Method for the estimation of the cutting points in tomato seedling grafting based on improved YOLO11n

Rongtao Li[1], Fahui Yuan[1], Sajad Ali[1], Xiang Yin[2], Yong He[1,3*], Yufei Liu[1,3*]

(1. *College of Biosystems Engineering and Food Science, Zhejiang University, Hangzhou 310058, China*;
2. *School of Agricultural Engineering and Food Science, Shandong University of Technology, Zibo 255000, Shandong, China*;
3. *Key Laboratory of Spectroscopy Sensing, Ministry of Agriculture and Rural Affairs, Hangzhou 310058, China*)

Abstract: A grafting robot needs to obtain the position information for the plant seedlings to perform automatic grafting operations. Accurately measuring the cutting points required during grafting plays a pivotal role in completing high-quality grafting tasks. Traditional visual detection models exhibit suboptimal performance on edge devices due to their large model size and suffer from limited detection efficiency. To achieve rapid and precise cutting point localization, this study proposes an all-new module termed the Stimulative Upsample Block (SUB). Additionally, the Spatial and Channel Reconstruction Convolution (SCConv) and a Local Importance-based Attention (LIA) mechanism are incorporated into the YOLO11n architecture, culminating in an enhanced model named YOLO11n-LSS. Our model achieved mean average precision (mAP) values of 93.2% for the instance segmentation task and 98.9% for the key point detection task. Compared to YOLOv8n and YOLO11n, our model reduces the number of parameters and computational cost by 4.6% and 3.8%, respectively, making it a high-performance and lightweight solution. The successful application of the new algorithm will significantly improve the production efficiency of automated tomato grafting and contribute to the advancement of the tomato cultivation industry.

Keywords: seedling grafting, stem detection, instance segmentation, key point detection, YOLO algorithm
DOI: 10.25165/j.ijabe.20261901.10095

Citation: Li R T, Yuan F H, Ali S, Yin X, He Y, Liu Y F. Method for the estimation of the cutting points in tomato seedling grafting based on improved YOLO11n. Int J Agric & Biol Eng, 2026; 19(1): 179–186.

1 Introduction

Tomato (*Solanum lycopersicum*), a member of the genus Solanum in the family Solanaceae, is one of the most important vegetables in modern human society[1]. Tomatoes are rich in vitamins, minerals, and various bioactive compounds that provide essential nutrients and offer significant health benefits[2]. Due to their edible and commercial value, tomatoes have the highest annual output among vegetable crops. However, with the annual increase in tomato planting areas, severe problems such as continuous cropping obstacles and soil-borne diseases have emerged. The incidence of tomato pests and diseases has worsened, posing significant challenges to both tomato yield and quality[3-5].

Grafting technology involves attaching a branch, bud, or other tissue (referred to as the scion) from one plant onto another plant with a root system (referred to as the rootstock). This allows the scion to receive nutrients from the rootstock and develop into an independent plant[6]. Figure 1 shows the process of plant grafting. Grafting can enhance plant vitality and fruit quality[7], increase tolerance to high or low temperature[8], enhance resistance to salt and heavy metal stress[9], and improve nutrient utilization efficiency[10]. Currently, grafted seedling production in China primarily relies on traditional manual methods, which are associated with issues such as low efficiency and unstable seedling quality, hindering the healthy development of the grafted planting industry[11]. Since the 1980s, significant progress has been made in the research into vegetable grafting robots. The research on semi-automatic grafting robots has been relatively mature and is advancing towards full automation[12,13]. The design of key components in vegetable grafting robots mainly focuses on the clamping, cutting, and alignment devices, which have gradually developed towards greater precision and automation[14]. In the grafting process, accurate identification of the components is the foundation of all subsequent steps. It provides the necessary data for the following tasks by obtaining the specific location information of the seedlings and the cutting points. Therefore, integrating computer vision to assist with the grafting process is a crucial step towards automated grafting.

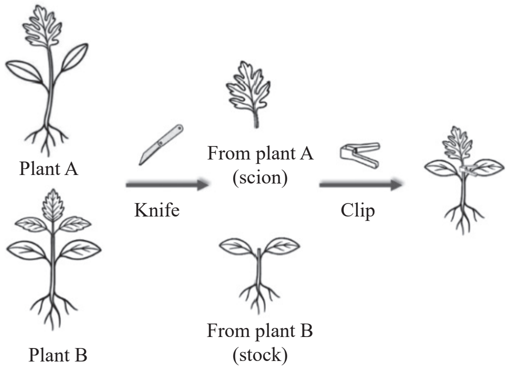


Figure 1 Diagram of the grafting process

Received date: 2025-08-13 **Accepted date:** 2025-11-02

Biographies: Rongtao Li, MS candidate, research interest: agricultural digitalization, Email: rtli@zju.edu.cn; **Fahui Yuan**, PhD candidate, research interest: agricultural digitalization, Email: fahuiyuan@zju.edu.cn; **Sajad Ali**, PhD candidate, research interest: agricultural digitalization, Email: sajadali@zju.edu.cn; **Xiang Yin**, PhD, Professor, research interest: agricultural automation and autonomy, Email: yinxiang@sdut.edu.cn.

***Corresponding author: Yong He**, PhD, Professor, research interest: intelligent agriculture. College of Biosystems Engineering and Food Science, Zhejiang University, Hangzhou 310058, China. Email: yhe@zju.edu.cn; **Yufei Liu**, PhD, Associate Professor, research interest: agricultural automation. College of Biosystems Engineering and Food Science, Zhejiang University, Hangzhou 310058, China. Email: yufeiliu@zju.edu.cn.

