agronomy

MDPI

Article

OccPepSeg-YOLO for Instance Segmentation of Occluded Peppers in Field Images

Xinran Yu [1,2], Mingxi Jiang [3], Fei Gao [4], Yize Fan [4], Yanyan Bai [4], Zhigang Peng [4], Qi Lu [4], Qian Liu [5] and Shengyong Xu [1,2,*]

[1] College of Engineering, Huazhong Agricultural University, Wuhan 430070, China; yuxinran_edu@163.com
[2] Key Laboratory of Agricultural Equipment for the Middle and Lower Reaches of the Yangtze River, Ministry of Agriculture, Huazhong Agricultural University, Wuhan 430070, China
[3] Zhejiang University-University of Illinois Urbana-Champaign Institute, International Campus, Zhejiang University, 718 East Haizhou Road, Haining 314400, China; jiangmingxi_edu@163.com
[4] Agriculture, Animal Husbandry and Water Conservancy Bureau of Otog Front Banner, Otog Front Banner, Ordos 016200, China; gaofei_edu@163.com (F.G.); fanyize_edu@163.com (Y.F.); byy_edu0723@163.com (Y.B.); pzhigang_edu@163.com (Z.P.); lqydoa@163.com (Q.L.)
[5] Wuhan Second Marine Design and Research Institute, Wuhan 430205, China; qianl_edu@163.com
* Correspondence: xsy@mail.hzau.edu.cn

Abstract

Agricultural operations such as pepper harvesting, fruit counting, and field phenotyping rely on accurate visual recognition and instance segmentation algorithms. However, pepper fruits in complex field environments often exhibit slender and curved shapes, partial occlusion, ambiguous boundaries, and adhesion between adjacent instances. Existing object detection and instance segmentation methods therefore struggle to obtain complete fruit masks, which adversely affects subsequent fruit counting, contour measurement, and picking-point localization. To improve the instance segmentation accuracy of occluded peppers in complex field scenes, this study proposes OccPepSeg-YOLO, an improved model based on YOLO11n-seg. First, a P2FreqFusion module is introduced to fuse shallow, high-resolution detail features with deep semantic features, thereby enhancing the representation of fruit edges and tip regions. Second, an ASC module is designed to model the directional and scale-related morphological characteristics of pepper fruits, while a BoundaryGate module strengthens responses at occlusion interfaces and boundaries between adjacent instances. Finally, an OccPepSegment multi-scale prototype segmentation head is constructed, and a BDoU loss function is introduced to improve the boundary consistency of instance masks. Experiments on a self-constructed field-pepper instance segmentation dataset showed that OccPepSeg-YOLO achieved M-P, M-R, M-mAP50, and M-mAP50–95 values of 93.87%, 92.09%, 97.17%, and 82.31%, respectively, representing improvements of 5.59, 3.18, 3.83, and 9.52 percentage points over YOLO11n-seg. Further comparisons with representative YOLO-based instance segmentation models, including YOLOv8n-seg, YOLOv9c-seg, YOLO12n-seg, and YOLOv26n-seg, demonstrated that OccPepSeg-YOLO achieved the best overall segmentation performance. In particular, its M-mAP50–95 exceeded the best competing result obtained by YOLOv9c-seg by 8.35 percentage points. Under a unified repeated-inference protocol on an RTX 3090 GPU using FP32 precision, a batch size of 1, and 640 × 640 inputs, OccPepSeg-YOLO achieved a mean inference latency of 15.801 ± 1.238 ms, a P95 latency of 17.323 ms, and a throughput of 63.29 FPS. These results demonstrate that the proposed model can produce more complete pepper instance masks under leaf occlusion, fruit overlap, and complex background conditions, providing technical support for field-pepper recognition, fruit counting, and visual perception by agricultural robots.

