

TABLE 1 Experimental parameter settings.

Hyperparameters	Setting	
Batch size	4	
Epoch	40	
Learning rate	Epoch 1–16	0.01
	Epoch 16–32	0.01*0.1
	Epoch 32–40	0.01*0.01
Optimizer	SGD	

$$F1 = 2 \times \frac{Precision \,.\, Recall}{Precision + Recall} \tag{18}$$

$$mIoU = \sum \frac{TP}{TP + FP + FN} \tag{19}$$

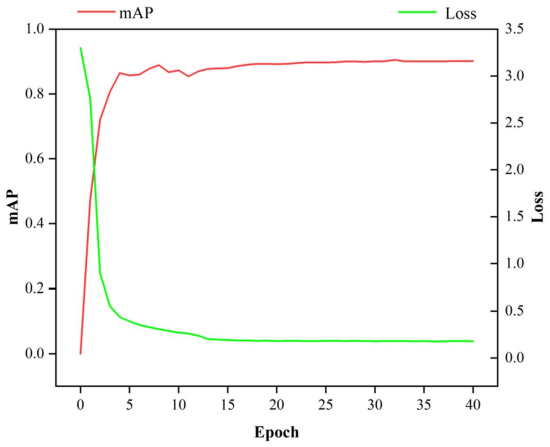

FIGURE 7
Training loss curve and mAP curve of the improved SOLOv2 model.

where TP denotes the number of correctly detected targets among all detected targets, FP denotes the number of incorrectly detected targets among all detected targets, FN indicates the number of incorrectly classified negative samples, $p(r)$ stands for the Precision-Recall curve, and N represents the number of categories in the dataset.

FLOPs and Params are critical metrics for evaluating model complexity and speed. FLOPs measure the amount of computation, and Params indicate the number of learnable parameters in the network. A larger computational and parameter count typically results in higher model complexity and slower detection speed. Therefore, a model suitable for edge devices such as apple picking in orchards should have fewer parameters and lower computational burden.

3.4 Experimental results of the improved method

The improved SOLOv2 is trained on the constructed apple instance segmentation dataset, and model evaluation is performed every epoch. The training loss curve and the test set mAP curve are shown in Figure 7, where red represents the mAP curve and green represents the loss curve.

As shown in Figure 7, the model's loss value gradually decreases and stabilizes as the training progresses, while the mAP metric steadily increases. It indicates that the model is progressively converging. Selecting the weights from the last epoch as the final result, the mAP on the test set of the apple instance segmentation dataset reaches 90.1%. Demonstrates that the proposed method achieves high precision and recall in apple instance segmentation tasks, and the model's overall performance is excellent.

3.5 Comparative experiments with other instance segmentation methods

To verify the effectiveness and advancement of the proposed method, it will be compared to other mainstream instance segmentation methods, specifically including the original SOLOv2 method before improvement, the one-stage instance segmentation method Yolact (Bolya et al., 2019), and the two-stage instance segmentation method MaskRCNN (He et al., 2020) and MS-RCNN (Huang et al., 2019). The mAP, mIoU and F1 scores of various segmentation models are depicted in Figure 8. It can be observed that, compared to other segmentation models, the improved SOLOv2 achieves the highest scores.

According to the results in Table 2, the improved SOLOv2 instance segmentation model performs best in the F1 score, mIoU, and mAP metrics, reaching 88.5, 83.2, and 90.1%, respectively. Compared to the original method, these three metrics were improved by 2.4, 3.6, and 3.8%, respectively, highlighting the effectiveness of the improved method. Compared with the two-stage models MaskRCNN and MS-RCNN, the improved SOLOv2 model improved the F1 scores by 0.2 and 0.6%, mIoU by 1.1 and 2.7%, and mAP by 2.3 and 2.1%, respectively. Compared to the one-stage model Yolact, the improved SOLOv2 model significantly improved all metrics, including a 7.9% improvement in mIoU, 2.3, and 4.4% in F1 score and mAP, respectively. These results highlight the superior precision and recall achieved by the proposed method, resulting in more effective instance segmentation.