The detection of plant stems is influenced by various factors, such as interference between seedlings in the tray, the potential loss caused by the limited number of stem pixels, and the similarity in visual appearance between the foreground and background. Deep learning, a rapidly developing subfield of machine learning, has become one of the most popular research directions in computer vision in recent years. Deep learning algorithms are capable of handling complex visual problems, offering stronger robustness against environmental variations. At the same time, they exhibit excellent generalization capability for unknown data, maintaining high recognition accuracy[15,16]. Based on the Inception-V2 network, Yuan et al.[17] improved the Single Shot MultiBox Detector (SSD) algorithm and combined it with deep learning technology to identify tomatoes in greenhouses efficiently and accurately. Ning et al.[18] combined the region growing algorithm with the Mask R-CNN model to achieve precise identification of grape stems and successfully located the picking points of grapes. Ko et al.[19] employed a multi-stream convolutional neural network (ConvNet) and enhanced the multi-view tomato ripeness detection algorithm through the stochastic decision fusion (SDF) method.

You Only Look Once (YOLO) series algorithms are single-stage detectors, which can process high-resolution images in real time and achieve fast inference speeds while maintaining accuracy[20]. It is a lightweight network model with extremely fast object detection speed. The YOLO series is commonly chosen for recognition tasks in agricultural production. Li et al.[21] proposed a tomato recognition and localization algorithm, which employed Quality Focal Loss (QFL) to improve the loss function, effectively enhancing the accuracy of detection and positioning. Zhang et al.[22] introduced the Deformable Large Kernel Attention (DLKA) module and dynamic snake convolution into YOLOv8 to accurately recognize the stems of cherry tomatoes and their cutting points. Yu et al.[23] improved the backbone network and optimized the feature extraction module of YOLOv8s-seg, enhancing the recognition accuracy and detection speed of watermelon rootstock seedlings, while the model's complexity and robustness in complex agricultural environments still require further improvement. Meanwhile, the existing automated grafting machines cannot differentiate and precisely position each plant when performing cutting operations on an entire row of grafted seedlings, which may affect the success rate of the grafting process[24]. Grafted seedlings are in the early stages of growth and occupy a small proportion of the image's pixels, which presents a significant challenge for detection models. Additionally, the limited computational power of edge computing terminals, driven by market demand for grafting machines, imposes constraints on model complexity. Consequently, there is a critical need to develop a tomato grafting point recognition algorithm that effectively balances detection accuracy with computational efficiency.

To address the above issues, our goal is to develop a network model, YOLO11n-LSS, designed for the accurate detection of cutting points on tomato seedlings. Based on the YOLO11n architecture, the proposed method integrates the LIA mechanism with the aim of emphasizing stem features. Additionally, a novel upsample module, SUB, is introduced to enhance feature extraction, thereby intending to improve the precision of stem recognition and cutting point detection. To streamline the model for practical production needs, SCConv is incorporated to reduce the overall number of parameters and lightweight the model. To better align with image recognition outcomes, this study proposes a cutting point positioning method based on the phenotypic characteristics of tomato seedlings, making the approach potentially applicable to various crop types.

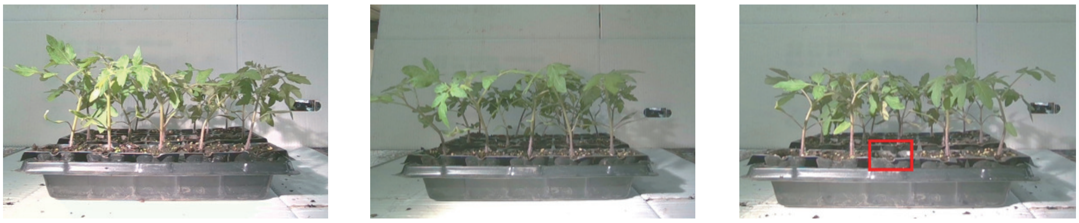
The goals of this study are: 1) to more accurately identify tomato stems, a new instance segmentation model based on the YOLO11n architecture is proposed, and 2) to rapidly and precisely locate the optimal cutting position in tomato seedling grafting, a method is developed that combines the outputs of key point detection and instance segmentation.

2 Materials and methods

2.1 Dataset

2.1.1 Data acquisition

To obtain more diverse data, the tomato seedling image dataset used in this study was obtained from the Wuwangnong Group Research Center in Hangzhou, Zhejiang Province, China. The images were captured using a stereo camera (RealSense D455, Intel, USA), with a total of 450 images at a resolution of 1280×720 pixels. To closely align with practical production requirements, the images were collected under various conditions: normal, lighting variations, and missing seedlings, as shown in Figure 2.



a. Normal condition b. Lighting variations (poor lighting) c. Missing seedling

Figure 2 Main environments of tomato stems

2.1.2 Image distortion correction

The positioning near the edge of the image may have deviations because of the camera's distortion. It is necessary to perform distortion correction on the collected images. Twenty checkerboard images of size 1280×720 pixels were captured using the D455 camera, with the checkerboard in different positions in each image. The "Camera Calibration" toolbox in MATLAB (version R2023b) was used for camera calibration to obtain the internal parameter matrix (mtx) and distortion matrix (dist) of the camera. The comprehensive average error of all labeled images was 0.24 pixels, which meets the detection requirements for key points of tomato plants in terms of accuracy. Using the obtained mtx and dist, the distorted image was corrected to an undistorted image after extracting the Region of Interest (ROI).

