

Article

YOLO11m-SCFPose: An Improved Detection Framework for Keypoint Extraction in Cucumber Fruit Phenotyping

Huijiao Yu ^{1,2}, Xuehui Zhang ^{1,2}, Jun Yan ^{1,2} and Xianyong Meng ^{1,2,*}

¹ College of Information Science and Engineering, Shandong Agricultural University, Tai'an 271000, China; yhj2328@163.com (H.Y.); zxh01090312@163.com (X.Z.)

² The Key Laboratory of Smart Agriculture Technology in Huang-Huai-Hai Region, Ministry of Agriculture and Rural Affairs, Tai'an 271018, China

* Correspondence: xym@sdaa.edu.cn

Abstract

To address the issues of low efficiency and large errors in traditional manual cucumber fruit phenotyping methods, this paper proposes the application of keypoint detection technology for cucumber phenotyping and designs an improved lightweight model called YOLO11m-SCFPose. Based on YOLO11m-pose, the original backbone network is replaced with the lightweight StarNet-S1 backbone, reducing model complexity. Additionally, an improved C3K2_PartialConv neck module is used to enhance information interaction and fusion among multi-scale features while maintaining computational efficiency. The Focaler-IoU loss function is employed to improve keypoint localization accuracy. Results show that the improved model achieves an mAP₅₀₋₉₅ of 0.924, with a floating-point operation count (GFLOPs) of 32.1, and reduces the model size to 1.229×10^7 parameters. This model demonstrates better computational efficiency and lower resource consumption, providing an effective lightweight solution for crop phenotypic analysis.

Keywords: cucumber fruit phenotyping; keypoint detection; lightweight model; YOLO11m-SCFPose

Academic Editor: Luigi De Bellis

Received: 30 May 2025

Revised: 9 July 2025

Accepted: 18 July 2025

Published: 20 July 2025

Citation: Yu, H.; Zhang, X.; Yan, J.; Meng, X. YOLO11m-SCFPose: An Improved Detection Framework for Keypoint Extraction in Cucumber Fruit Phenotyping. *Horticulturae* **2025**, *11*, 858. <https://doi.org/10.3390/horticulturae11070858>

Copyright: © 2025 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In modern agriculture, phenotypic research plays a central role in understanding the mechanisms of crop growth, development, and interactions with the environment. It is of great significance for crop genetic improvement, precision breeding, and intelligent cultivation management. Accurate phenotypic data not only provide a solid scientific foundation for genetic improvement and precision breeding but also support the development and implementation of intelligent cultivation strategies, thereby enhancing agricultural productivity and resource-use efficiency [1,2]. As the global demand for high-throughput breeding continues to grow, the development of automated, non-contact, and high-precision phenotypic information extraction technologies has become a key technical pathway for advancing the transformation and upgrading of smart agriculture [3]. With the ongoing refinement of deep-learning theory and the continuous improvement of computational hardware, deep learning has achieved extensive and in-depth progress in image processing, feature extraction, and pattern recognition. Its outstanding performance in

high-dimensional data modeling, nonlinear feature representation, and end-to-end learning has made it a critical research focus in the field of artificial intelligence, providing a robust technical foundation for the automation of various complex tasks [4].

Among various economic crops, cucumber is an important vegetable crop whose fruit appearance contains rich phenotypic information, including fruit length, peduncle-to-fruit length ratio, transverse diameter variation, and arch height. These fruit phenotypic traits are not only closely related to key agronomic characteristics such as yield, quality, and disease resistance [5,6], but they also directly influence market value and consumer preferences. For instance, the uniformity and integrity of fruit shape are often regarded as important criteria for high-quality cucumbers, directly affecting retail price and brand perception. High-precision and automated phenotypic measurement technologies for cucumber fruits play a crucial role in breeding and cultivation management. By accurately modeling and quantitatively analyzing fruit morphological traits, these technologies provide stable and reliable phenotypic evidence for genetic improvement and variety selection, thereby enhancing breeding efficiency and the scientific basis of decision-making. Moreover, phenotype-based quality evaluation systems can support standardized grading and traceability management, effectively improving product market competitiveness and consumer trust. Overall, such technologies offer significant advantages in promoting the intelligent and refined transformation of the cucumber industry, demonstrating strong application potential and value for broader adoption.

Traditional phenotypic extraction methods mainly rely on manual operations, which suffer from low efficiency and large errors, making them inadequate for the demands of smart agriculture in terms of high-throughput and precise phenotypic analysis. In recent years, deep-learning theories and image processing algorithms have been widely applied in the field of phenotypic extraction, providing new technical tools for crop phenotyping research [7,8]. Existing studies have primarily focused on fruit target detection and the extraction of basic geometric parameters, such as fruit count, length, width, and area. These tasks mainly depend on image segmentation and object detection frameworks. In the field of object detection, Qiu et al. [9] proposed a lightweight object detection network, GSE-YOLO, which builds upon YOLOv8n by incorporating a feature reparameterization module. This approach significantly reduces model complexity while maintaining high accuracy, thereby improving the performance of dragon fruit maturity recognition. Xu et al. [10] combined YOLOv8 with OpenCV to extract ten quantitative and four qualitative phenotypic traits of non-heading Chinese cabbage leaves, demonstrating the feasibility of integrating deep learning with traditional image processing algorithms. In terms of image segmentation, existing research has achieved notable success in extracting regions of fruits and plant structures. For example, Ndikumana et al. [11] developed a multitask detection framework combining YOLOv4 and U-Net, which effectively balances strawberry fruit detection and maturity classification, showing strong robustness under varying lighting and occlusion conditions. Zhang et al. [12] used image segmentation techniques to precisely extract spot patterns on pear fruits, obtaining phenotypic traits such as spot count, area, and color. Wei et al. [13] employed a synthetic occlusion image augmentation strategy to achieve high-throughput instance segmentation of mushroom caps, enabling accurate size estimation without the need for explicit masks. Wang et al. [14] introduced an edge attention mechanism into the apple instance segmentation framework, which significantly enhanced the extraction of structural details. Gai et al. [15] designed the HPPEM model by integrating hybrid task cascades with ConvNeXt, enabling high-precision segmentation of blueberry clusters and automated extraction of phenotypic traits such as count, maturity, and compactness. Cao et al. [16] proposed the EGF-Former model for instance segmentation of sweet peppers in complex scenes. With the guidance of multi-

scale feature fusion and boundary-aware modules, the model achieved a mean Intersection over Union (mIoU) of 83.45%, significantly outperforming existing segmentation methods.

Although deep-learning technologies have achieved remarkable progress in crop phenotypic information extraction, most existing methods rely on coarse-grained structural representations such as bounding boxes or segmentation masks. These approaches primarily focus on the overall shape and edge regions of fruits, making it difficult to effectively capture spatial relationships and geometric features between different parts. For modeling tasks involving significant variations in fruit posture, complex morphology, or indistinct local features, such methods often struggle to ensure representation accuracy and structural consistency, thereby limiting their applicability in fine-grained phenotypic analysis. To enhance the model's ability to describe fruit appearance and structure, it is essential to introduce more geometrically expressive and structured modeling approaches. Such methods can enable precise localization of key components and effectively represent their spatial relationships, thereby improving the accuracy and robustness of phenotypic characterization.

In recent years, structural modeling has gradually become a key direction in visual perception and representation tasks. Some methods have introduced keypoint detection, pose estimation, and structure-aware mechanisms to enhance the model's ability to capture spatial structures and morphological features [17]. Compared to traditional bounding box and segmentation mask approaches, keypoint detection enables explicit modeling of essential structural parts of fruits or organisms, offering greater geometric consistency and generalization capability in tasks such as morphology description, geometric parameter measurement, and behavior pattern analysis. Keypoint detection methods have shown strong accuracy and adaptability in animal structure modeling. For instance, Peng et al. [18] designed a non-contact system combining monocular depth images with keypoint detection to measure major body parts of cattle. Su et al. [19] reformulated the traditional horse body measurement problem as a keypoint localization task and proposed a top-down automatic Mongolian horse body measurement method by integrating object detection and keypoint detection algorithms. The method achieved an average accuracy of 90.1%, with the RTMPose algorithm reaching 91.4%. Li et al. [20] employed an improved YOLOv8-based keypoint detection algorithm to monitor cattle rumination behavior non-invasively, achieving an average error of only 5.6% in chew count. Chen et al. [21] developed the YOLOv8l-Pose model for high-precision detection of carapace keypoints in the Chinese mitten crab, extending the application of this strategy to aquatic structure modeling. Owing to its promising performance in animal structural modeling, keypoint detection techniques are increasingly being applied to plant phenotyping tasks. He et al. [22] developed the DEKR-SPrior network, which incorporates target location priors to guide the learning of discriminative features, enabling the high-precision detection of densely arranged and occluded keypoints in soybean pods. Zheng et al. [23] proposed a deep-learning vision algorithm based on RGB images for fruit instance segmentation and picking point localization, significantly improving task efficiency and performance through an end-to-end network that performs both tasks in parallel. Du et al. [24] combined image keypoints and point cloud processing methods to achieve accurate 3D pose estimation of tomato fruits, supporting structural modeling and size measurement tasks. Gao et al. [25] introduced SCPE, a structure-aware maize plant architecture estimation method based on YOLOv7-SlimPose for keypoint detection, combined with geometric reconstruction to obtain traits such as plant height and leaf angles. Chen et al. [26] proposed a grapefruit and picking point co-detection method based on the improved YOLOv8-GP model, improving the localization accuracy and efficiency of harvesting robots. Zheng et al. [27] developed an enhanced YOLOv8n-CK model for fast and accurate keypoint detection of cabbage

heads, achieving 99.2% accuracy even under occlusion while significantly reducing model parameters and computational cost. Wang et al. [28] designed the improved YOLOv8n-Pose-LBD algorithm for efficient detection and localization of goji berries and picking points in complex environments. Wang et al. [29] proposed the YOLO-TMPPD model, which improves tomato ripeness classification and picking point localization in complex greenhouse environments by integrating Depthwise Convolution, CARAFE up-sampling, and CBAM attention mechanisms, achieving higher detection accuracy and reduced model size. Ma et al. [30] developed the STRAW-YOLO method to jointly detect strawberries and their keypoints, enhancing the association between fruit objects and structural components.