Furthermore, the improved SOLOv2 apple instance segmentation method has also been optimized for Params, FLOPs, and FPS. Compared to the original method, it reduces Params by 1.94M, FLOPs by 31 GFLOPs, and maintains detection speed almost the same, with a slight decrease of 0.7 frames per second. Compared to MaskRCNN, Params remain similar, but FLOPs decrease by 39 GFLOPs, and FPS increases by 1.3. Compared to MS-RCNN, Params and FLOPs are significantly reduced by 15.94M and 78 GFLOPs, respectively, with FPS increasing by 2.8. Although Yolact performs best in detection speed-related metrics, the proposed method significantly improves segmentation accuracy. Overall, the proposed method strikes a balance between model accuracy and complexity, performing excellently in apple instance segmentation tasks.

Figure 9 displays a comparison of Precision-Recall (P-R) curves for each method within the apple category. The red curve represents the proposed enhanced SOLOv2 instance segmentation method. Notably, the red curve encompasses the largest area, and even at high recall rates, it sustains a remarkable level of accuracy. These findings underscore the enhanced method's ability to attain superior precision

and recall, showcasing improved stability and performance when contrasted with other methods.

Figure 10 illustrates a comparison of segmentation results between the enhanced SOLOv2 and other methods on the test set of the apple instance segmentation dataset. Notably, the improved SOLOv2 maintains accurate segmentation even in scenarios where apples are closely spaced. In Figure 10C, SOLOv2 exhibits segmentation errors when distinguishing overlapping objects, failing to separate the two instances. Moreover, in Figure 10D, MaskRCNN experiences segmentation omission issues with overlapping objects. However, Figure 10B illustrates that these issues were substantially addressed following the improvements. The improved model can accurately segment and differentiate overlapping instances. This further underscores the effectiveness of the proposed lightweight spatial attention module, which excels at distinguishing objects based on their spatial characteristics when semantic features pose challenges in differentiation.

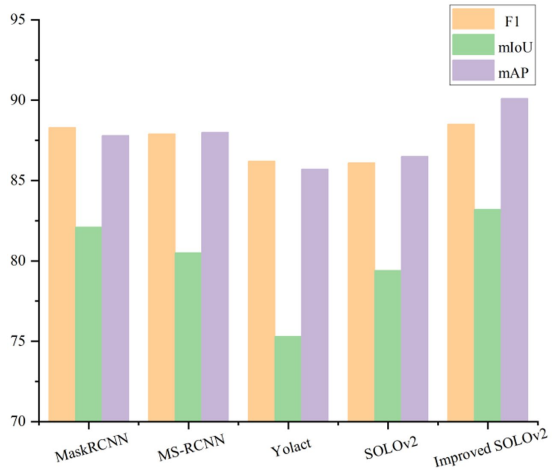

FIGURE 8
Comparison of F1 score, mIoU and mAP of different segmentation models.

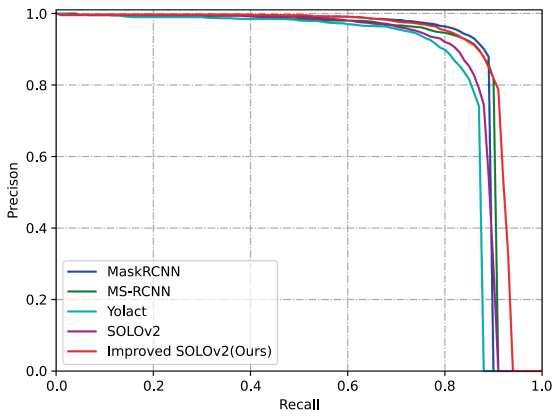

FIGURE 9
Comparison of P-R curves for different segmentation models.

3.6 Ablation study

In order to further validate the impact of improvements on model performance, this section conducts ablation experiments to assess the effectiveness of both the backbone feature extraction network and the lightweight attention module. Firstly, we replace the original ResNet50 in the SOLOv2 backbone feature extraction network with EfficientNetV2 while keeping all other aspects unchanged. This step aims to evaluate how the improved backbone feature extraction network influences model performance. Subsequently, we conduct experiments to individually introduce the proposed lightweight attention module into the mask feature branch, the mask kernel branch, and simultaneously into both branches. These experiments are designed to assess the impact of the proposed lightweight attention module. The results of the specific ablation experiments can be seen in Table 3.