$$\text{mtx} = \begin{pmatrix} 707.1304 & 0 & 681.2313 \\ 0 & 708.9954 & 398.8167 \\ 0 & 0 & 1 \end{pmatrix} \tag{1}$$

$$\mathrm{dist} = [-0.1148 \quad 0.1410 \quad -0.0297 \quad 0 \quad 0] \tag{2}$$

2.1.3 Data augmentation and annotation

To increase the sample diversity under different environmental influences, enhance the robustness and generalization ability of the model trained on the dataset samples, and reduce the risk of overfitting of the model, this study expanded the dataset through offline augmentation methods, which included changing the image brightness, adding noise, horizontal flipping, and random cropping. These methods were randomly combined to conduct data augmentation. 1050 images were acquired. After inspection, the images that clearly did not meet the requirements were discarded by us. Eventually, 974 images were obtained, which were randomly divided into the training, validation, and test sets in a ratio of 7:2:1. Therefore, there were 682 images in the training set, 195 images in the validation set, and 97 images in the test set. Each image was annotated using Labelme toolbox (version 6.1.1). The contours of the tomato seedling stems were marked with the polygon tool, the regions containing the key points were selected with the rectangle tool, and the key points were marked with the point tool.

2.2 YOLO11n-LSS algorithm modeling

This study focuses on identifying tomato seedling stems. Considering that these stems are prone to misdetection and are difficult to identify when obstructed by branches and leaves, the YOLO11n-LSS model was proposed in this study. To balance the computational cost and detection performance, the YOLO11n was chosen as the basic framework. This choice was made to achieve an optimal combination of efficient detection and lightweight deployment. The YOLO11n model was the latest version of the Ultralytics YOLO series. It adopted an improved backbone and neck architecture, enhancing the feature extraction capabilities and enabling more precise target detection and performance for complex tasks[25]. However, the tomato seedling stems are slender and easily confused with the background, resulting in a detection accuracy that was insufficient for our requirements. To address this issue, the proposed SUB module can better extract and enhance features, fuse different levels, and allow the model to focus more effectively on target features. The LIA module was incorporated into the backbone feature extraction network, which expands the receptive field of the model and enhances its perception capability for small targets. The SCConv module was introduced, which effectively reduces redundant information by adaptively adjusting the spatial structure and channel relationships of features, and lightens the entire model without compromising the detection accuracy. Figure 3 presents the YOLO11n-LSS algorithm model proposed in this study.