Academic Editor: Jayme Garcia Arnal Barbedo

Received: 21 July 2026
Revised: 13 August 2026
Accepted: 22 August 2026
Published: 27 August 2026

Keywords: field-grown peppers; instance segmentation; occluded pepper segmentation; YOLO11n-seg; agricultural machine vision; boundary awareness

1. Introduction

Pepper (*Capsicum annuum* L.) is an important vegetable and cash crop with considerable value for fresh consumption, food processing, and condiment production [1]. In open-field pepper production, mature-fruit harvesting, field fruit counting, yield estimation, and plant phenotyping are still performed mainly by manual labor, resulting in high labor intensity, low operational efficiency, limited capacity for continuous operation, and substantial subjective error. As cultivation expands and agricultural labor costs rise, agricultural robots and intelligent monitoring systems must reliably acquire information on fruit position, number, and morphology from field images to support automated harvesting, yield assessment, and precision management. In real open-field production scenes, however, pepper fruits rarely appear as isolated, regularly shaped targets with clear boundaries. They are commonly surrounded by leaves, branches, soil, shadows, and neighboring fruits, and they exhibit slender and curved shapes, pronounced scale variation, partial occlusion, fruit overlap, and boundary adhesion. When mature fruits are densely distributed or heavily occluded by leaves, the visible region of an individual pepper instance is often discontinuous, irregular, and poorly defined. Consequently, visual perception of occluded field-grown peppers requires more than simply detecting the presence of peppers or estimating their approximate locations. Each fruit's visible region, slender contour, and occlusion boundary must be accurately separated from complex backgrounds to support fruit counting, contour measurement, picking-point localization, and vision-guided agricultural robotic operations.

In recent years, deep learning-based object detection methods have been widely applied to pepper recognition and the localization of agricultural operations. Duan et al. developed an improved YOLOv8-based model for Chaotian pepper detection and demonstrated the feasibility of YOLO-series models for recognizing occluded pepper fruits in complex field environments [2]. Chen et al. proposed YOLO-Chili for pepper fruit detection and picking-point localization in complex environments, improving its applicability to harvesting tasks [3]. Huang et al. developed Pepper-YOLO to enhance green pepper detection and picking-point localization in natural scenes [4]. Nan et al. compressed YOLOv5l using an NSGA-II-based pruning strategy, reducing model complexity while maintaining real-time green pepper detection performance in the field [5]. Beyond pepper-related tasks, an improved YOLOv8 architecture has been used to localize mango picking points; combining object detection with instance segmentation improved the reliability of robotic picking decisions [6]. Research on apple picking-point localization showed that target regions obtained by semantic segmentation can provide more precise spatial constraints for picking-point calculation [7]. A tea-bud segmentation, detection, and picking-point localization model improved the consistency of tender-bud recognition and operation-point localization for selective harvesting [8]. Similarly, a method combining an improved YOLOv8 model with a depth camera improved the reliability of tomato-stem picking-point localization in complex fruit-cluster scenes [9]. These studies show that detection and localization models can provide target positions and operational references for agricultural tasks. Nevertheless, object detection models normally output bounding boxes, which describe only the enclosing extent of a target and cannot accurately represent the fruit's true visible contour. For slender, curved, and mutually occluded field-grown peppers, bounding boxes tend to include pixels from leaves, branches, the background, and adjacent

fruits, making them inadequate for pixel-level tasks such as instance counting, contour measurement, discrimination of occlusion interfaces, and precise picking-point localization.

To obtain more detailed representations of target regions, instance segmentation and semantic segmentation have gradually been introduced into agricultural visual perception. Cong et al. used an improved Mask R-CNN to segment greenhouse sweet pepper fruits, improving fruit-region extraction in complex greenhouse environments [10]. Paul et al. applied YOLO-series methods to sweet pepper detection, segmentation, counting, and mobile recognition, establishing a multitask vision pipeline for harvesting scenarios [11]. Li et al. combined an improved YOLOv7 model with RGB-D sensing to recognize strawberries and estimate their spatial positions, improving a harvesting robot's ability to acquire three-dimensional target coordinates [12]. Jia et al. proposed Polar-Net for green-fruit instance segmentation in complex orchard environments, enhancing the separation of fruits from leaves under similar-color backgrounds [13]. Peng et al. developed ResDense-Focal-DeepLabV3+ for litchi-branch segmentation, improving the representation of branch edges and elongated structures [14]. Xie et al. used an improved DeepLabv3+ model to segment litchi branches in complex backgrounds and further enhanced edge extraction [15]. Huang et al. employed an improved Mask R-CNN for grape-cluster detection and instance segmentation in orchards, improving target localization and mask prediction in natural scenes [16]. These studies demonstrate that pixel-level segmentation constrains target regions more accurately than bounding boxes and provides refined visual input for robotic harvesting, target counting, and complex-scene perception. However, most existing agricultural segmentation studies focus on relatively regular fruits, clustered targets, branches, or buds, whose morphology and occlusion patterns differ substantially from those of field-grown peppers. Field-grown peppers have more pronounced elongated major axes, curved postures, narrow tip boundaries, and irregular visible regions. Under leaf occlusion, fruit overlap, and similar-color background interference, they are particularly susceptible to contour discontinuity, mask adhesion, and boundary displacement. Therefore, instance segmentation of occluded peppers is not merely a general fruit-region extraction problem; it also requires multi-scale instance recognition, preservation of locally visible regions, separation of occlusion boundaries, and fine contour representation for subsequent operation-point localization.