Keypoint detection, as an intermediate representation that combines lightweight design with structural constraints, has demonstrated significant advantages in structured modeling tasks for agricultural imagery. Compared to traditional coarse-grained approaches based on bounding boxes or segmentation masks, keypoints are capable of capturing internal spatial structures and geometric axes of the target, making them particularly suitable for modeling phenotypic traits with complex morphology and tightly coupled structures. This contributes to improving both the accuracy and interpretability of structural representations. Although keypoint detection methods have seen preliminary applications in crop phenotyping studies, their systematic exploration in modeling fruit crops remains limited. Issues such as annotation protocols and model architecture design still require further optimization. Taking cucumber as an example, its fruit is typically elongated and naturally curved, with the stem and blossom ends exhibiting high similarity in color and texture. This makes it difficult for traditional appearance-based methods to distinguish and localize these regions precisely. Furthermore, geometric traits such as fruit length and curvature rely heavily on accurate modeling of the overall structural axis, which cannot be effectively represented using bounding boxes or segmentation masks due to their lack of continuity and structural constraints. Several keypoint detection approaches have been proposed, and models in the YOLO series have shown promising performance in agricultural keypoint detection tasks, offering both lightweight and efficient solutions. However, most of these models are trained on general-purpose datasets and lack targeted optimization for the morphological features of fruit crops. As a result, they still face challenges related to model complexity and computational cost, limiting their applicability in field conditions where low power consumption, real-time performance, and ease of deployment are essential requirements.

To address the challenges of inaccurate localization of key cucumber fruit structures, insufficient precision in geometric feature extraction, and high deployment costs of existing methods, this study proposes a lightweight phenotypic modeling model based on keypoint detection, named YOLO11m-SCFPose. By strategically designing the keypoint layout, the model extracts critical phenotypic indicators such as fruit length, stem-to-fruit length ratio, transverse diameter variation, and arch height. While maintaining high detection accuracy, the model features a low parameter count and minimal computational overhead, meeting the practical demands of real-time performance and ease of deployment in agricultural field environments. This provides a feasible solution for efficient fruit phenotyping and holds significant application value and potential for broader adoption.

2. Materials and Methods

This chapter consists of six sections: Section 2.1 describes the dataset acquisition and preprocessing procedures; Section 2.2 explains the calculation methods of cucumber phenotypic parameters; Section 2.3 focuses on the improved approach based on the YOLO11m-Pose model; Section 2.4 presents the keypoint detection visualization results

of the original and improved models; Section 2.5 outlines the evaluation metrics used for model performance assessment.

2.1. Data Acquisition and Processing

The cucumber sample images used in this study were collected at the Cucumber Science and Technology Courtyard in Juye, Shandong, focusing on the North China-type spined cucumber. To ensure image quality and experimental consistency, all images were captured indoors under controlled conditions between 3:00 PM and 5:00 PM each day, using a fixed artificial lighting system to minimize the impact of external light variations. Image acquisition was conducted using a Nikon Z30 (Nikon Corporation, Bangkok, Thailand) camera equipped with a 16–50 mm standard lens, with the focal length set to 30 mm. The camera was mounted on a stable tripod, with the lens positioned 50 cm above the ground. To maintain consistency, the camera position and angle remained unchanged throughout the acquisition process. All cucumbers used for shooting were harvested within 1–2 days and were not subjected to any cleaning, trimming, or other human interventions before shooting, thus preserving their natural state. The samples were placed on a calibration background board with a white base and black grid lines, featuring 1 cm grid intervals to enhance contrast between the cucumber contour and background and to serve as a reference for subsequent image annotation and dimensional measurements. As cucumber keypoints are morphologically similar, the samples were placed horizontally on the calibration board during acquisition to standardize their orientation and improve annotation accuracy and model training performance. The collected cucumber sample images are shown in Figure 1. A total of 900 images were collected, covering various natural shape variations, and served as the data foundation for subsequent model training and evaluation, as shown in Figure 1. Twenty images were randomly selected from the dataset, and 10 calibration grids were measured in each image to determine their pixel lengths. The average real-world distance represented by a single pixel was calculated as 0.638 mm, and this value was used as the scale for all subsequent distance-related calculations and analyses.

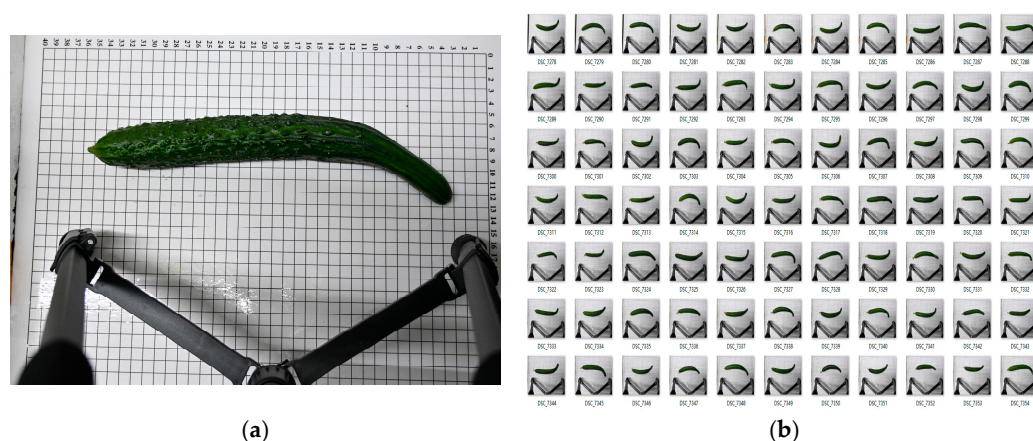


Figure 1. Cucumber fruit images: (a) Original image. (b) Image dataset.

This study employed the Roboflow annotation tool to manually label keypoints on cucumber images, with a total of 11 key landmarks annotated per cucumber. These include the peduncle point, the two endpoints, the central point, cross-sectional diameter points located 3 cm from both the apex and the base, and the central cross-sectional diameter points. An example of the annotation is shown in Figure 2. To enhance data diversity and complexity, the dataset was systematically augmented using multiple techniques, including increasing saturation to 1.8 times, enhancing brightness to 1.3 times, applying

Gaussian blur with a kernel size of 5×5 , and adding Gaussian noise with a mean of 0 and a standard deviation of 10, thereby improving the model's generalization ability and robustness. Figure 3 illustrates the visual effects of these augmentation operations on the original images. The dataset was divided into training, validation, and test sets in a ratio of 8:1:1. After augmentation, the final dataset consisted of 2270 images, with 1816 used for training and 227 each for validation and testing. Images were randomly selected from the original dataset to form the test set, without any augmentation, to ensure an objective evaluation of the model's generalization ability.

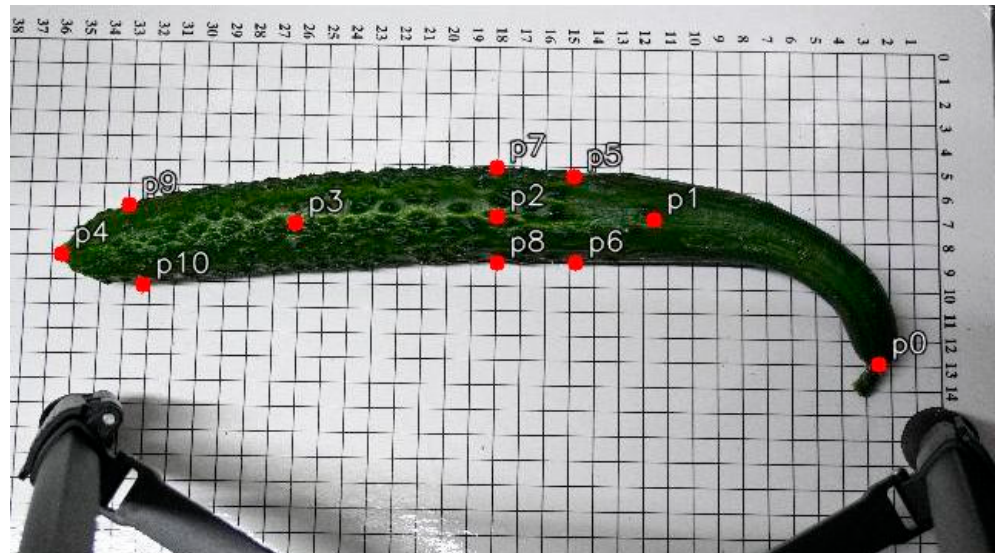
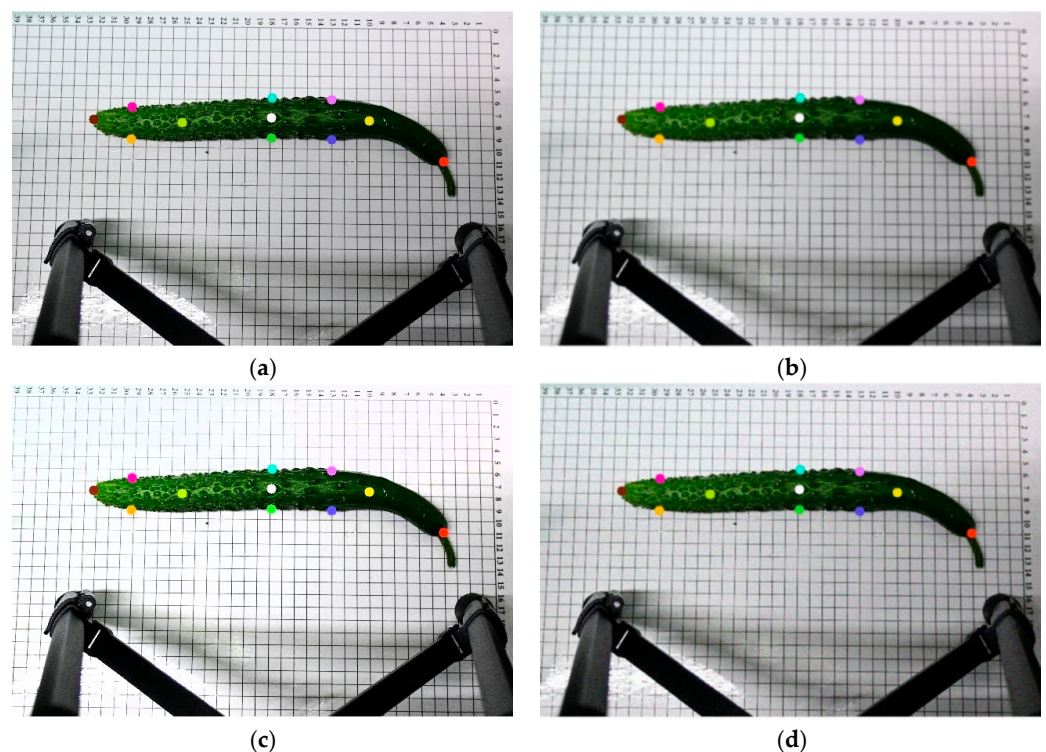


Figure 2. Schematic diagram of key point annotation for cucumber images. A total of 11 key landmarks were annotated, including: peduncle endpoint (p_0); points along the central axis of the cucumber (p_0 – p_4); cross-sectional diameter points located 3 cm from the peduncle end (p_5 – p_6); central cross-sectional diameter points (p_7 – p_8); and cross-sectional diameter points located 3 cm from the apex (p_9 – p_{10}).