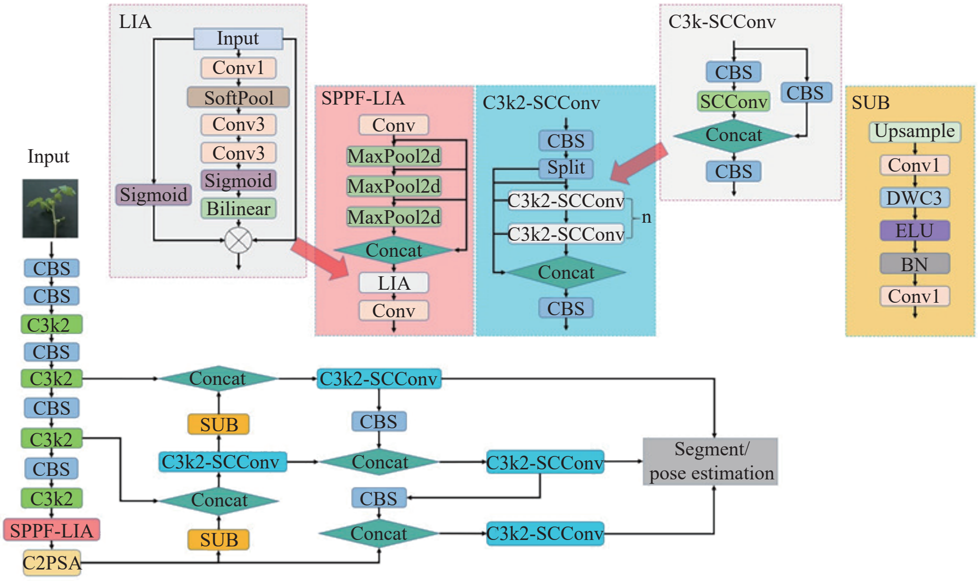


Figure 3 Network structure diagram of YOLO11n-LSS

2.2.1 LIA module

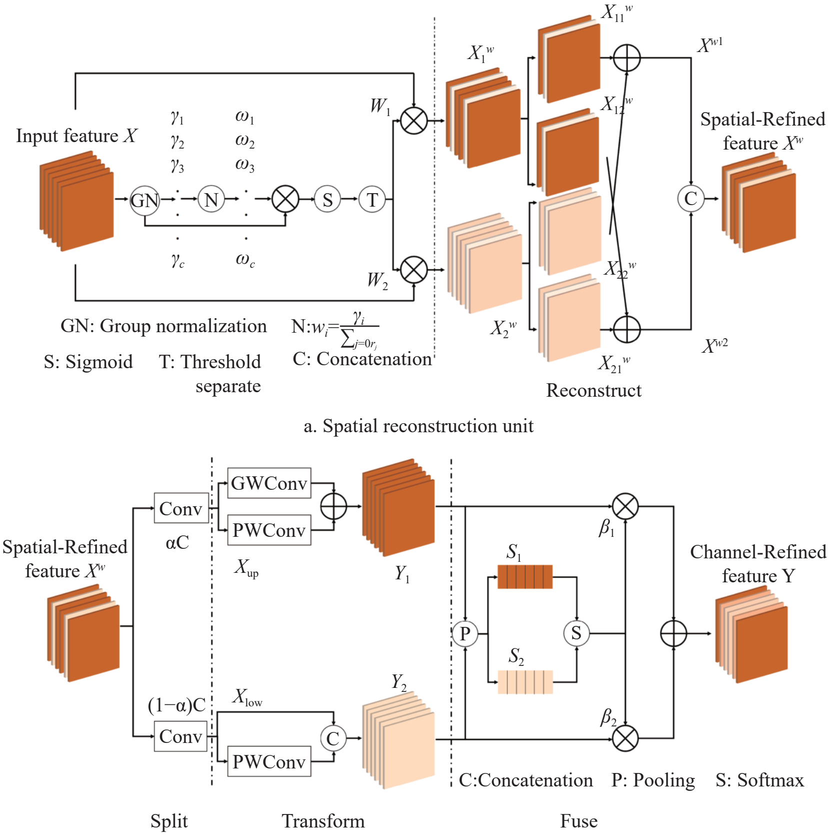
The LIA module employed the SoftPool module to downsample the image features, extracting and retaining the main feature information. It filtered out the relatively peripheral features, enhancing the computational efficiency[26]. As shown in Figure 3, the 3×3 convolution further refines the extracted local features, enhancing the model's perception of local information. Stride and squeeze convolutions compress the computational space, increasing the receptive field of the model. The Sigmoid function activates features and attention weights, enhancing the model's capability to understand more complex and subtle images and differentiate the significance of features at various positions. Bilinear interpolation resizes the feature maps to match the size of the input feature maps, ensuring the effectiveness of the attention mechanism. The gate channel selects the feature maps, preventing artifacts caused by stride convolution and bilinear interpolation. The LIA module focuses on the target features while disregarding the disruptive information, achieving the ability to capture features of small targets on the low-resolution feature maps. The LIA module was integrated with the SPPF module in the backbone section to form the SPPF-LIA module, which significantly enhances the model's detection of tiny targets such as the seedling stems.

2.2.2 SUB module

To better extract the features of tomato seedling stems and avoid the confusion caused by the distinction between foreground and background, a new module named SUB was proposed, an innovation based on the upsample module in YOLO11. As shown in Figure 3, the SUB module doubles the size of the input feature map and increases the resolution. The 1×1 convolution is applied to reduce the channel numbers of the input data. It can reduce the model's number of parameters while retaining the required feature maps. A 3×3 depth-wise convolution (DWC) further extracts feature information, enhancing the model's capability to perceive potential detection regions. The Exponential Linear Unit (ELU) activation function introduces non-linear factors to neurons, increasing the expressive power of the deep neural network in the model. Batch normalization (BN) ensures that the output features fall within the same scale range. It accelerates the convergence speed during training. The 1×1 convolution matches the input channel number of the next stage, enabling the SUB module to play its due role.

2.2.3 SCConv module

SCConv module consists of two units arranged in sequence, namely the Spatial Reconstruction Unit (SRU) and the Channel Reconstruction Unit (CRU)[27]. The input features are firstly processed by SRU to obtain the spatially refined features, and then passed through CRU to obtain the channel-synthesized features as the output of the entire module. It is introduced into the model to achieve higher performance at a lower computational cost while reducing model complexity. The network structure of SCConv is shown in Figure 4.



a. Spatial reconstruction unit

b. Channel reconstruction unit

Figure 4 Structure diagram of the SCConv module

In SRU, the separation operation is designed to extract features differently based on the varying level of information entropy. The reconstruction operation aims to enhance feature representation by fusing features with different information densities, while maintaining minimal spatial overhead. This is described by Equations (3) and (4) as follows:

$$X_{\text{out}} = \text{GN}(X) = \gamma\frac{x-\mu}{\sqrt{\sigma^2+\varepsilon}}+\beta \tag{3}$$

$$W = \text{Gate}\left(\text{Sigmoid}(W_r(GN(X)))\right) \tag{4}$$

where, μ and σ are the mean and standard deviation of feature X; γ and β are parameters in the training process; ε is a tiny constant; X_{out} represents the standardized feature X; W_γ is the correlation weights in normalization process; W is the feature values after feature mapping; X_{11}^w, X_{12}^w, X_{21}^w, and X_{22}^w are respectively obtained by equally dividing X_1^w and X_2^w; ⊗ represents the element multiplication; ⊕ represents the element addition; and U represents the operation of finding the union.

In CRU, the split operation divides the original feature space into two parts, αC and (1−α)C, where α is a hyperparameter and $0\leq\alpha\leq1$. The 1×1 convolution kernels are applied to compress channel numbers to obtain X_{up} and X_{low}, respectively. In the transformation operation, X_{up} is used as the part with rich information content processed by efficient convolution for calculation. X_{low} serves as a supplement. In the fusion stage, Y_1 and Y_2 are merged using the point convolution method described by Equations (5) and (6):

$$S_m = \text{Pooling}(Y_m) = \frac{1}{H\times W}\sum_{i=1}^{H}\sum_{j=1}^{W}Y_c(i,j),\ \ m=1,2 \tag{5}$$

$$Y=\beta_1Y_1+\beta_2Y_2 \tag{6}$$

where, $\beta_1=\frac{e^{s_1}}{e^{s_1}+e^{s_2}}$, $\beta_2=\frac{e^{s_2}}{e^{s_1}+e^{s_2}}$, $\beta_1+\beta_2=1$.