As shown in Table 3, improving the backbone feature network to EfficientNetV2 results in a 0.5% increase in the F1 score and a 0.2% increase in mAP. Additionally, EfficientNetV2's parameter-efficient design enhances the computational efficiency of the model. The performance is improved when introducing the lightweight spatial attention module separately into the mask feature branch and the mask kernel branch. Specifically, adding the attention module to the mask feature branch increases mAP by 1%. Incorporating the attention module into the mask kernel branch results in a 1.2% improvement in the F1 score and a 2.3% improvement in mAP. Simultaneously, adding the attention module to both branches yields even more significant effects, with the F1 score improving by 2.4% and mAP by 3.4%. This unequivocally demonstrates that the proposed lightweight spatial attention module significantly enhances the precision of apple instance segmentation.

3.7 Positioning error analysis

For validation of the localization accuracy of the proposed RGB-D-based apple localization method, 20 sets of RGB images and

TABLE 2 Comparative experimental results of mIoU, mAP, F1 score, Params, FLOPs and FPS for different segmentation models.

Methods	F1 (%)	(%)	mIoU (%)	FLOPs (GFLOPs)	Params (M)	FPS
MaskRCNN	88.3	87.8	82.1	186	43.97	28.2
MS-RCNN	87.9	88.0	80.5	225	60.23	26.7
Yolact	86.2	85.7	75.3	61.427	34.73	51.4
SOLOv2	86.1	86.5	79.4	178	46.23	30.2
Improved SOLOv2	88.5	90.1	83.2	147	44.29	29.5