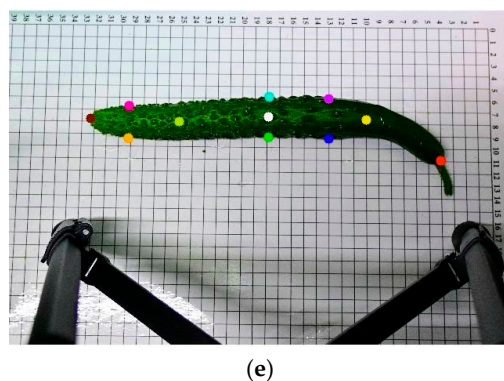


Figure 3. Data-augmented cucumber fruit images: (a) Original image. (b) Blurred image. (c) Brightness-enhanced image. (d) Noisy image. (e) Saturation-adjusted image.

2.2. Cucumber Phenotypic Parameter Calculation Method

The phenotypic traits of cucumbers primarily include fruit length, the ratio of stem length to fruit length, horizontal diameter difference, and arch height. To automate the extraction of these features, this study utilizes the normalized coordinates of annotated keypoints in the images. By combining image dimensions and a known scale reference, distances in pixel space are converted into actual physical measurements. This conversion process effectively corrects for scale variations caused by differences in image resolution and shooting conditions, ensuring the accuracy and consistency of the measurements. Based on this mapping, the relevant phenotypic parameters are derived and quantified using precise geometric formulas, as detailed below:

(1) Cucumber Length

$$L_c = \sum_{i=0}^3 \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \quad (1)$$

In the formula, L_c denotes the cucumber length; $(x_i, y_i) \in R^2$ represents the two-dimensional coordinates of the i -th keypoint on the cucumber's central axis.

(2) Peduncle-to-Fruit Length Ratio

$$L_s = \sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2} \quad (2)$$

$$R = \frac{L_s}{L_c} \quad (3)$$

In the formula, L_s denotes the stem length; R represents the ratio of stem length to cucumber length.

(3) Horizontal Diameter Difference

$$D = \max(|d_{top} - d_{mid}|, |d_{bottom} - d_{mid}|) \quad (4)$$

In the formula, D represents the horizontal diameter difference; d_{top} denotes the horizontal diameter measured 3 cm from the top of the cucumber; d_{mid} denotes the horizontal diameter at the center of the cucumber; and d_{bottom} denotes the horizontal diameter measured 3 cm from the bottom of the cucumber.

(4) Curvature Height

To overcome the limitations of traditional curvature height measurement methods that rely on complete contour information, this study proposes an indirect characterization approach to approximate the curvature height of cucumbers. Using conventional methods, cucumber samples were collected and their corresponding curvature heights

were extracted. These values were then converted into angular ranges formed by keypoints, establishing an interval mapping relationship between the traditional curvature height and the target angle.

The keypoints along the central axis of the cucumber are denoted as P_0, P_1, P_2, P_3, P_4 from right to left, where P_1, P_2 , and P_4 are used to determine the bending direction and calculate the angle. To distinguish the bending direction of the cucumber, the relative vertical positions of P_1 and P_4 are compared: if $y_4 < y_1$, the cucumber bends upward; if $y_4 > y_1$, the cucumber bends downward. Taking P_2 as the vertex, two vectors are constructed, as shown in Equations (5) and (6).

$$\vec{v_1} = \vec{P_2P_1} = (x_1 - x_2, y_1 - y_2) \quad (5)$$

$$\vec{v_2} = \vec{P_2P_4} = (x_4 - x_2, y_4 - y_2) \quad (6)$$

The formula for calculating the angle between two vectors is shown in Equation (7).

$$\theta = \cos^{-1} \left(\frac{\vec{v_1} \cdot \vec{v_2}}{\|\vec{v_1}\| \cdot \|\vec{v_2}\|} \right) \quad (7)$$

Let the curvature height of the cucumber be denoted as δ . The specific mapping relationship is as follows: $\theta \in [168.58^\circ, 180^\circ]$, $\delta \leq 0.5$ cm, $\theta \in [157.38^\circ, 168.58^\circ]$, 0.5 cm $< \delta \leq 1$ cm, $\theta \in [157.38^\circ, 168.58^\circ]$, and 1 cm $< \delta \leq 2$ cm.

2.3. Model Improvement Based on YOLO11m-Pose

The YOLO11m-Pose model is based on the YOLO architecture and integrates object detection with keypoint estimation techniques, enabling fast inference while maintaining high localization accuracy. As a result, it has been applied in the field of keypoint detection. By utilizing an end-to-end deep neural network structure, the model effectively captures spatial features and keypoint distributions of the target, significantly improving the efficiency of automated acquisition and analysis of phenotypic data. However, cucumber fruits exhibit complex and variable morphological characteristics, with similar appearances among keypoints such as the peduncle, ends, and arch height, making them difficult to distinguish. This increases the challenge for the model in expressing and fusing multi-scale features. Moreover, the backbone network used in YOLO11m-Pose has a relatively large number of parameters and high computational resource demands, which limits its applicability on resource-constrained edge devices or real-time systems. In addition, traditional keypoint localization loss functions lack sufficient sensitivity to positional deviations in keypoints located at edges or regions with morphological variation, resulting in room for improvement in localization accuracy in these critical areas.

Based on the aforementioned issues, this study introduces three improvements to the YOLO11m-Pose model and employs the enhanced YOLO11m-SCFPose model for training and testing. The final architecture of the proposed YOLO11m-SCFPose model is shown in Figure 4.

- (1) A lightweight backbone network, StarNet-S1, is adopted to replace the original backbone, significantly reducing computational complexity and improving operational efficiency.
- (2) To enhance the model's capability in representing and fusing multi-scale features, the efficient C3K2_PartialConv module is introduced into the neck network. This module strengthens feature interaction and fusion across scales while maintaining computational efficiency, thereby optimizing the overall detection performance.

- (3) The Focaler-IoU loss function is employed to replace the traditional loss function, improving the model's sensitivity to keypoint positional deviations. This enhancement leads to better overall keypoint localization accuracy and greater robustness, particularly in edge regions.

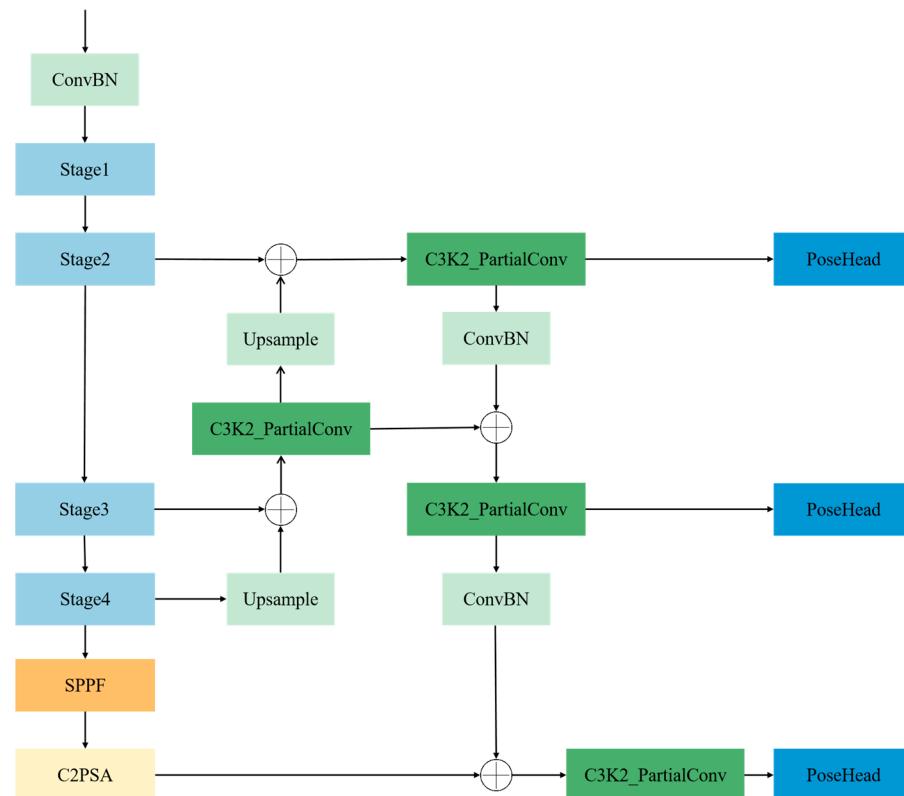


Figure 4. Schematic diagram of the YOLO11m-SCFPose model network structure.

2.3.1. StarNet-S1 Backbone Network

To further enhance the deployment efficiency and inference capability of the model on edge devices, the original backbone network in the YOLO11m model was replaced with the lightweight StarNet-S1 backbone. The StarNet series [31,32], designed specifically for efficient visual perception tasks, represents a class of lightweight convolutional neural networks that significantly reduce the number of parameters and computational complexity while maintaining high detection accuracy.

The architecture of StarNet-S1 consists of four hierarchically structured feature extraction stages (Stage1–Stage4), which include 2, 2, 8, and 3 deep residual blocks, respectively, forming a pyramidal structure. Each stage comprises a combination of downsampling modules and lightweight residual units, enabling progressive extraction of multi-level feature information from shallow to deep layers. StarNet-S1 incorporates depth-wise separable convolutions and point-wise convolutions in its modular design, decoupling the spatial and channel dimensions of traditional convolutions. This significantly reduces redundant computations. The architecture maintains a sufficient receptive field and strong feature representation capability while reducing the overall number of parameters, thereby improving runtime efficiency on edge devices. Additionally, the network integrates a multi-branch nonlinear activation mechanism. In this design, a portion of the channels undergo computationally expensive activation operations, while the remaining channels utilize more lightweight operations. This balances the model's nonlinear expressiveness with reduced computational complexity.

To enhance training stability and generalization capability, StarNet-S1 incorporates the DropPath strategy [33] during the training phase. The core idea of DropPath is to randomly drop certain residual connection paths during each forward pass, thereby reducing inter-layer dependencies and strengthening the model's regularization. The architecture of the StarNet-S1 backbone network is illustrated in Figure 5.