The SCConv module was incorporated into the C3k2 module of the neck part, replacing the standard convolution module with the SCConv module to create a C2k2-SCConv module. This reduction of redundant information significantly decreased the model's number of parameters, achieving a lightweight model design while enhancing its detection performance.

2.3 Method for positioning grafting points

When making tomato scion cuttings, it is usually necessary to horizontally cut off the roots of the scion seedlings at a distance of about 5 mm above the cotyledons[28]. Due to the inconsistency between the pixel and world coordinate systems, the process was divided into several steps to simplify the operation. Figure 5 shows how to position the cutting point for grafting on the tomato stem. The growth direction of the tomato seedling stem is obtained via instance segmentation and is indicated in blue. The leaf axillae are then positioned at the intersection points of the cotyledons and the stem with red dots, based on key points detection. We find the final target, the cutting point required for grafting, by extending the key points 5 mm upwards along the direction of the tomato seedling stem, marked with yellow dots. A 1 cm×2 cm reference block is used to align the world coordinate system with the pixel coordinate system. By relating the block's known physical dimensions to its measured size in pixels, a scaling factor is derived. Any pixel length in the image can be converted into a physical length. In the actual operation of the grafting machine, the designed camera and working device will fix the position of the camera and the grafting seedling. This means that the scaling coefficient remains constant and only needs to be calibrated once before it can be used.

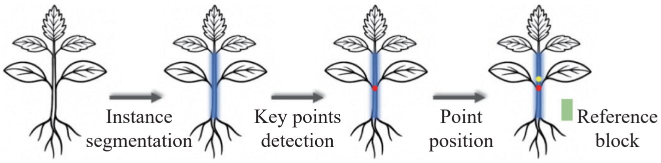


Figure 5 Schematic diagram of the grafting cutting point positioning

2.4 Evaluation indices

This study uses precision (*P*), recall (*R*), mAP, and F1 score (F1) to evaluate the performance of models. The specific equations are as follows:

$$P = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}} \tag{7}$$

$$R = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \tag{8}$$

$$\mathrm{AP} = \int_0^1 p(r)\,\mathrm{d}r \tag{9}$$

$$\mathrm{mAP} = \frac{1}{n}\sum_{k=1}^{k=n}\mathrm{AP}_k \tag{10}$$

$$\mathrm{F1} = \frac{2\times\mathrm{Precision}\times\mathrm{Recall}}{\mathrm{Precision}+\mathrm{Recall}} \tag{11}$$

where, TP stands for the actual positive samples that are correctly predicted as positive, FP stands for the actual negative samples that are incorrectly predicted as positive, FN stands for the actual positive samples that are incorrectly predicted as negative, and AP calculates the area under the *P*-*R* curve, which is a value encompassing the model's *P* and *R* performance. mAP@0.5 indicates that when the intersection over union (IoU) threshold is set to 0.5, the average AP value of all images of each category is calculated.

2.5 Experimental environment and parameter settings

The processor used was an Intel(R) Core(TM) i7-6850K CPU @ 3.60 GHz. The graphics card model was the NVIDIA GeForce RTX 4090. All experiments were conducted on a PC with Windows 10 as the operating system. The deep learning framework used was Pytorch 2.1.0+cu121, and Compute Unified Device Architecture (CUDA) 12.1, which is accelerated by cuDNN 8.9.0. The language used is Python 3.8.20. In the experiment, the settings of the hyperparameters are listed in Table 1. The YOLO algorithm will automatically resize the output original image to 640×640 pixels.

Table 1 Hyperparameter settings

Hyperparameter	Configuration
Epoch	300
Batch size	32
Initial learning rate	0.01
Momentum	0.937
Weight decay	0.0005
Image size	640×640 pixels

3 Results and analysis

3.1 Experiments on YOLO11n-LSS

The computational cost and model weight are crucial factors in determining whether a model can be deployed in practice. The YOLO algorithm stands out for its ability to strike a balance between model performance and computational efficiency[29]. YOLOv8 is the most mature and widely used representative of the YOLO series of algorithms, whereas YOLO11 is one of the latest algorithms, exemplifying recent advances in algorithm development[30]. A comparative experiment was conducted with YOLOv8 and YOLO11 to discuss the relative advantages of the new algorithm.

3.1.1 Instance segmentation

As listed in Table 2, the *P*, *R*, mAP@0.5, and F1 score of our model are 0.915, 0.884, 0.932, and 0.899, respectively. Compared with the YOLOv8n model, the accuracies are improved by 3.1%, 3.3%, 4.0%, and 3.2%, respectively, and the model size is reduced by 16.9%. Compared with the YOLO11n model, the accuracies are improved by 2.0%, 1.8%, 3.1%, and 1.9%, respectively, and the model size is reduced by 4.6%. The model of this study shows a notable improvement in tomato stem instance segmentation and recognition with fewer parameters. The YOLOv8n model has the shortest processing time, but it also has the lowest accuracy. The processing time of YOLO11n-LSS is 2.9% longer than that of YOLO11n. This indicates that YOLO11n-LSS can maintain real-time performance while improving detection accuracy. When comparing the detection performance of different models, the *P*-*R* curve graph is a highly effective tool. In the *P*-*R* graph, the abscissa represents recall, and the ordinate represents precision. It visually illustrates the changes in precision and recall rate under different thresholds, providing important quantitative indicators for the target detection task. An optimal model should have both high precision and high recall rates in general. That means the *P*-*R* curve should be as close as possible to the top right corner. As shown in Figure 6, the curve of our model is closer to the upper right corner and nearly encloses the curves of other models, which means our model has the highest recall rate, while the recall rate of the YOLOv8n model is the lowest at the same level of accuracy. This confirms that our model has superior performance. Figure 7 shows the actual instance segmentation results when detecting the tomato seedling stems. The blue part represents the section of the tomato stem from the base to the first true leaf. By precisely segmenting this part, we can obtain the growth trend of the tomato seedling. It can be seen that our model can effectively segment stems 1 to 5, which helps in locating the subsequent cutting points.