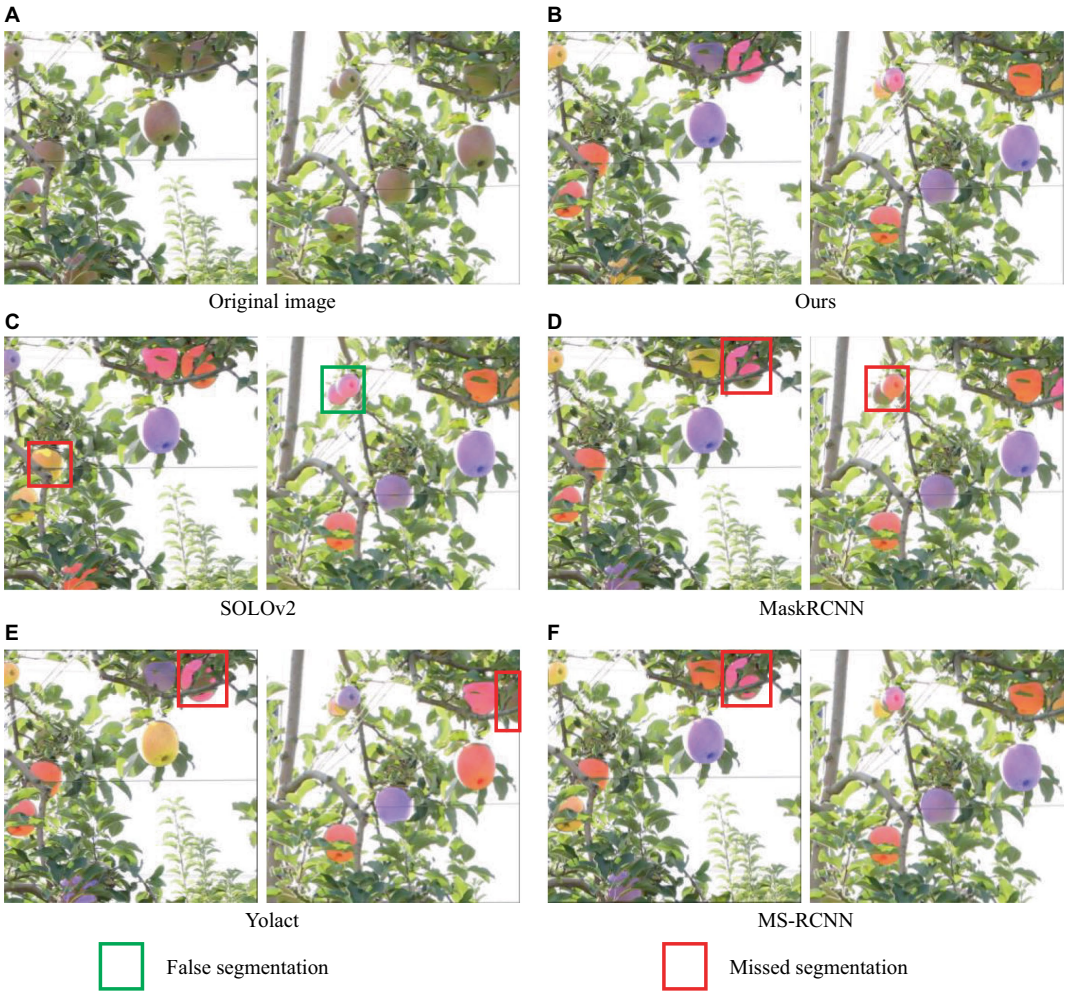

FIGURE 10
Comparison of segmentation results of different segmentation models.

TABLE 3 Ablation experiment results.

Baseline	EfficientNetV2	Att-Block		F1 (%)	mAP (%)
		Mask feature branch	Mask kernel branch		
✓	×	×	×	86.1	86.5
✓	✓	✓	×	86.6	86.7
✓	✓	✓	×	86.6	87.7
✓	✓	×	✓	87.8	89.0
✓	✓	✓	✓	88.5	90.1

Baseline is SOLOv2-ResNet50.

the corresponding depth maps, totaling about 60 apples, are captured using the Realsense L515 depth camera. The true picking point of an apple is defined as the camera's three-dimensional coordinates (x,y,z), obtained by combining the pixel coordinates of the manually annotated center of the largest bounding rectangle around the apple, camera intrinsic parameters, and the corresponding depth information. Subsequently, the improved SOLOv2 instance segmentation method and the depth-based apple localization method are used to derive the predicted three-dimensional coordinates $\left(\hat{x},\hat{y},\hat{z}\right)$ for the apple's estimated picking point. Finally, the error between the predicted and true picking points is calculated to assess the positioning accuracy. Table 4 presents some true picking points, predicted picking points, and their absolute errors. Figure 11 illustrates box plots of the positioning errors in the *X*, *Y*, and *Z* directions for approximately 60 sets of apples.

TABLE 4 The positioning error of some picking points, in which the data unit is mm.

x	y	z	$\hat{x}$	$\hat{y}$	$\hat{z}$	$\lvert x-\hat{x}\rvert$	$\lvert y-\hat{y}\rvert$	$\lvert z-\hat{z}\rvert$
20.04	47.83	733.75	20.04	47.83	733.75	0	0	0
−43.68	−53.3	801	−43.66	−53.3	801	0.02	0	0
109.13	−12.59	892	109.51	−12.58	801	0.38	0.01	0
87.21	−76.57	823.75	89.17	−75.05	823.75	1.96	1.52	0
−132.2	104.1	923.5	−130.1	106.2	923.8	2.1	2.1	0.3
⋮								
113.29	−149.57	791	114.77	−148.24	791	1.48	1.33	0
279.69	75.82	653.75	281.47	74.4	653.75	1.78	1.42	0
−132.23	104.11	923.5	−130.12	106.19	923.75	2.11	2.08	0.25

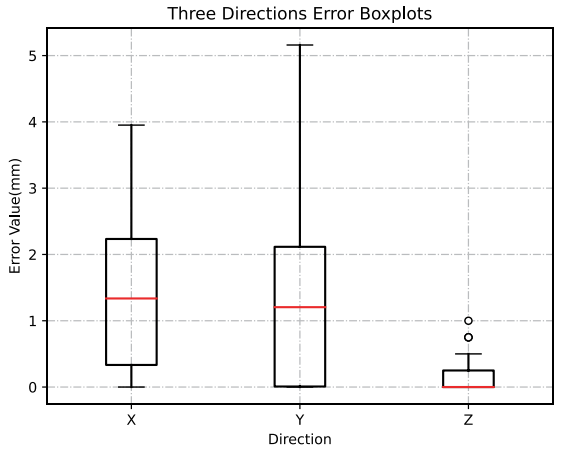

FIGURE 11
X, *Y*, *Z* direction positioning error.