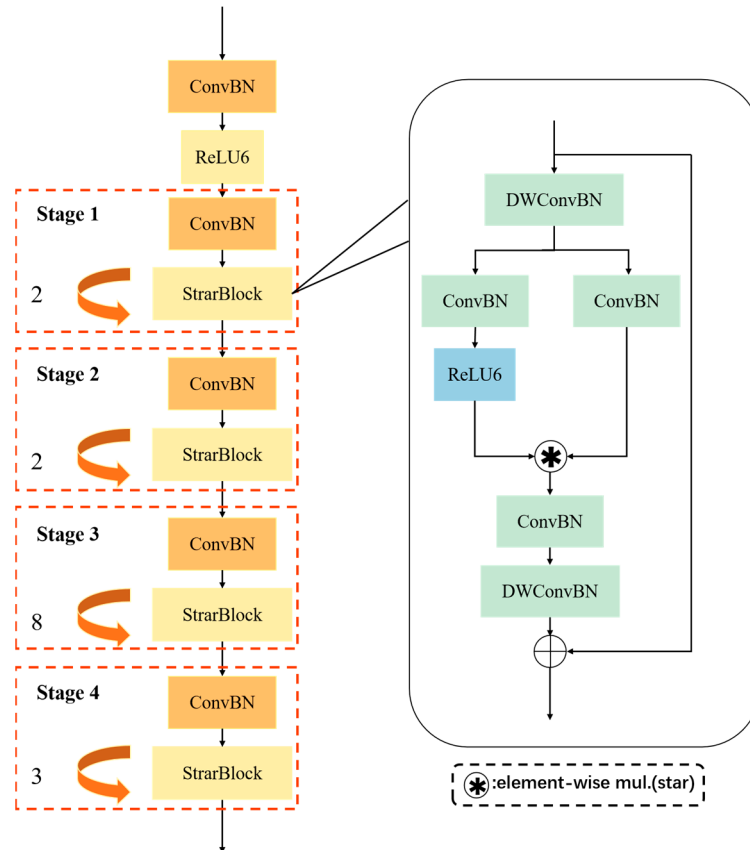


Figure 5. Schematic diagram of the backbone network structure of StarNet-S1.

2.3.2. C3K2_PartialConv Module

This study optimizes the structure of the C3K2 module in the neck network. The C3K2 module originates from the widely used C3 structure in the YOLO series networks. Its core consists of multiple residual-connected Bottleneck units, which effectively enhance the model's nonlinear modeling capability and semantic representation through cross-layer connections and feature reuse mechanisms. Compared to the standard C3 module, C3K2 deepens the convolutional path structurally, thereby further strengthening the ability to capture and fuse key features within local regions.

However, the C3K2 module still heavily relies on standard convolution operations involving all channels. Such convolutions perform joint computations across spatial and channel dimensions, which significantly increase the model's parameter count and computational complexity, especially in scenarios involving high-resolution image processing and multi-scale feature extraction. This high computational complexity severely limits the model's real-time deployment capability on edge computing devices, making it difficult to meet the practical requirements for low latency and low power consumption.

To address this, the present study proposes the C3K2_PartialConv module, which replaces the standard convolutions in the original module with Partial Convolution (PConv) [34], reducing computational costs while maintaining strong feature representation capability. The architecture of the C3K2_PartialConv module is shown in Figure 6. PConv is a lightweight convolution strategy that performs convolution operations on only

a subset of channels, while the remaining channels retain the input features directly, thereby reducing redundant convolution computations and improving computational efficiency. In the improved structure, the convolution channels are divided into two parts: one part is used to extract new feature representations, while the other part preserves the original information through skip connections and merges it at the output stage. This design not only significantly reduces the model's floating-point operations and parameter size but also alleviates information loss and gradient vanishing issues during deep feature learning, enhancing the network's stability and generalization performance in edge environments.

Let the spatial size of the input feature map be $h \times w$, the convolution kernel size be $k \times k$, and the total number of channels be c , where the number of channels involved in the convolution operation is c_p . The computational cost of PConv is shown in Equation (8).

$$FLOPs_{PConv} = h \times w \times k^2 \times c \times c_p \quad (8)$$

The computational cost of standard convolution is shown in Equation (9).

$$FLOPs_{PConv} = h \times w \times k^2 \times c^2 \quad (9)$$

when $c_p = \frac{1}{4}c$, the computational complexity of PConv can be reduced to one-quarter of that of standard convolution. By introducing PConv into the C3K2 module, the model's computational complexity and parameter count are effectively reduced. Activating only a portion of the channels ensures that the information flow in key feature channels is not weakened, thereby minimizing the loss of feature information.

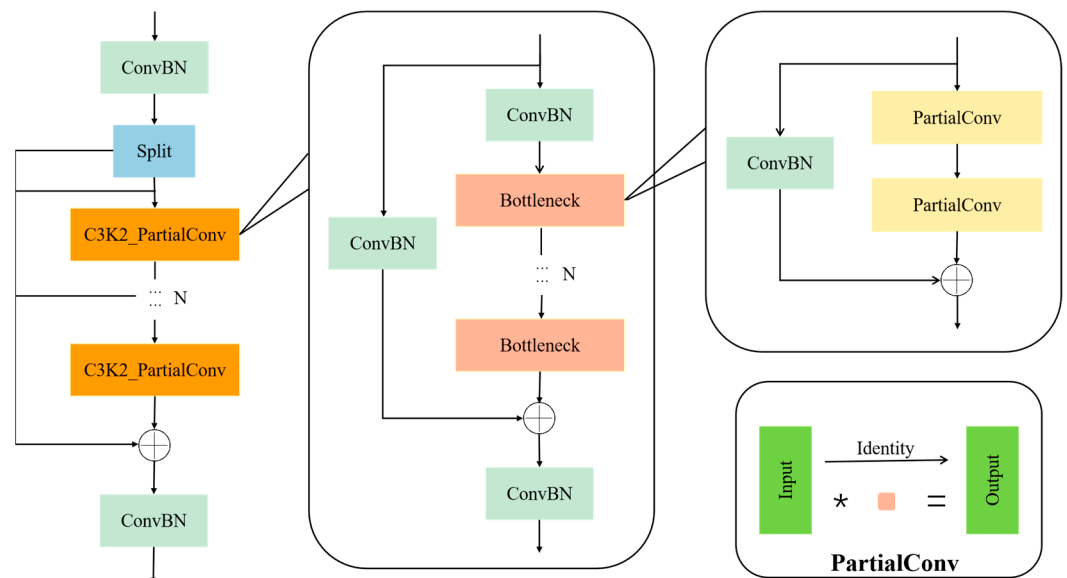


Figure 6. Schematic diagram of C3K2_PartialConv module structure. "*" represents convolution.

2.3.3. Optimization of Network Training Loss Function

Focaler-IoU [35] is a localization loss function improved based on the IoU loss, which combines a dynamic sample focusing mechanism with a gradient weight adjustment strategy to achieve adaptive weighting of different samples. This method reconstructs the loss response within a linear interval and establishes a mapping relationship between prediction error and gradient weight, effectively enhancing the model's attention to samples that are difficult to localize accurately. In this study, Focaler-IoU is applied to the keypoint detection task to strengthen the model's representation and learning ability for keypoint samples with large localization errors, thereby improving the accuracy and robustness of

keypoint localization. The formula of the Focaler-IoU loss function is shown in Equation (10).

$$IoU^{focaler} = \begin{cases} 0, & IoU < d \\ \frac{IoU - d}{u - d}, & d \ll IoU \ll u \\ 1, & IoU > u \end{cases} \quad (10)$$

In Focaler-IoU, a hard-thresholding function based on the IoU value is introduced to assign weights to predicted boxes in a piecewise manner according to their quality. The core idea of this function is to divide the predicted boxes into three intervals based on their IoU with the ground truth and assign different training weights accordingly. When the IoU is below the lower threshold d , the predicted box is regarded as a low-quality sample, and its weight is set to zero to avoid interfering with the optimization process. Within the intermediate interval $[d, u]$, a linear transition is applied to smoothly adjust the weight, enhancing the model's focus on hard-to-localize samples and increasing their contribution during gradient computation. When the IoU exceeds the upper threshold u , the prediction is considered sufficiently accurate, and the weight is fixed at 1 to prevent high-quality samples from dominating the training and causing overfitting. The loss is defined as shown in Equation (11).

$$L_{Folcaler-IoU} = 1 - IoU^{focaler} \quad (11)$$

2.4. Evaluation Metrics

In performance evaluation, the model uses the Object Keypoint Similarity (OKS) metric to assess the accuracy of cucumber fruit phenotypic keypoint detection. This metric measures the positional similarity between predicted keypoints and ground-truth keypoints by calculating their Euclidean distance. The OKS calculation formula is shown in Equation (12).

$$OKS_p = \frac{\sum_i \exp\left(-\frac{d_{pi}^2}{2s_p^2\sigma_i^2}\right) \cdot \delta(v_{pi} > 0)}{\sum_i \delta(v_{pi} > 0)} \quad (12)$$

In the formula, p_i denotes the i -th keypoint of the p -th object; d_{pi} represents the Euclidean distance between the predicted and ground-truth keypoints; v_{pi} is the visibility indicator of a keypoint, where 1 indicates that the keypoint is visible and annotated; s_p denotes the square root of the area of the object's bounding box; and σ_i represents the standard deviation for the i -th keypoint, and this study adopts the official values set by the COCO dataset.

The evaluation metrics of the model include Precision (P), Recall (R), F1 score, and mean Average Precision (mAP), which serve as the primary indicators for assessing cucumber phenotypic recognition performance. Precision refers to the proportion of correctly predicted positive keypoints among all keypoints predicted as positive, reflecting the accuracy of the predictions; Recall denotes the proportion of correctly predicted positive keypoints out of all actual positive keypoints, indicating the model's ability to capture relevant instances; and the F1 score is the harmonic mean of precision and recall, providing a balanced measure of both accuracy and completeness. In addition, floating point operations (GFLOPs) are used to measure the inference speed, while the number of parameters (Parameters) is used to evaluate the model's scale and complexity, serving as an indicator of its lightweight design. The formulas for computing Precision, Recall, and F1 score are provided in Equations (13)–(15).

$$P = \frac{TP}{TP + FP} \quad (13)$$

$$R = \frac{TP}{TP + FN} \quad (14)$$

$$F1 = 2 \times \frac{P \times R}{P + R} \quad (15)$$

In the equations, TP denotes the number of positive samples correctly predicted as positive; FP denotes the number of negative samples incorrectly predicted as positive; and FN denotes the number of positive samples incorrectly predicted as negative.

In this study, the evaluation method of keypoint accuracy and recall is not based on the traditional object detection statistics of TP, FP, and FN determined by the intersection over union (IoU) of bounding boxes, but instead adopts the object keypoint similarity to measure the matching degree between the model-predicted keypoints and the ground truth keypoints. By calculating the numbers of TP and FP under different object keypoint similarity thresholds (0.50, 0.95), the corresponding average precision mAP₅₀ and multi-threshold average precision mAP₅₀₋₉₅ are obtained, comprehensively quantifying the performance of the keypoint detection algorithm. The calculation formulas of mAP₅₀ and mAP₅₀₋₉₅ are shown in formulas (16) and (17).

$$mAP_{50} = AP_{50} \quad (16)$$

$$mAP_{50-95} = \frac{1}{10} \sum_{t=0.50}^{0.95} AP_t \quad (17)$$

To comprehensively assess the accuracy and stability of the proposed key-point regression model for cucumber fruit phenotypic trait prediction, three commonly used statistical error metrics are adopted, Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Standard Deviation (SD), to quantify the deviation between the model's predictions and the true values. Mean Absolute Error is a statistical measure representing the arithmetic mean of absolute prediction errors, reflecting the average discrepancy between predicted and actual observations. Root Mean Square Error is the square root of the mean of squared prediction errors, placing higher emphasis on larger errors. Standard Deviation is a standard statistical metric describing the dispersion of a random variable or error distribution, representing the average deviation of all data points from their mean. MAE, RMSE, and SD are computed according to Equations (18)–(20).