3.1.2 Key points detection

As listed in Table 3, the *P*, *R*, mAP@0.5, and F1 of the YOLO11n-LSS model are 0.962, 0.956, 0.989, and 0.959, respectively. Compared with the YOLOv8n model, the accuracies are improved by 3.0%, 1.5%, 1.0%, and 2.3%, respectively, and the model size is reduced by 17.9%. Compared with the YOLO11n model, the accuracies are improved by 0.2%, 3.1%, 0.6%, and 1.7%, respectively, and the model size decreases 4.9%. It is a pleasant finding to discover that our model not only achieves better detection performance but is also more compact. The processing time of YOLO11n-LSS is 2.1% longer than that of YOLO11n, which indicates that our model can ensure real-time performance. This gives our model a significant advantage when it comes to actual production deployment. Figure 8 shows the *P*-*R* graphs for the three algorithms' segmentation results. We find that the curve of YOLO11n-LSS almost completely encloses the curves of the other models. After adding the attention mechanism, our model notices more detailed information. At the same recall rate, our model can achieve better detection accuracy. It can be said that our model is expected to perform better in actual production tasks. Figure 9 presents the actual detection results of key points at the axillary regions of tomato stems and leaves. The blue box represents the part of the cotyledon containing the key points, while the red dots indicate the key points we are interested in, namely the axillae.

Table 2 Instance segmentation performance comparison of different YOLO algorithms

Model	Precision	Recall	mAP@0.5	F1	Parameters/M	Processing time per photo/ms
YOLOv8n	0.884	0.851	0.892	0.867	3.26	20.7
YOLO11n	0.895	0.866	0.901	0.880	2.84	24.2
YOLO11n-LSS	0.915	0.884	0.932	0.899	2.71	24.9

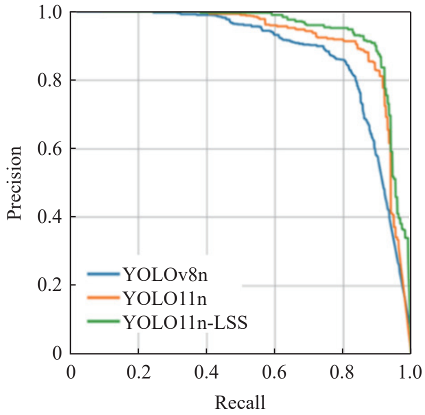


Figure 6 *P*-*R* curve graph for instance segmentation

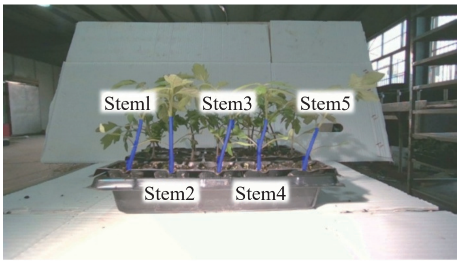


Figure 7 Actual effect of instance segmentation

Table 3 Key points detection performance comparison of different YOLO algorithms

Model	Precision	Recall	mAP@0.5	F1	Parameters/M	Processing time per photo/ms
YOLOv8n	0.932	0.941	0.979	0.936	3.08	17.6
YOLO11n	0.960	0.925	0.983	0.942	2.66	19.1
YOLO11n-LSS	0.962	0.956	0.989	0.959	2.53	19.5

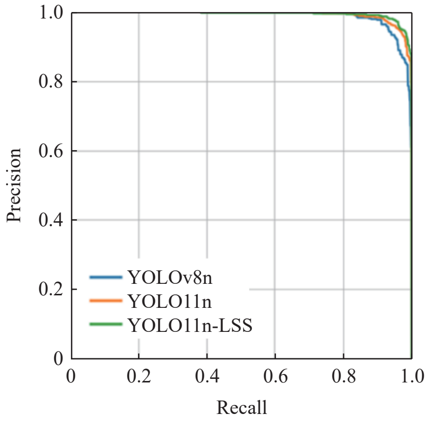


Figure 8 *P*-*R* curve graph for key point detection

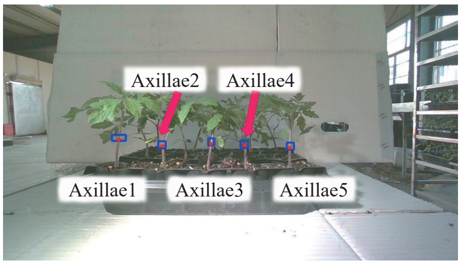


Figure 9 Actual effects of key point detection

3.2 Ablation studies

Based on the ablation study results in Table 4, we assess the impact of various components on model performance metrics. After adding the LIA module, our model pays more attention to the detailed parts of the tomato seedling stems, thereby improving segmentation accuracy, resulting in improvements of 0.8% in F1-score and 1.8% in mAP@0.5. The SUB module enhances the fusion of feature maps from different stages and helps the model better focus on the target features. After adding the SUB module, our model demonstrates stronger localization for targets such as tomato stems. However, the total number of parameters and the entire model computational cost have both increased after incorporating the LIA and the SUB modules. We replaced the ordinary convolution with the Spatial and Channel Reconstruction Convolution, which reduced the number of parameters and computational cost by 8.8% and 5.7%, respectively. Although there is a 0.3% decrease in *R*, accuracy is a more important indicator in the grafting operation. Therefore, it can be considered that each module contributed effectively, enabling our improved model to surpass the original YOLO model and achieve better detection performance.