Figure 11 displays the median errors represented by red line segments. The median positioning errors in the *X* and *Y* directions are less than 1.5 mm. Furthermore, the median positioning error in the *Z* direction approaches zero, with a maximum *Z*-direction positioning error of approximately 1 mm. These observations demonstrate that the proposed RGB-D-based apple-picking point localization method attains remarkable precision, fulfilling practical picking needs.

Figure 12 illustrates the process of apple-picking point localization. Figure 12A shows the original image, while Figure 12B displays the instance segmentation result. Figure 12C shows the pickable apples after IoU filtering and confirmation of the depth information of the center point, where the blue circles indicate the pickable apples and the red circles indicate the non-pickable apples. Figure 12D presents the localization results of picking points in the camera coordinate system, obtained by combining depth information and camera intrinsic parameters with coordinates measured in meters. It can be observed from the figure that the proposed RGB-D-based picking point localization method effectively achieves accurate apple localization. Furthermore, when the depth information at the center of the bounding circle of the apple segmentation mask does not correspond to the apple category, the localization method can provide correct feedback.

4 Conclusion

The orchard environment is complex, and detection and segmentation-based methods exhibit lower accuracy in recognizing and localizing overlapping or occluded apples. Detection-based instance segmentation methods heavily rely on detection results and do not consider global features, such as MaskRCNN. Therefore, this study introduces a high-precision method based on RGB-D data and an enhanced SOLOV2 instance segmentation method for orchard apple recognition and picking point localization. This method does not rely on detection results, performs well in the face of occlusion, and can accurately locate the apple picking point. The specific conclusions of this research are outlined below:

(1) An improved SOLOv2 high-precision apple instance segmentation method is introduced. To enhance the efficiency of the instance segmentation network, EfficientNetV2 is adopted as the backbone feature extraction network, which has a highly efficient parameter design. When faced with scenarios involving overlapping or occluded apples, as their semantic features are quite similar, we introduce a lightweight spatial attention module to improve segmentation accuracy. This module can increase position sensitivity, thus distinguishing based on positional features even when semantic features are similar. Through comparative experimental analysis, the improved SOLOv2 instance segmentation method performs exceptionally well, achieving the highest F1 score and mAP values on the apple instance segmentation dataset, 88.5 and 90.1%, respectively. Furthermore, compared to the previous version, the model's parameter count and computational load have slightly decreased by 1.94M and 31 GFLOPs.

(2) To achieve precise apple-picking point localization, an apple localization method based on RGB-D is proposed. Firstly, the pickable apples are filtered by the IoU of the mask and its maximum outer circle and then determine whether the midpoint of the maximum outer circle is an apple category. Finally, the 3D coordinates of the picking point are obtained based on the depth information of the midpoint and the camera's intrinsic parameters. Experimental verification indicates that, in the collection of 60 datasets, the median