To address the segmentation requirements arising from the morphological characteristics and occlusion relationships of field-grown peppers, agricultural vision research has gradually progressed from simple target recognition toward more refined tasks involving multi-scale structural representation, occluded-region separation, and extraction of downstream operational information. An improved YOLOv8 method was applied to multi-scale, multi-target, and three-dimensional position detection of flowering Chinese cabbage, improving the model's adaptability to complex crop structures [17]. A weed apical-meristem localization method based on YOLO instance segmentation and connected-component analysis demonstrated that pixel-level instance masks can provide a more accurate contour basis for locating small growth points [18]. A ripe-tomato picking-point recognition method combining semantic segmentation with morphological processing improved the stability of picking-point extraction under occlusion [19], while YOLO-CornSeg, through corn-seedling segmentation and indirect weed detection, showed that high-quality crop masks can improve the reliability of downstream agricultural inference tasks [20]. In addition, studies on safflower, tea, and citrus further used segmentation results for picking-point or operation-point localization [21–23], and real-time strawberry detection and instance segmentation in unstructured environments improved mask prediction quality for occluded fruits in natural harvesting scenes [24]. More recently, YOLO11 has been increasingly applied to a range of agricultural vision tasks, including field cotton topping target segmentation [25],

cross-domain adaptation for tomato phenotyping [26], multi-crop leaf detection and segmentation [27], apple target-region segmentation and pose estimation [28], and blueberry instance segmentation in complex greenhouse environments [29]. These studies indicate that agricultural visual perception is evolving beyond determining whether a target can be detected toward more detailed representation of target morphology, local structures, occlusion relationships, and operational information. They also demonstrate the strong potential of newer frameworks such as YOLO11 for adaptation to diverse agricultural tasks. For field-grown peppers, such refined perception requires not only identifying the main fruit region, but also preserving narrow tips and local boundary details, maintaining the structural continuity of slender and curved fruits, and accurately distinguishing occlusion interfaces and adjacent adhered instances.

In response to these model capability requirements, previous studies have provided methodological references for instance segmentation in complex scenes from the perspectives of occluded-object segmentation, mask-quality assessment, multi-scale feature representation, spatial-context modeling, and prototype-mask generation. A comparison between YOLOv8 and Mask R-CNN for instance segmentation in complex orchard environments showed that one-stage and two-stage models exhibit different characteristics in terms of segmentation accuracy, inference efficiency, and adaptability to complex scenes [30]. MAE-YOLOv8 improved small green-plum detection in complex orchards under occlusion and background interference [31]; a detection and segmentation model for obscured green fruits further improved the accuracy of occluded-fruit region extraction in complex natural environments [32]; and an improved Mask Scoring R-CNN enhanced mask scoring and segmentation quality for apple detection and instance segmentation in natural environments [33]. In more general vision research, Frequency-Aware Feature Fusion strengthens the joint representation of high-frequency details and low-frequency semantic information in dense prediction tasks through frequency-aware fusion [34]; the Large Selective Kernel Network improves spatial feature modeling for large-scale context and multi-scale targets [35]; and YOLACT++ achieves a favorable balance between mask-generation efficiency and prediction accuracy through one-stage prototype-mask prediction [36]. Collectively, these studies provide valuable insights into difficult-target recognition, occluded-region extraction, mask-quality optimization, high- and low-frequency feature fusion, large-range contextual representation, and prototype-mask generation, thereby establishing a methodological basis for improving the completeness and boundary quality of instance masks in complex field environments.