Table 4 Ablation study on LIA, SUB, SCConv in YOLO11n-LSS

Model	Precision	Recall	mAP@0.5	F1	Parameters/M	FLOPs/G
YOLO11n	0.895	0.866	0.901	0.880	2.84	10.4
YOLO11n-L	0.902	0.874	0.919	0.888	2.93	10.4
YOLO11n-LS	0.910	0.887	0.927	0.898	2.97	10.6
YOLO11n-LSS	0.915	0.884	0.932	0.899	2.71	10.0

3.3 Positioning of the cutting points

The performance of our model in key point detection and instance segmentation was verified. Next, the actual positioning tests on the initial target will be conducted, namely, the cutting position of tomato stems during grafting operations. Based on the grafting standard, the point located 14 pixels above the axillae, equal to 5 mm, is the final cutting point we are looking for. Figure 10 shows the actual effect of locating the cutting point. The red dots represent the ones we obtained through key point detection. The green lines represent the middle columns of the stems, which are obtained by solving the intermediate values of the segmented regions of the instance. The blue dots represent the final cutting points determined based on geometric relationships.

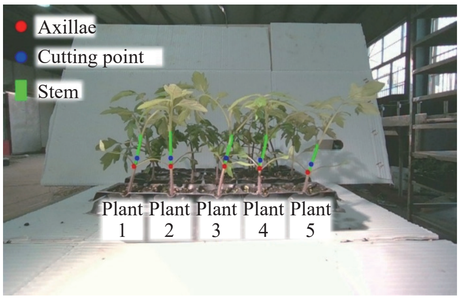


Figure 10 Actual effects of locating the cutting point

4 Conclusions

In this study, a deep learning positioning algorithm was proposed for the cutting points of tomato seedlings grafting, namely YOLO11n-LSS. We proposed a novel upsample module named SUB and introduced the LIA module and the SCConv module. It's worth noting that, compared with YOLOv8n and YOLO11n, our model increased mAP@0.5 by 1% and 0.6%, respectively, in key point detection and increased mAP@0.5 by 4% and 3.1%, respectively, in instance segmentation. Moreover, compared with YOLOv8n and YOLO11n, our model reduced the number of parameters by 16.9% and 4.6%, respectively. This indicates that our model has lower complexity while maintaining higher detection accuracy, which provides a unique advantage for its application in agricultural production. Through the ablation study, we verified that each module plays a corresponding role in the recognition task. With the collaboration of these modules, our model has better detection performance. These experimental results indicate that YOLO11n-LSS is a lightweight computer vision model with high performance, and has shown satisfactory detection performance in practical tests. We also verified that the point positioning method based on instance segmentation and key points detection is effective. Owing to its superior performance, our research is expected to be applied in practical tomato grafting operations, which will reduce the reliance on human labor and improve the overall efficiency of tomato cultivation.

Future work will involve the integration of three-dimensional point cloud data to support the innovative design of mechanical structures for grafting operations. The ultimate objective is to develop a fully automated tomato grafting system, thereby providing an effective solution for modern, large-scale tomato production.

Acknowledgements

This work was supported by the Zhejiang Province Agricultural Machinery Research, Manufacturing, and Application Integration Project (Grant No. 2023-YT-06) and the Fundamental Research Funds for the Zhejiang Provincial Universities (Grant No. 226-2024-00191). Thanks to Mr. L. Lv from Zhejiang University, China, and Mr. M. Zhao from Wuwangnong Farm, China, for their contribution to dataset production.

[References]

[1] Valderas-Martinez P, Chiva-Blanch G, Casas R, Arranz S, Martínez-Huélamo M, Urpi-Sarda M, et al. Tomato sauce enriched with olive oil exerts greater effects on cardiovascular disease risk factors than raw tomato and tomato sauce: A randomized trial. Nutrients, 2016; 8(3): 170.

[2] Song Y J, Teakle G, Lillywhite R. Unravelling effects of red/far-red light on nutritional quality and the role and mechanism in regulating lycopene synthesis in postharvest cherry tomatoes. Food Chemistry, 2023; 414: 135690.

[3] Geng W C, Ma Y, Zhang Y X, Zhu F. Research progress in soil health regulation technology for protected agriculture. Chinese Journal of Eco-Agriculture, 2022; 30(12): 1973–1984.

[4] Lu W H, Zhang N M, Bao L, Zhang L, Qin T F. Study advances on characteristics, causes and control measures of continuous cropping obstacles of facility cultivation in China. Soils, 2020; 52(4): 651–658.

[5] Weng P Y, Zheng H Y. Causes and mechanism on crop continuous monoculture problems and its control strategy. Subtropical Plant Science, 2020; 49(2): 157–162.

[6] Warschefsky E J, Klein L L, Frank M H, Chitwood D H, Londo J P, von Wettberg E J, et al. Rootstocks: Diversity, domestication, and impacts on shoot phenotypes. Trends in Plant Science, 2016; 21(5): 418–437.