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (18)$$

y_i denotes the i -th observed value; $\hat{y}_i$ represents the corresponding predicted value; and n is the total number of samples.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (19)$$

y_i denotes the ground truth value; an y represents the predicted value; and n is the total number of samples.

$$SD = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2} \quad (20)$$

x denotes the i -th value in the sample; y represents the sample mean; and n is the total number of samples.

2.5. Experimental Platform and Parameter Settings

The model was trained for 350 epochs with a batch size of 16, an initial learning rate of 0.01, and an initial momentum of 0.937. Table 1 presents the experimental environment and configuration used in this study. PyTorch 2.4.0 was employed as the deep-learning framework, combined with Python 3.12.4 for image preprocessing, model construction, and training. The training process leveraged CUDA 12.1.1 for GPU acceleration to improve computational efficiency. The operating environment was based on the Ubuntu system, and the hardware setup included an Intel Xeon E5-2680 processor and an NVIDIA GeForce RTX 3090 Ti graphics card, ensuring stability and high performance during large-scale image data training and inference.

Table 1. Experimental environment and configuration.

Configuration	Parameter
Development Framework	PyTorch 2.4.0
Programming Language	Python 3.12.4
Training Environment	CUDA 12.1.1
Operating System	Ubuntu
CPU	Intel Xeon E5-2680
GPU	NVIDIA GeForce RTX 3090Ti

3. Results and Analysis

3.1. Loss Function Comparison Experiments and Results

To further validate the effectiveness of the Focaler-IoU loss function in keypoint detection tasks, this study conducted multiple comparative experiments using different loss functions under a unified model architecture and consistent training settings to systematically evaluate their performance on the same dataset. As shown in Table 2, compared with commonly used loss functions such as CIoU, PIoU, ShapeIoU, and Inner-SIoU, Focaler-IoU achieved an accuracy of 0.924, outperforming the others and indicating its significant advantage in precise localization and error regression. Traditional loss functions typically assign similar optimization weights to all samples even when there are large discrepancies in error magnitude, which reduces the model's ability to distinguish between easy and hard samples and weakens its learning capacity for complex sample features. Focaler-IoU introduces a dynamic weighting mechanism through the Focal Loss adjustment factor, assigning higher weights to samples with large prediction errors and guiding the model to focus more on hard-to-predict keypoint regions during training. Although using PIoU alone improves the model's capability to capture key features compared to CIoU, integrating PIoU into Focaler-IoU leads to decreased model accuracy, suggesting that the additional penalty terms from PIoU, when dynamically weighted, amplify error fluctuations and negatively impact overall model performance. Overall, the results demonstrate that Focaler-IoU improves localization accuracy and stability in keypoint detection tasks, enhances the representational ability of the model for keypoint structures, and strengthens generalization performance.

Table 2. Comparison test results of different loss functions.

Loss Function	P	R	F1	mAP50	mAP50-95
CIoU	0.997	0.998	0.998	0.994	0.900
PIoU	0.999	0.999	0.999	0.995	0.905
ShapeIoU	0.999	0.998	0.999	0.995	0.900
NWDloss	0.991	0.992	0.992	0.994	0.559
Inner-SIoU	0.997	0.998	0.998	0.994	0.898

Focaler-PIoU	0.999	0.999	0.999	0.995	0.906
Focaler-IoU	0.999	0.999	0.999	0.995	0.924

3.2. Model Ablation Experiments and Results

To systematically evaluate the enhancement effects of the StarNet-S1 backbone network, the C3K2_PartialConv module, and the Focaler-IoU loss function on the performance of YOLO11m-Pose, this study designed five ablation experiments to analyze the impact of each module and their different combinations on keypoint detection accuracy, computational complexity, and model size.

The results of the keypoint detection experiments are shown in Table 3. To improve the model's inference efficiency, the original backbone was replaced with the lightweight StarNet-S1. This module reduces the model's floating-point operations (GFLOPs) to 37.8 through deep compression and convolutional structure reconstruction while also decreasing the parameter size, thereby improving inference speed while retaining fundamental feature extraction capability. However, due to the reduced model capacity and consequent decline in representational power, the detection accuracy slightly decreased compared to the original model. To further enhance the model's feature modeling ability while controlling network complexity, the C3K2_PartialConv module was introduced to improve semantic construction in intermediate layers and multi-scale feature fusion. Results show that this module can improve the model's accuracy in keypoint detection without increasing computational cost or parameter size. When StarNet-S1 and the C3K2_PartialConv module are used together, the overall computational complexity further decreases, while mAP50-95 improves to a higher level, indicating good complementarity between the two that maintains detection performance while compressing model size. On this basis, replacing the loss function with Focaler-IoU, which incorporates IoU awareness and reweighting strategies for positive and negative samples, further enhances the model's localization accuracy and robustness, with mAP50-95 increasing to 0.924. Overall, the improved model structure significantly optimizes the comprehensive performance of YOLO11m-Pose, validating the feasibility and practical value of the aforementioned modules in this keypoint detection task.

Table 3. Ablation experiment results.

StarNet-S1	C3K2_Partial Conv	Focaler-IoU	P	R	F1	mAP50	mAP50-95	GFLOPs	Parameters
			0.998	0.998	0.998	0.994	0.926	71.4	2.089×10^7
√			0.997	0.998	0.998	0.994	0.733	37.8	1.409×10^7
	√		0.998	0.999	0.999	0.994	0.923	65.6	1.909×10^7
√	√		0.998	0.998	0.998	0.995	0.900	32.1	1.229×10^7
√	√	√	0.999	0.999	0.999	0.995	0.924	32.1	1.229×10^7

Note: A “√” in the table indicates that the corresponding component is used.

3.3. Comparison Experiments and Results of Different Models

To verify the performance of the improved model, we compared the training results of the proposed YOLO11m-SCFPose model with YOLOv7-w6-Pose, YOLOv8m-Pose, YOLOv9-Pose, YOLOv10m-Pose, and YOLOv11m-Pose on the same dataset, as shown in Table 4. The experimental results indicate that YOLOv7-w6-Pose achieves high detection accuracy, but its floating-point operations reach 101.9 GFLOPs and the parameter size reaches 8.01×10^7 , significantly increasing the computational resource requirements and making it unsuitable for deployment on resource-constrained edge devices. In contrast, YOLOv8m-Pose, YOLOv9-Pose, and YOLOv10m-Pose maintain mAP50-95 above 0.9 while keeping relatively low computational complexity. However, their GFLOPs and

model sizes still fall short of ideal efficiency. YOLOv11m-Pose improves the mAP50-95 to 0.926 under relatively low computational resource consumption, demonstrating strong feature extraction capabilities. Building upon this, the proposed YOLO11m-SCFPose further optimizes the backbone structure and compresses redundant computation paths, reducing GFLOPs to 32.1 and parameters to 1.23×10^7 . While maintaining an accuracy of 0.924, it significantly improves computational efficiency. These results demonstrate that compared with other models, the improved YOLO11m-SCFPose offers superior computational efficiency and lower resource consumption without compromising detection accuracy.

Table 4. Results of model comparison.

Model	P	R	F1	mAP50	mAP50-95	GFLOPs	Parameters
YOLOv7-w6-Pose	0.996	0.999	0.998	0.996	0.697	101.9	8.010×10^7
YOLOv8m-Pose	0.997	0.998	0.998	0.995	0.918	69.6	2.377×10^7
YOLOv9-Pose	0.998	0.999	0.999	0.995	0.917	101.4	2.796×10^7
YOLOv10m-Pose	0.992	0.998	0.995	0.995	0.735	70.7	2.038×10^7
YOLOv11m-Pose	0.999	0.999	0.999	0.995	0.926	71.4	2.089×10^7
YOLO11m-SCFPose	0.999	0.999	0.999	0.995	0.924	32.1	1.229×10^7

As shown in Figure 7, to more intuitively compare the differences among the models in terms of mAP50-95, computational complexity, and parameter scale, the corresponding line chart was plotted. This figure presents the variation trends of model performance metrics and computational resource consumption, facilitating a comprehensive evaluation of the trade-offs between accuracy and efficiency for each model. The results indicate that the proposed model maintains a high mAP50-95 while effectively reducing computational complexity and parameter count, demonstrating excellent lightweight design advantages and strong potential for practical applications.