Despite the methodological foundation established by previous studies for refined visual perception in complex agricultural scenes, directly applying existing methods to images of occluded field-grown peppers still presents several coupled, task-specific challenges. First, pepper tips and narrow boundary regions account for only a small proportion of the target pixels, and repeated downsampling can easily weaken shallow high-resolution details, resulting in incomplete edge predictions. Second, peppers exhibit pronounced elongated major-axis structures and curved postures, while conventional convolutional features are limited in modeling directional variations and major-axis continuity, making structural breakage more likely under local occlusion and complex background interference. Third, leaf occlusion, fruit overlap, and similar-color backgrounds can weaken boundary responses at interfaces between adjacent instances, leading to mask adhesion, contour displacement, or local omissions. Fourth, existing segmentation heads generally emphasize coverage of the main target region and have difficulty simultaneously preserving multi-scale mask generation, boundary refinement, and continuity of elongated structures. In addition, the commonly used BCE mask loss mainly constrains predictions from a pixel-classification perspective and provides insufficient supervision for boundary errors in small

but critical regions such as fruit tips, narrow edges, and occlusion interfaces. Therefore, instance segmentation of occluded field-grown peppers requires more than a local improvement targeting a single issue; it calls for coordinated modeling of high-resolution detail preservation, elongated-shape continuity, occlusion-boundary discrimination, and multi-scale mask representation, while maintaining acceptable inference efficiency.

To address these challenges, this study constructs a dataset for instance segmentation of occluded peppers in complex field environments and proposes OccPepSeg-YOLO, an improved model based on YOLO11n-seg. The main contributions are as follows:

(1) An instance segmentation dataset of occluded peppers in complex field environments was constructed. The dataset covers representative field conditions involving different illumination levels, fruit postures, leaf occlusion, fruit overlap, and background interference, and provides a data foundation for occluded-pepper segmentation, fruit counting, and phenotypic analysis.

(2) OccPepSeg-YOLO, an occluded-pepper instance segmentation network based on YOLO11n-seg, was developed. To accommodate the slender and curved shapes, pronounced scale variation, complex occlusion interfaces, and adhesion between neighboring instances of field-grown peppers, the network incorporates P2FreqFusion, ASC, BoundaryGate, and OccPepSegment structures. These components enhance the fusion of shallow details and deep semantics, representation of elongated fruit morphology, boundary-region responses, and multi-scale mask generation, thereby improving segmentation accuracy in complex field scenes.

(3) A BDoU loss function was introduced to strengthen supervision of mask-boundary errors during training. Compared with BCE loss, which constrains predicted masks mainly through pixel classification, BDoU places greater emphasis on small but critical boundary regions, including fruit tips, narrow edges, and occlusion interfaces, helping to reduce adhesion between adjacent instances and mask-boundary displacement.

(4) The effectiveness of the proposed method was verified through model comparisons, ablation experiments, complexity analysis, and field-recognition visualization. The results show that OccPepSeg-YOLO improves the completeness and boundary accuracy of pepper instance masks under complex backgrounds, leaf occlusion, fruit overlap, and instance adhesion, providing technical support for field-pepper counting, phenotypic measurement, and vision-guided agricultural operations.

2. Materials and Methods

2.1. Overall System Workflow

As shown in Figure 1, the overall workflow of this study comprises four stages: image acquisition, dataset construction, model development, and model training and performance evaluation. (1) Image acquisition: RGB images of pepper plants were captured with an ORBBEC Gemini 335L camera in natural field environments, covering variations in illumination, scale, posture, occlusion severity, and background interference. (2) Dataset construction: the acquired images were screened, cropped, and annotated with instance-level polygons. Each pepper fruit was treated as an independent instance, and the images were divided into training, validation, and test sets; only the training set was further augmented. (3) Model development: YOLO11n-seg was used as the baseline, and OccPepSeg-YOLO was designed to address the slender and curved morphology, local occlusion, ambiguous boundaries, and adhesion between neighboring pepper instances. P2FreqFusion, ASC, BoundaryGate, OccPepSegment, and BDoU loss were incorporated to enhance boundary-detail representation, elongated-shape modeling, occlusion-interface perception, and multi-scale mask generation. (4) Model training and performance evaluation: all models were trained under a unified experimental environment and parameter configuration and were

evaluated in terms of segmentation accuracy, detection performance, and model complexity. Comparative experiments, ablation studies, complexity analysis, and field-recognition experiments were conducted to validate the proposed method.