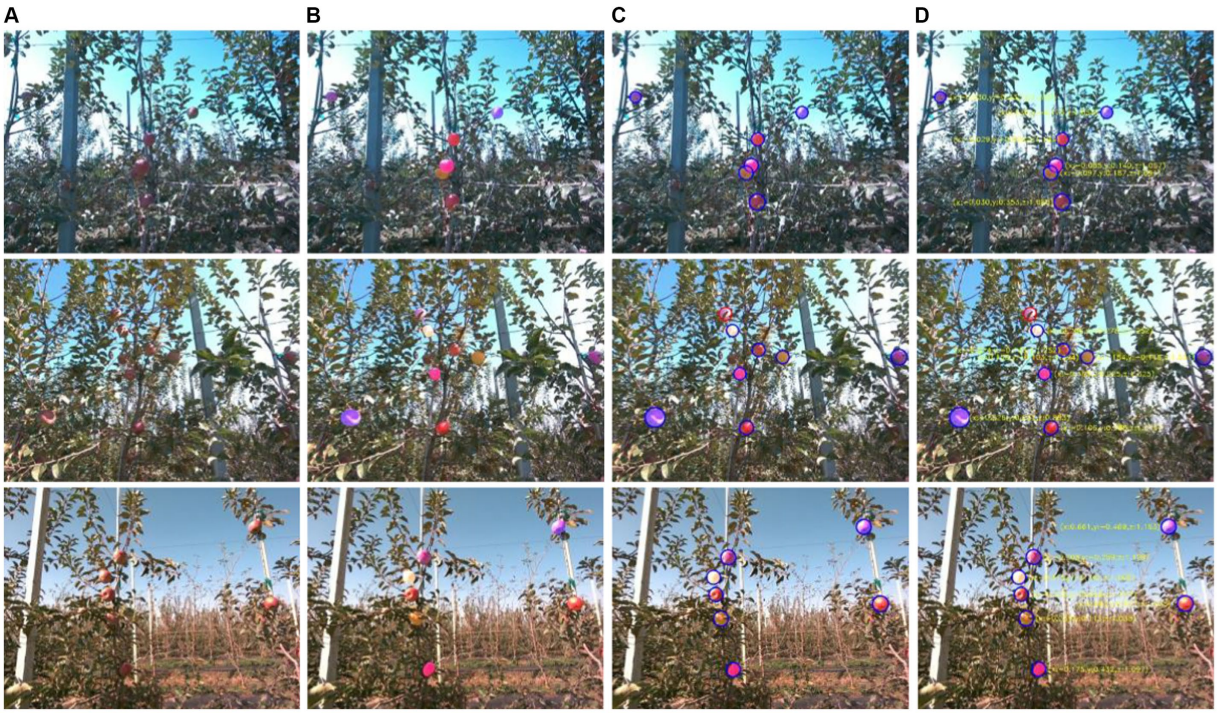

FIGURE 12
Apple picking point localization process. **(A)** Original image. **(B)** Segmentation result. **(C)** Pickable apples. **(D)** Localization result of pickable apples.

errors in the *X* and *Y* directions for localization are less than 1.5 mm, while the median error in the *Z* direction is close to 0. Moreover, the maximum error in the *Z* direction is approximately 1 mm, demonstrating high accuracy.

In the future, due to the high cost of obtaining instance segmentation data and issues related to the real-time performance of the models, we will focus on in-depth research in two critical areas: data generation and model lightweight. This will enable practical applications on edge devices and embedded systems.

Data availability statement

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

Author contributions

ST: Conceptualization, Funding acquisition, Supervision, Writing – original draft, Writing – review & editing. ZX: Data curation, Methodology, Writing – original draft, Writing – review & editing, Conceptualization, Formal analysis, Funding acquisition, Investigation, Project administration, Resources, Software, Supervision, Validation, Visualization. JG: Funding acquisition, Investigation, Writing – original draft, Writing – review & editing. WW: Validation, Investigation, Writing – original draft, Writing – review & editing. ZH: Writing – review & editing, Writing – original draft, Visualization, Validation, Formal analysis. WZ: Writing – review & editing, Validation, Visualization, Software.

Funding

The author(s) declare that financial support was received for the research, authorship, and/or publication of this article. This research was funded by the Key Project of Jiangsu Province Key Research and Development Program (No. BE2021016-3), the Jiangsu Agricultural Science and Technology Independent Innovation Fund Project (No. CX (22) 3016), and the Key R&D Program (Agricultural Research and Development) Project in Yancheng City (No. YCBN202309).

Acknowledgments

The authors express their gratitude to the editors and reviewers for their invaluable comments and suggestions.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

References

Ahmad, M. T., Greenspan, M., Asif, M., and Marshall, J. A.. (2018). Robust apple segmentation using fuzzy logic. 5th International Multi-Topic ICT Conference: Technologies For Future Generations, IMTIC 2018—Proceedings. 1–5.