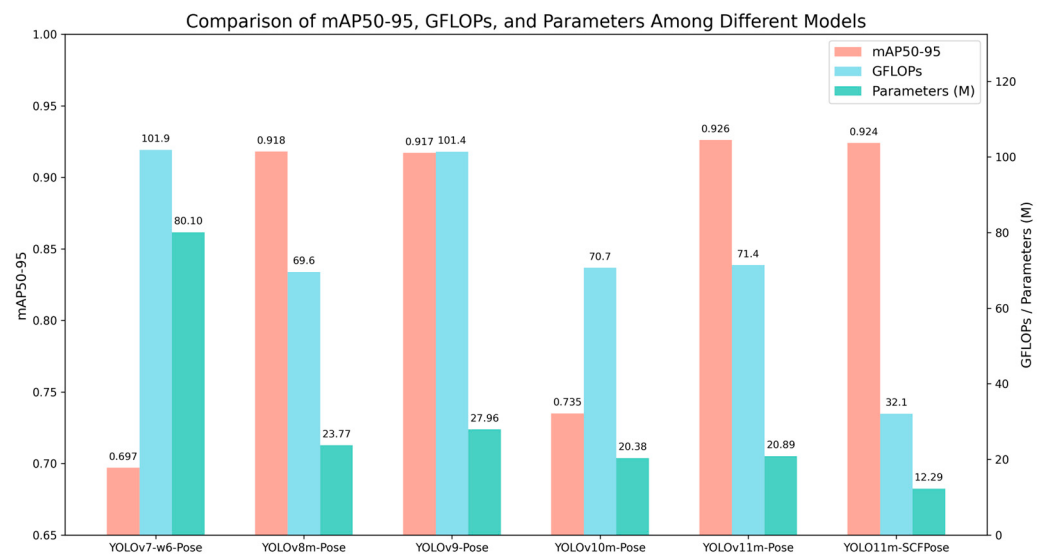
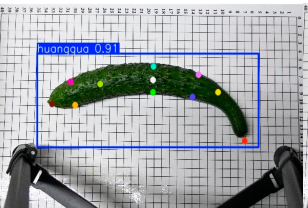
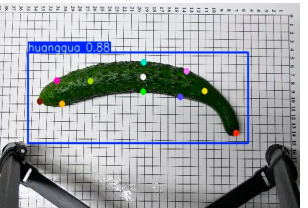
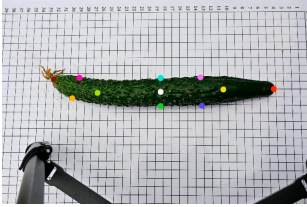
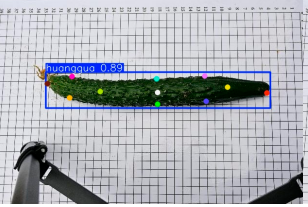
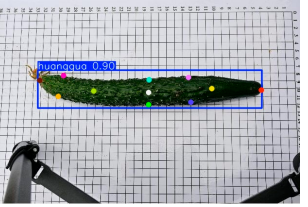
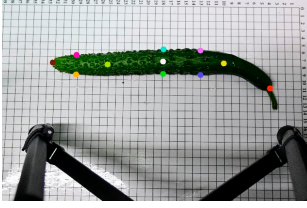
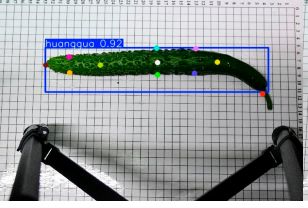
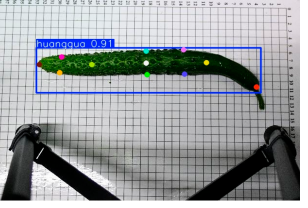
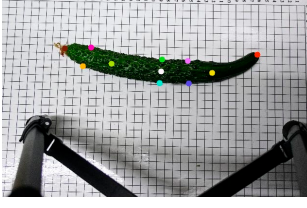
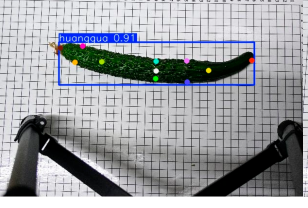
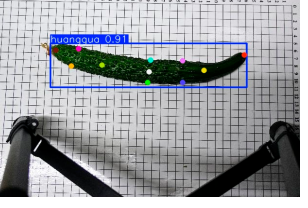
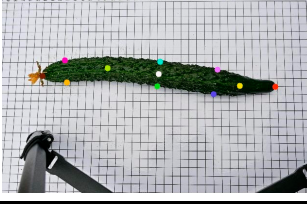
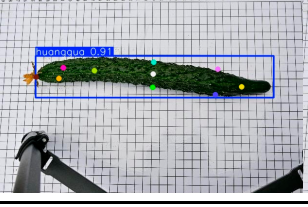
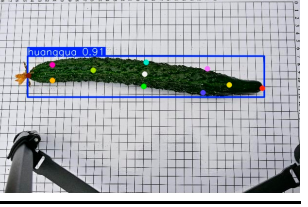


Figure 7. Comparison of mAP50-95, GFLOPs, and parameters among different models.

3.4. Evaluation of Cucumber Keypoint Detection Performance

To intuitively demonstrate the performance of the YOLO11m-SCFPose model in cucumber keypoint detection tasks, this study randomly selected four sets of cucumber images for keypoint detection visualization, including the original annotation diagrams, the keypoint detection results of the YOLO11m-Pose model, and those of the improved YOLO11m-SCFPose model, as shown in Table 5. Each detected image includes model-generated bounding boxes, along with the corresponding class confidence scores, which reflect the model’s confidence in identifying the cucumber target regions. The displayed keypoints include a total of 11 landmarks: the stem point, both ends of the fruit, the central point, two endpoints of the cross-sectional diameter located 3 cm from both the fruit apex and base, and two endpoints of the central cross-sectional diameter. The results demonstrate that YOLO11m-SCFPose achieves more accurate spatial alignment of keypoints with the original annotations, particularly in the stem and cross-sectional regions, exhibiting higher geometric consistency and localization precision. Compared to YOLO11m-Pose, YOLO11m-SCFPose shows superior performance in keypoint localization and structural consistency, further validating the effectiveness and applicability of the proposed improvements for cucumber phenotypic trait extraction.

Table 5. Keypoint detection results.

	Annotation Diagram	YOLO11m-Pose Key-point Detection Results	YOLO11m-SCFPose Key-point Detection Results
Group 1			
Group 2			
Group 3			
Group 4			
Group 5			

To further evaluate the completeness of keypoint detection, the number of detected keypoints was analyzed, as shown in Table 6. The test set contains a total of 227 images, each annotated with 11 keypoints, resulting in 2497 keypoints in total. The original YOLO11m-Pose model detected 2492 keypoints, with the p_0 point missing in five images. Additionally, although some keypoints were detected, their predicted locations deviated significantly from the ground truth, which affected the accuracy of structural modeling. In contrast, the improved YOLO11m-SCFPose model successfully detected all keypoints in every image, and its predicted positions were closer to the annotations, providing a more accurate and stable basis for phenotypic feature extraction.

Table 6. Results of keypoint detection counts.

	p_0	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}
YOLO11m-Pose	222	227	227	227	227	227	227	227	227	227	227
YOLO11m-SCFPose	227	227	227	227	227	227	227	227	227	227	227

To further evaluate the performance of the proposed model, four key metrics were computed and visualized: Precision, Recall, mean Average Precision (mAP50–95), and the Precision–Recall (PR) curve. Precision and Recall respectively measure the accuracy and completeness of keypoint predictions, while mAP50–95 represents the mean Average Precision across IoU thresholds ranging from 0.5 to 0.95 (in steps of 0.05), offering a comprehensive assessment of model performance under varying levels of matching strictness. The PR curve illustrates the trade-off between Precision and Recall across different confidence thresholds, confirming the stability and robustness of the model in cucumber keypoint detection tasks. The visualized results are shown in Figure 8.

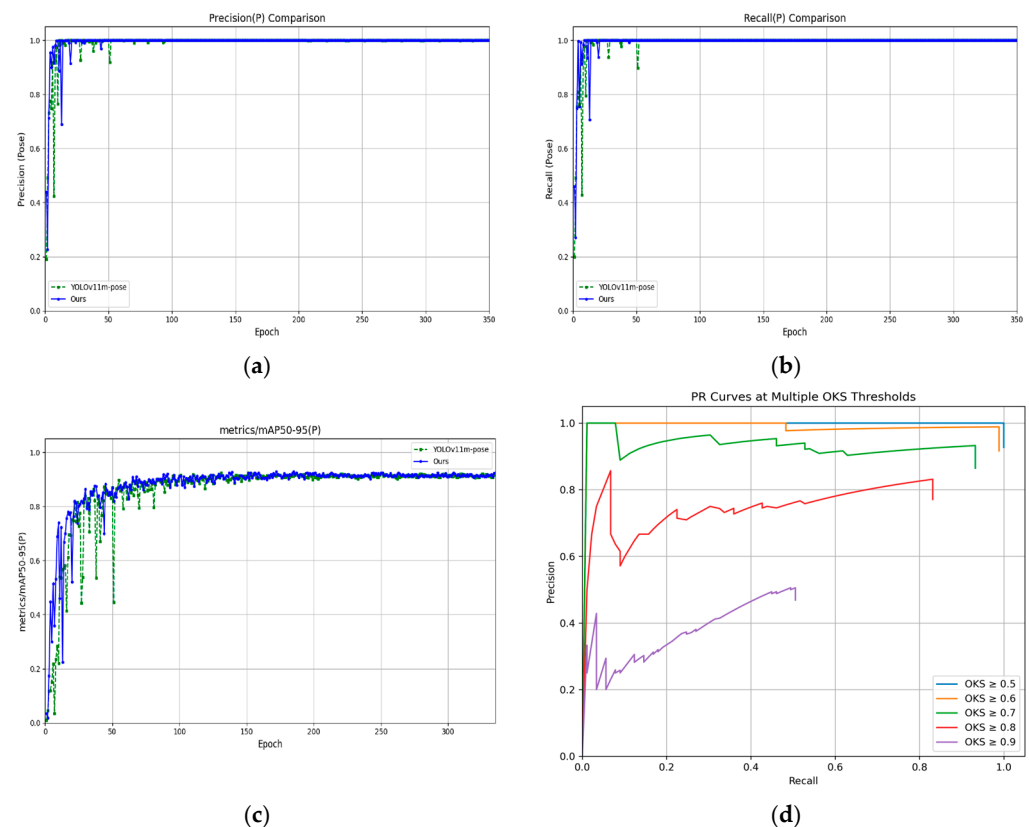


Figure 8. Comprehensive analysis of model performance. (a) Precision (P) curve. (b) Recall (R) curve. (c) mAP50–95 curve. (d) Precision–Recall (PR) curves at multiple OKS thresholds.

3.5. Comparison and Analysis of Cucumber Fruit Phenotypic Measurement Results

To evaluate the performance of the proposed YOLO11m-SCFPose model in cucumber fruit phenotypic trait extraction, ten cucumber samples were randomly selected, and four key phenotypic traits were analyzed by comparing predicted values with manually annotated ground truths. These traits included fruit length, fruit-stem-to-length ratio, lateral diameter difference, and curvature height. Figure 9 illustrates the variation trends of predicted and actual values for each trait across different samples.

The results demonstrate that YOLO11m-SCFPose exhibits high accuracy in fruit length prediction, with the predicted curves closely matching the actual values in both magnitude and trend. This indicates the model's robustness and reliability in scale modeling and length estimation tasks. For the structural trait of fruit-stem-to-length ratio, the model shows stable fitting performance, reflecting strong perceptual and modeling capabilities regarding the structural proportions of cucumber fruits. In the prediction of more morphologically complex traits—lateral diameter difference and curvature height—YOLO11m-SCFPose effectively captures key geometric variations. The generated curves align well with the annotated values, particularly in curvature height, where the model demonstrates notable sensitivity to curvature. These findings further validate the model's adaptability and expressive power in extracting complex morphological features.