[7] Riga P, Benedicto L, García-Flores L, Villaño D, Medina S, Gil-Izquierdo Á. Rootstock effect on serotonin and nutritional quality of tomatoes produced under low temperature and light conditions. Journal of Food Composition and Analysis, 2016; 46: 50–59.

[8] Ma Q, Niu C X, Wang C, Chen C H, Li Y, Wei M. Effects of differentially expressed microRNAs induced by rootstocks and silicon on improving chilling tolerance of cucumber seedlings (*Cucumis sativus* L.). BMC Genomics, 2023; 24(1): 250.

[9] Fu S J, Chen J Q, Wu X L, Gao H B, Lü G Y. Comprehensive evaluation of low temperature and salt tolerance in grafted and rootstock seedlings combined with yield and quality of grafted tomato. Horticulturae, 2022; 8(7): 595.

[10] Coşkun O F. The effect of grafting on morphological, physiological and molecular changes induced by drought stress in cucumber. Sustainability, 2023; 15(1): 875.

[11] Tian H M, Cai K, Tao Z, Yang J S, Zhang J, Wang C, et al. Design of automatic grafting machine for vegetables. Journal of Anhui Agricultural Sciences, 2024; 52(23): 197–200, 236.

[12] Zhang K L, Chu J, Zhang T Z, Yin Q, Kong Y S, Liu Z. Development status and analysis of automatic grafting technology for vegetables. Transactions of the Chinese Society for Agricultural Machinery, 2017; 48(3): 1–13.

[13] Belmonte-Ureña L J, Garrido-Cardenas J A, Camacho-Ferre F. Analysis of world research on grafting in horticultural plants. HortScience, 2020; 55(1): 112–120.

[14] Yan G P, Feng M S, Lin W G, Huang Y, Tong R Z, Cheng Y. Review and prospect for vegetable grafting robot and relevant key technologies. Agriculture, 2022; 12(10): 1578.

[15] Gulzar Y. Fruit image classification model based on MobileNetV2 with deep transfer learning technique. Sustainability, 2023; 15(3): 1906.

[16] Gao F F, Fang W T, Sun X M, Wu Z C, Zhao G N, Li G, et al. A novel apple fruit detection and counting methodology based on deep learning and trunk tracking in modern orchard. Computers and Electronics in Agriculture, 2022; 197: 107000.

[17] Yuan T, Lyu L, Zhang F, Fu J, Gao J, Zhang J X, et al. Robust cherry tomatoes detection algorithm in greenhouse scene based on SSD. Agriculture, 2020; 10(5): 160.

[18] Ning Z T, Luo L F, Liao J X, Wen H J, Wei H L, Lu Q H. Recognition and the optimal picking point location of grape stems based on deep learning. Transactions of the Chinese Society of Agricultural Engineering (Transactions of the CSAE), 2021; 37(9): 222–229.

[19] Ko K, Jang I, Choi J H, Lim J H, Lee D U. Stochastic decision fusion of convolutional neural networks for tomato ripeness detection in agricultural sorting systems. Sensors, 2021; 21(3): 917.

[20] Redmon J, Divvala S, Girshick R, Farhadi A. You Only Look Once: Unified, real-time object detection. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas: IEEE, 2016; pp.779–788. doi: 10.1109/CVPR.2016.91.

[21] Li T H, Sun M, He Q H, Zhang G S, Shi G Y, Ding X M, et al. Tomato recognition and location algorithm based on improved YOLOv5. Computers and Electronics in Agriculture, 2023; 208: 107759.

[22] Zhang G M, Cao H, Jin Y W, Zhong Y, Zhao A B, Zou X J, et al. YOLOv8n-DDA-SAM: Accurate cutting-point estimation for robotic cherry-tomato harvesting. Agriculture, 2024; 14(7): 1011.

[23] Yu Q C, Xu Z H, Zhu Y. Watermelon rootstock seedling detection based on improved YOLOv8 image segmentation. International Journal of Advanced Computer Science and Applications, 2025; 16(2): 160247.

[24] Yu Y T, Li Y J, Xie F X, Song J, Bai Y, Fan Y. Design and experiment of key components of a insertion vegetable grafting machine with six plants synchronous. Scientific Reports, 2025; 15(1): 16650.

[25] Khanam R, Hussain M. YOLOv11: An overview of the key architectural enhancements. 2024; arXiv: 2410.17725. doi: 10.48550/arXiv.2410.17725.

[26] Wang Y, Li Y S, Wang G, Liu X G. PlainUSR: Chasing faster ConvNet for efficient super-resolution. Computer Vision - ACCV2024, 2024; 15475: 246–264.

[27] Li J F, Wen Y, He L H. SCConv: Spatial and channel reconstruction convolution for feature redundancy. In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023; pp.6153–6162. doi: 10.1109/CVPR52729.2023.00596.

[28] Kleinhenz M D, Waiganjo M, Erbaugh J M, Miller S A. Preparing grafted tomato plants using the cleft graft method. Available: https://horticulture.ucdavis.edu/information/tomato-grafting-guide. Accessed on [2025-03-21].

[29] Wu M N, Lin H R, Shi X R, Zhu S J, Zheng B. MTS-YOLO: A multi-task lightweight and efficient model for tomato fruit bunch maturity and stem detection. Horticulturae, 2024; 10(9): 1006.

[30] Sapkota R, Flores-Calero M, Qureshi R, Badgujar C, Nepal U, Poulose A, et al. YOLO advances to its genesis: a decadal and comprehensive review of the You Only Look Once (YOLO) series. Artificial Intelligence Review, 2025; 58(9): 274.