Bolya, D., Zhou, C., Xiao, F., and Lee, Y. J.. (2019). YOLACT: real-time instance segmentation. Proceedings of the IEEE International Conference on Computer Vision. 9157–9166

Chen, L. C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. (2017). DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Trans. Pattern Anal. Mach. Intell.* 40, 834–848. doi: 10.1109/TPAMI.2017.2699184

Chen, W., Zhang, J., Guo, B., Wei, Q., and Zhu, Z. (2021). An apple detection method based on des-YOLO v4 algorithm for harvesting robots in complex environment. *Math. Probl. Eng.* 2021, 1–12. doi: 10.1155/2021/7351470

Gai, R., Chen, N., and Yuan, H. (2023). A detection algorithm for cherry fruits based on the improved YOLO-v4 model. *Neural Comput. Appl.* 35, 13895–13906. doi: 10.1007/s00521-021-06029-z

Gené-Mola, J., Ferrer-Ferrer, M., Gregorio, E., Blok, P. M., Hemming, J., Morros, J. R., et al. (2023). Looking behind occlusions: a study on amodal segmentation for robust on-tree apple fruit size estimation. *Comput. Electron. Agric.* 209:107854. doi: 10.1016/j.compag.2023.107854

Gené-Mola, J., Gregorio, E., Guevara, J., Auat, F., Sanz-Cortiella, R., Escolà, A., et al. (2019). Fruit detection in an apple orchard using a mobile terrestrial laser scanner. *Biosyst. Eng.* 187, 171–184. doi: 10.1016/j.biosystemseng.2019.08.017

He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2020). Mask R-CNN. *IEEE Trans. Pattern Anal. Mach. Intell.* 42, 386–397. doi: 10.1109/TPAMI.2018.2844175

Hu, T., Wang, W., Gu, J., Xia, Z., Zhang, J., and Wang, B. (2023). Research on apple object detection and localization method based on improved YOLOX and RGB-D images. *Agronomy* 13:1816. doi: 10.3390/agronomy13071816

Huang, Z., Huang, L., Gong, Y., Huang, C., and Wang, X.. (2019). Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 6409–6418.

Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q.. (2017). Densely connected convolutional networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 4700–4708

Jia, W., Tian, Y., Luo, R., Zhang, Z., Lian, J., and Zheng, Y. (2020). Detection and segmentation of overlapped fruits based on optimized mask R-CNN application in apple harvesting robot. *Comput. Electron. Agric.* 172:105380. doi: 10.1016/j.compag.2020.105380

Jia, W., Wang, Z., Zhang, Z., Yang, X., Hou, S., and Zheng, Y. (2022a). A fast and efficient green apple object detection model based on Foveabox. *J. King Saud Univ. - Comput. Inf. Sci.* 34, 5156–5169. doi: 10.1016/j.jksuci.2022.01.005

Jia, W., Zhang, Z., Shao, W., Hou, S., Ji, Z., Liu, G., et al. (2021). FoveaMask: a fast and accurate deep learning model for green fruit instance segmentation. *Comput. Electron. Agric.* 191:106488. doi: 10.1016/j.compag.2021.106488

Jia, W., Zhang, Z., Shao, W., Ji, Z., and Hou, S. (2022b). RS-Net: robust segmentation of green overlapped apples. *Precis. Agric.* 23, 492–513. doi: 10.1007/s11119-021-09846-3

Kang, H., and Chen, C. (2019). Fruit detection and segmentation for apple harvesting using visual sensor in orchards. *Sensors* 19:4599. doi: 10.3390/s19204599

Kang, H., and Chen, C. (2020). Fruit detection, segmentation and 3D visualisation of environments in apple orchards. *Comput. Electron. Agric.* 171:105302. doi: 10.1016/j.compag.2020.105302

Kang, H., Wang, X., and Chen, C. (2022). Accurate fruit localisation using high resolution LiDAR-camera fusion and instance segmentation. *Comput. Electron. Agric.* 203:107450. doi: 10.1016/j.compag.2022.107450