Comparison Between Predicted and Ground Truth for Cucumber Phenotypic Traits

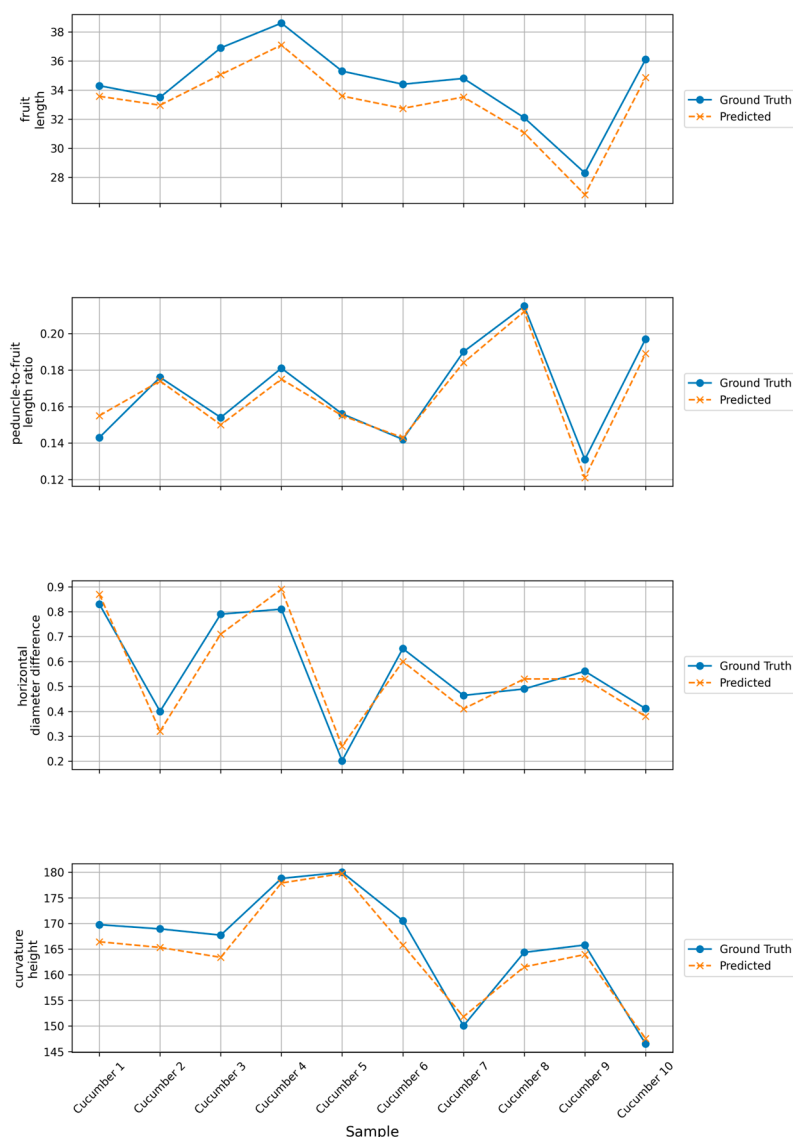


Figure 9. Comparison between predicted and ground truth for cucumber phenotypic traits.

The computed results of MAE, RMSE, and SD are presented in Table 7. The results demonstrate that the proposed model exhibits stable performance across multiple phenotypic trait prediction tasks in cucumber, with consistently low prediction errors, indicating favorable numerical accuracy. For the prediction of fruit length, the Mean Absolute Error (MAE) is 1.304 and the Root Mean Square Error (RMSE) is 1.366, both reflecting a low level of error and the model's effective fitting capability for this continuous trait. For the ratio of peduncle length to fruit length, the MAE is 0.0053 and the RMSE is 0.0064, suggesting that the model achieves highly stable and minimal error in predicting ratio-based traits. For the transverse diameter difference, the MAE and RMSE are 0.0546 and 0.0577, respectively, indicating a narrow error range and strong consistency in estimating this relative trait. The arch height, expressed in angular units, yields an MAE of 2.468 and an RMSE of 2.865; the errors fall within an acceptable range, showing that the model can accurately capture this angular characteristic. Overall, the model maintains low errors and minimal variability across diverse phenotypic traits, demonstrating strong predictive stability and practical applicability for automated cucumber phenotyping. Future work may focus on further enhancing the model's capacity to fit nonlinear and complex features, aiming for more precise extraction of phenotypic parameters.

Table 7. Performance evaluation of the proposed model on phenotypic trait prediction.

	MAE	RMSE	SD
Cucumber Length	1.3040	1.3656	0.4054
Peduncle-to-Fruit Length Ratio	0.0053	0.0064	0.0058
Horizontal Diameter Difference	0.0546	0.0577	0.0567
Curvature Height	2.4680	2.8649	2.1407

4. Discussion

In cucumber variety breeding and fruit commodity quality evaluation, morphological parameters such as fruit length, peduncle-to-fruit length ratio, transverse diameter difference, and arch height are key phenotypic indicators. These parameters not only affect the external quality of the fruit but are also closely related to market acceptance, transportation adaptability, and management decisions during cultivation. Traditional measurement methods primarily rely on manual operations, such as grid methods or ruler measurements. Although these methods are simple to perform, they are inefficient for large-scale planting or breeding trials and are prone to significant subjective errors. This is especially true when fruit morphology is irregular or exhibits considerable curvature, resulting in poor consistency and reliability of measurements, which in turn affects the scientific basis of breeding strategies and cultivation management.

The keypoint detection method based on YOLOv11m-Pose proposed in this study derives the aforementioned phenotypic parameters from the detected keypoint locations, preliminarily validating the feasibility and practicality of this approach for automated phenotypic analysis of cucumber fruits. Experimental results demonstrate that the model achieves a mean absolute error (MAE) of 1.304 and a root mean square error (RMSE) of 1.366 in predicting fruit length, indicating good fitting capability. For the prediction of the peduncle-to-fruit length ratio, the errors are stable and low, with an MAE of 0.0053 and an RMSE of 0.0064. The errors for transverse diameter difference and arch height also fall within reasonable ranges, with MAEs of 0.0546 and 2.468 and RMSEs of 0.0577 and 2.865, respectively. The improved model further enhances performance, achieving an mAP₅₀₋₉₅ of 0.924, reducing floating-point operations (GFLOPs) to 32.1, and compressing the model parameter count to 1.229×10^7 . These results indicate that the model maintains high

stability and accuracy across various phenotypic parameter extraction tasks, demonstrating strong advantages in lightweight design and promising potential for practical applications.

Although the current model performs stably under standard acquisition conditions, variations in fruit morphology and lighting changes in greenhouse environments may pose greater challenges to its generalization ability. Despite the inclusion of a certain degree of sample diversity in the experiments, practical deployment still requires appropriate fine-tuning or incremental learning tailored to specific production scenarios. As this technology advances toward broader applications, further refinement in model design and system integration is necessary to enhance adaptability and reliability.

5. Conclusions

This study addresses the limitations of traditional manual methods for cucumber fruit phenotypic trait extraction, which suffer from low efficiency, high error rates, and difficulties in automation. We propose a keypoint detection-based approach for extracting cucumber phenotypic features. A YOLO11m-SCFPose model based on YOLOv11m-Pose was constructed, wherein optimizations were made to the model architecture and loss function to achieve accurate localization of cucumber keypoints and a lightweight model design. The model incorporates the lightweight StarNet-S1 structure in the backbone network to replace the original modules, significantly improving inference speed and deployment flexibility while substantially reducing the number of parameters without compromising detection accuracy. In the neck, the traditional C3K2 module is replaced by an improved C3K2_PartialConv module, enhancing information interaction among multi-scale features and target representation capability while reducing computational resource consumption. To improve the model's sensitivity to keypoint localization errors, especially at edge regions and low-confidence samples, the Focaler-IoU loss function was introduced to enhance robustness and localization precision. Experimental results demonstrate that YOLO11m-SCFPose achieves excellent accuracy and computational efficiency in cucumber keypoint detection tasks, reliably extracting key phenotypic parameters such as fruit length, peduncle-to-fruit length ratio, transverse diameter difference, and arch height while maintaining low model complexity and enabling efficient computation.

Although the proposed YOLO11m-SCFPose model has achieved promising results in keypoint detection tasks on static images, certain limitations remain in its practical deployment and large-scale application. The current system is based on static image analysis and lacks the capability to process dynamic scenarios, which hinders its ability to meet the demands for real-time and continuous extraction of fruit phenotypic parameters. Extending this method to video stream data processing is expected to enhance its practicality and timeliness in actual agricultural production. To advance the system toward higher levels of automation and integration, it is imperative to strengthen the collaborative design with customized image acquisition devices, thereby constructing a complete measurement system that integrates image capture, keypoint detection, and phenotypic parameter output, which can improve overall system stability and deployment efficiency. In the follow-up work, we will further explore the dynamic image acquisition method based on the professional cucumber keypoint acquisition device to achieve efficient and stable acquisition of keypoint information. By integrating technical means such as high-quality industrial cameras, stable tracking, and synchronous labeling, the accuracy and automation level of data collection are improved so as to provide higher-quality input data for cucumber key point detection model and then optimize the detection efficiency and deployment convenience of the model in practical applications.

Author Contributions: The authors confirm contribution to the paper as follows: Study conception and design: H.Y., X.M. and J.Y. data collection: H.Y. and X.Z.; Analysis and interpretation of results: H.Y. and X.M.; Draft manuscript preparation: H.Y. and X.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Shandong Province Higher Education Program for the Introduction and Cultivation of Young Innovative Talents (2021).

Data Availability Statement: The data that support the findings of this study are available from the corresponding author, X.M. upon reasonable request.

Acknowledgments: The authors gratefully acknowledge the Shandong Juye Facility Cucumber Science and Technology Backyard for providing essential data support for this study. Their generous assistance in offering access to high-quality cucumber phenotypic data greatly contributed to the success of the experimental evaluation.

Conflicts of Interest: The authors declared no conflicts of interest.