Kang, H., Zhou, H., Wang, X., and Chen, C. (2020). Real-time fruit recognition and grasping estimation for robotic apple harvesting. *Sensors* 20:5670. doi: 10.3390/s20195670

Kong, T., Sun, F., Liu, H., Jiang, Y., Li, L., and Shi, J. (2020). FoveaBox: beyond anchor-based object detection. *IEEE Trans. Image Process.* 29, 7389–7398. doi: 10.1109/TIP.2020.3002345

Li, Q., Jia, W., Sun, M., Hou, S., and Zheng, Y. (2021). A novel green apple segmentation algorithm based on ensemble U-Net under complex orchard environment. *Comput. Electron. Agric.* 180:105900. doi: 10.1016/j.compag.2020.105900

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

Ronneberger, O., Fischer, P., and Brox, T.. (2015). U-net: convolutional networks for biomedical image segmentation. Medical Image Computing and Computer-Assisted Intervention—MICCAI. 234–241.

Tan, M., and Le, Q. V.. (2019). EfficientNet: rethinking model scaling for convolutional neural networks. International Conference on Machine Learning. 6105–6114.

Tan, M., and Le, Q. V.. (2021). EfficientNetV2: smaller models and faster training. International Conference on Machine Learning. 10096–10106.

Tan, M., Pang, R., and Le, Q. V.. (2020). EfficientDet: scalable and efficient object detection. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 10781–10790.

Tian, Y., Duan, H., Luo, R., Zhang, Y., Jia, W., Lian, J., et al. (2019). Fast recognition and location of target fruit based on depth information. *IEEE Access* 7, 170553–170563. doi: 10.1109/ACCESS.2019.2955566

Wang, D., and He, D. (2022a). Apple detection and instance segmentation in natural environments using an improved mask scoring R-CNN model. *Front. Plant Sci.* 13:1016470. doi: 10.3389/fpls.2022.1016470

Wang, D., and He, D. (2022b). Fusion of mask RCNN and attention mechanism for instance segmentation of apples under complex background. *Comput. Electron. Agric.* 196:106864. doi: 10.1016/j.compag.2022.106864

Wang, X., Kang, H., Zhou, H., Au, W., Wang, M. Y., and Chen, C. (2023). Development and evaluation of a robust soft robotic gripper for apple harvesting. *Comput. Electron. Agric.* 204:107552. doi: 10.1016/j.compag.2022.107552

Wang, X., Kong, T., Shen, C., Jiang, Y., and Li, L.. (2020a). SOLO: segmenting objects by locations. Computer Vision—ECCV 2020. 649–665.

Wang, W., Zhang, Y., Gu, J., and Wang, J. (2022). A proactive manufacturing resources assignment method based on production performance prediction for the smart factory. *IEEE Trans. Ind. Inform.* 18, 46–55. doi: 10.1109/TII.2021.3073404

Wang, X., Zhang, R., Kong, T., Li, L., and Shen, C.. (2020b). SOLOv2: dynamic and fast instance segmentation. Advances in Neural Information Processing Systems. 17721–17732.

Wu, L., Ma, J., Zhao, Y., and Liu, H. (2021). Apple detection in complex scene using the improved yolov4 model. *Agronomy* 11:476. doi: 10.3390/agronomy11030476

Xia, Z., Gu, J., Wang, W., and Huang, Z. (2023). Research on a lightweight electronic component detection method based on knowledge distillation. *Math. Biosci. Eng.* 20, 20971–20994. doi: 10.3934/mbe.2023928

Xia, Z., Gu, J., Zhang, K., Wang, W., and Li, J. (2022). Research on multi-scene electronic component detection algorithm with anchor assignment based on K-means. *Electronics* 11:514. doi: 10.3390/electronics11040514

Zhang, S., Chi, C., Yao, Y., Lei, Z., and Li, S. Z.. (2020). Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 9759–9768.

Zou, K., Ge, L., Zhou, H., Zhang, C., and Li, W. (2022). An apple image segmentation method based on a color index obtained by a genetic algorithm. *Multimed. Tools Appl.* 81, 8139–8153. doi: 10.1007/s11042-022-11905-4