References

1. Zhao, C.J.; Zhang, Y.; Du, J.J.; Guo, X.Y.; Wen, W.L.; Gu, S.H.; Wang, J.L.; Fan, J.C. Crop Phenomics: Current Status and Perspectives. *Front. Plant Sci.* **2019**, *10*, 714. <https://doi.org/10.3389/fpls.2019.00714>.
2. Shi, Y.Y.; Zhu, Y.; Wang, X.C.; Sun, X.; Ding, Y.F.; Cao, W.X.; Hu, Z.C. Progress and development on biological information of crop phenotype research applied to real-time variable-rate fertilization. *Plant Methods* **2020**, *16*, 11. <https://doi.org/10.1186/s13007-020-0559-9>.
3. Meraj, T.; Sharif, M.I.; Raza, M.; Alabrah, A.; Kadry, S.; Gandomi, A.H. Computer vision-based plants phenotyping: A comprehensive survey. *Iscience* **2024**, *27*, 108709. <https://doi.org/10.1016/j.isci.2023.108709>.
4. Bai, X.; Wang, X.; Liu, X.L.; Liu, Q.; Song, J.K.; Sebe, N.; Kim, B. Explainable Deep Learning for Efficient and Robust Pattern Recognition: A Survey of Recent Developments. *Pattern Recognit.* **2021**, *120*, 108102. <https://doi.org/10.1016/j.patcog.2021.108102>.
5. Grumet, R.; Lin, Y.C.; Rett-Cadman, S.; Malik, A. Morphological and Genetic Diversity of Cucumber (*Cucumis sativus* L.) Fruit Development. *Plants* **2023**, *12*, 23. <https://doi.org/10.3390/plants12010023>.
6. Shi, Y.Y.; Zhu, Y.; Wang, X.C.; Sun, X.; Ding, Y.F.; Cao, W.X.; Hu, Z.C. Mapping of fruit apex shape related QTLs across multi-genetic backgrounds in cucumber (*Cucumis sativus* L.). *Hortic. Plant J.* **2022**, *8*, 328–340. <https://doi.org/10.1016/j.hpj.2021.12.001>.
7. Santos, C.F.G.; Arrais, R.R.; Silva, J.V.S.; Silva, M.H.M.; Neto, W.B.G.; Lopes, L.T.; Bileki, G.A.; Lima, I.O.; Rondon, L.B.; Souza, B.M.; et al. ISP Meets Deep Learning: A Survey on Deep Learning Methods for Image Signal Processing. *ACM Comput. Surv.* **2025**, *57*, 1–44. <https://doi.org/10.1145/3708516>.
8. Murphy, K.M.; Ludwig, E.; Gutierrez, J.; Gehan, M.A. Deep Learning in Image-Based Plant Phenotyping. *Annu. Rev. Plant Biol.* **2024**, *75*, 771–795. <https://doi.org/10.1146/annurev-arplant-070523-042828>.
9. Qiu, Z.; Huang, Z.Y.; Mo, D.Y.; Tian, X.J.; Tian, X.Y. GSE-YOLO: A Lightweight and High-Precision Model for Identifying the Ripeness of Pitaya (Dragon Fruit) Based on the YOLOv8n Improvement. *Horticulturae* **2024**, *10*, 852. <https://doi.org/10.3390/horticulturae10080852>.
10. Xu, H.B.; Fu, L.X.; Li, J.N.; Lin, X.Y.; Chen, L.X.; Zhong, F.L.; Hou, M.M. A Method for Analyzing the Phenotypes of Nonheading Chinese Cabbage Leaves Based on Deep Learning and OpenCV Phenotype Extraction. *Agronomy* **2024**, *14*, 699. <https://doi.org/10.3390/agronomy14040699>.
11. Ndikumana, J.N.; Lee, U.; Yoo, J.H.; Yeboah, S.; Park, S.H.; Lee, T.S.; Yeoung, Y.R.; Kim, H.S. Development of A Deep-learning Phenotyping Tool for Analyzing Image-based Strawberry Phenotypes. *Front. Plant Sci.* **2024**, *15*, 1418383. <https://doi.org/10.3389/fpls.2024.1418383>.
12. Zhang, X.B.; Zhu, Y.H.; Su, Y.L.; Xie, B.L.; Gu, Q.; Zheng, K.F. Quantitative Extraction and Analysis of Pear Fruit Spot Phenotypes Based on Image Recognition. *Comput. Electron. Agric.* **2021**, *190*, 106474. <https://doi.org/10.1016/j.compag.2021.106474>.
13. Wei, Q.; Wang, Y.L.; Yang, S.L.; Guo, C.H.; Wu, L.S.; Yin, H. Moving toward Automaticity: A Robust Synthetic Occlusion Image Method for High-Throughput Mushroom Cap Phenotype Extraction. *Agronomy* **2024**, *14*, 1337. <https://doi.org/10.3390/agronomy14061337>.

14. Wang, Z.C.; Cui, W.Y.; Huang, C.J.; Zhou, Y.H.; Zhao, Z.H.; Yue, Y.C.; Dong, X.R.; Lv, C.L. Framework for Apple Phenotype Feature Extraction Using Instance Segmentation and Edge Attention Mechanism. *Agriculture* **2025**, *15*, 305. <https://doi.org/10.3390/agriculture15030305>.
15. Gai, R.L.; Gao, J.; Xu, G.H. HPPEM: A High-Precision Blueberry Cluster Phenotype Extraction Model Based on Hybrid Task Cascade. *Agronomy* **2024**, *14*, 1178. <https://doi.org/10.3390/agronomy14061178>.
16. Cao, L.Y.; Li, S.L.; Jiang, D.H.; Sun, M.; Liu, X.G. EGF-Former: An Efficient Network for Structural Segmentation and Phenotype extraction of Sweet Peppers in Complex Environments. *Ind. Crop. Prod.* **2025**, *227*, 120850. <https://doi.org/10.1016/j.indcrop.2025.120850>.
17. Zhai, Z.Y.; Zhang, Z.H.; Xu, H.L.; Wang, H.Q.; Chen, X.; Yang, C.M. Review of Applying YOLO Family Algorithms to Analyze Animal and Plant Phenotype. *Trans. Chin. Soc. Agric. Mach.* **2024**, *55*, 1–20. <https://doi.org/10.6041/j.issn.1000-1298.2024.11.001>.
18. Peng, C.; Cao, S.S.; Li, S.J.; Bai, T.; Zhao, Z.Y.; Sun, W. Automated Measurement of Cattle Dimensions Using Improved Keypoint Detection Combined with Unilateral Depth Imaging. *Animals* **2024**, *14*, 2453. <https://doi.org/10.3390/ani14172453>.
19. Su, L.D.; Li, M.H.; Zhang, Y.; Zong, Z.Y.; Gong, C.L. Fusion of Target and Keypoint Detection for Automated Measurement of Mongolian Horse Body Measurements. *Agriculture* **2024**, *14*, 1069. <https://doi.org/10.3390/agriculture14071069>.
20. Li, J.X.; Liu, Y.H.; Zheng, W.X.; Chen, X.W.; Ma, Y.B.; Guo, L.F. Monitoring Cattle Ruminating Behavior Based on an Improved Keypoint Detection Model. *Animals* **2024**, *14*, 1791. <https://doi.org/10.3390/ani14121791>.
21. Chen, K.; Chen, Z.Q.; Wang, C.B.; Zhou, Z.F.; Xiao, M.H.; Zhu, H.; Li, D.F.; Liu, W.M. Improved YOLOv8-Based Method for the Carapace Keypoint Detection and Size Measurement of Chinese Mitten Crabs. *Animals* **2025**, *15*, 941. <https://doi.org/10.3390/ani15070941>.
22. He, J.J.; Weng, L.; Xu, X.G.; Chen, R.C.; Peng, B.; Li, N.N.; Xie, Z.C.; Sun, L.J.; Han, Q.; He, P.F. DEKR-SPrior: An Efficient Bottom-Up Keypoint Detection Model for Accurate Pod Phenotyping in Soybean. *Plant Phenomics* **2024**, *6*, 0198. <https://doi.org/10.34133/plantphenomics.0198>.
23. Zheng, C.; Chen, P.F.; Pang, J.; Yang, X.F.; Chen, C.X.; Tu, S.Q.; Xue, Y.J. A Mango Picking Vision Algorithm on Instance Segmentation and Key Point Detection from RGB Images in an Open Orchard. *Biosyst. Eng.* **2021**, *206*, 32–54. <https://doi.org/10.1016/j.biosystemseng.2021.03.012>.
24. Du, X.Q.; Meng, Z.C.; Ma, Z.H.; Lu, W.W.; Cheng, H.C. Tomato 3D Pose Detection Algorithm Based on Keypoint Detection and Point Cloud Processing. *Comput. Electron. Agric.* **2023**, *212*, 108056. <https://doi.org/10.1016/j.compag.2023.108056>.
25. Gao, Y.L.; Li, Z.; Li, B.; Zhang, L.F. Extraction of Corn Plant Phenotypic Parameters with Keypoint Detection and Stereo Images. *Agronomy* **2024**, *14*, 1110. <https://doi.org/10.3390/agronomy14061110>.
26. Chen, J.Q.; Ma, A.Q.; Huang, L.X.; Li, H.W.; Zhang, H.Y.; Huang, Y.; Zhu, T.T. Efficient and Lightweight Grape and Picking Point Synchronous Detection Model Based on Key Point Detection. *Comput. Electron. Agric.* **2024**, *217*, 108612. <https://doi.org/10.1016/j.compag.2024.108612>.
27. Zheng, J.M.; Wang, X.C.; Shi, Y.Y.; Zhang, X.L.; Wu, Y.; Wang, D.Z.; Huang, X.K.; Wang, Y.X.; Wang, J.H.; Zhang, J.F. Keypoint Detection and Diameter Estimation of Cabbage (*Brassica oleracea* L.) Heads under Varying Occlusion Degrees via YOLOv8n-CK Network. *Comput. Electron. Agric.* **2024**, *226*, 109428. <https://doi.org/10.1016/j.compag.2024.109428>.
28. Wang, J.N.; Tan, D.Z.; Sui, L.M.; Guo, J.; Wang, R.W. Wolfberry Recognition and Picking-point Localization Technology in Natural Environments Based on Improved Yolov8n-Pose-LBD. *Comput. Electron. Agric.* **2024**, *227*, 109551. <https://doi.org/10.1016/j.compag.2024.109551>.
29. Wang, X.F.; Wen, X.; Li, Y.; Du, C.F.; Zhang, D.K.; Sun, C.X.; Chen, B.H. A Precise Detection Method for Tomato Fruit Ripeness and Picking Points in Complex Environments. *Horticulturae* **2025**, *11*, 585. <https://doi.org/10.3390/horticulturae11060585>.
30. Ma, Z.H.; Dong, N.S.; Gu, J.Y.; Cheng, H.C.; Meng, Z.C.; Du, X.Q. STRAW-YOLO: A Detection Method for Strawberry Fruits Targets and Key Points. *Comput. Electron. Agric.* **2025**, *230*, 109853. <https://doi.org/10.1016/j.compag.2024.109853>.
31. Qian, L.W.; Yu, G.Y.; Liu, H.C.; He, Z.J. Segmentation and Calculation of Splashes Area during Fish Feeding Using Deep Learning and Linear Regression. *Comput. Electron. Agric.* **2025**, *234*, 110208. <https://doi.org/10.1016/j.compag.2025.110208>.
32. Wang, X.F.; Yang, W.J.; Qi, W.X.; Wang, Y.; Ma, X.J.; Wang, W. STaRNet: A Spatio-temporal and Riemannian Network for High-performance Motor Imagery Decoding. *Neural Netw.* **2024**, *178*, 106471. <https://doi.org/10.1016/j.neunet.2024.106471>.
33. Zhong, X.Y.; Liu, C. Toward Mitigating Architecture Overfitting on Distilled Datasets. *IEEE Trans. Neural Netw. Learn. Syst.* **2025**, 1–13. <https://doi.org/10.1109/TNNLS.2024.3525062>.
34. Park, S.; Yeo, Y.J.; Shin, Y.G. PConv: Simple yet Effective Convolutional Layer for Generative Adversarial Network. *Neural Comput. Appl.* **2022**, *34*, 7113–7124. <https://doi.org/10.1007/s00521-021-06846-2>.

-
35. Liu, G.X.; Zhang, Y.H.; Liu, J.; Liu, D.Y.; Chen, C.L.; Li, Y.J.; Zhang, X.J.; Touko Mbouembe, P.L. An improved YOLOv7 Model Based on Swin Transformer and Trident Pyramid Networks for Accurate Tomato Detection. *Front. Plant Sci.* **2024**, *15*, 1452821. <https://doi.org/10.3389/fpls.2024.1452821>.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.