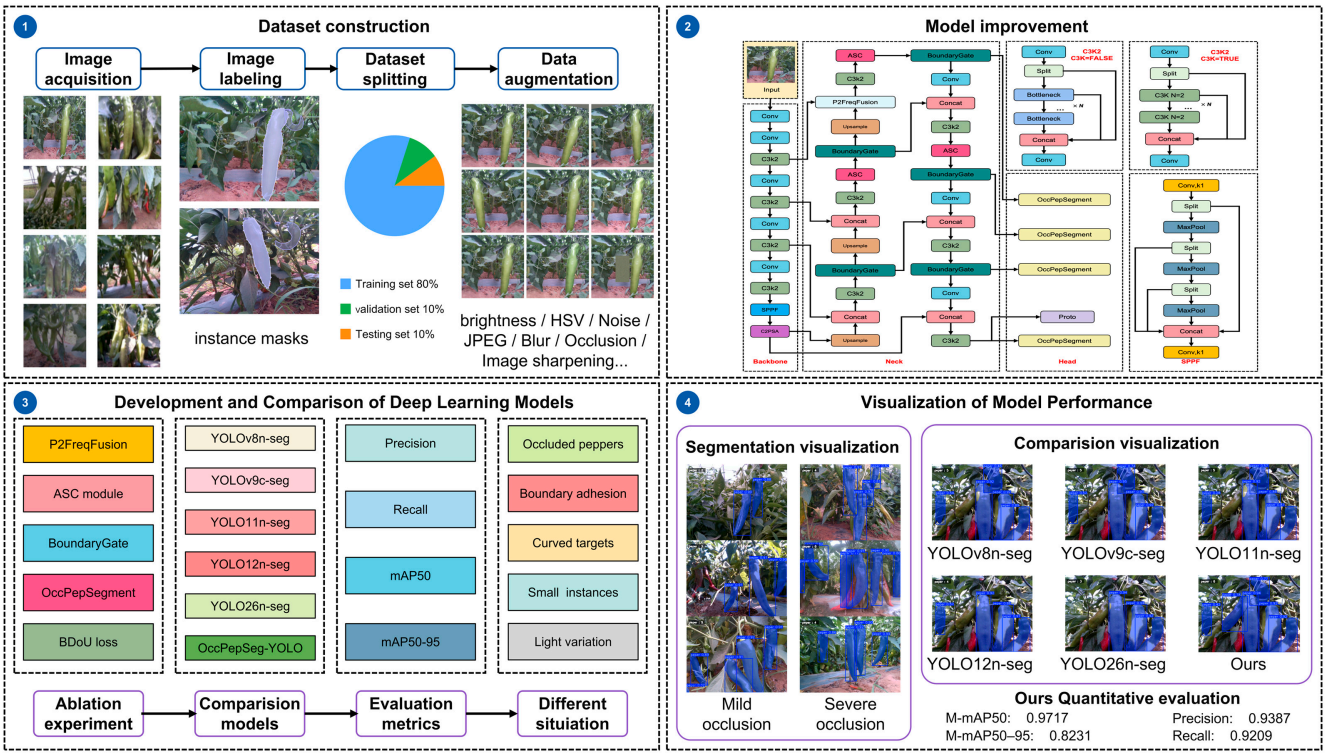

Figure 1. Overall workflow of OccPepSeg-YOLO for instance segmentation of occluded peppers in field environments.

2.2. Dataset Image Collection

Experimental images were collected from June to July 2025 at a pepper-growing base in Otog Front Banner, Inner Mongolia Autonomous Region, China. The site was an open-field pepper production area, and the subjects were naturally growing pepper plants and fruits. Images were captured using an ORBBEC Gemini 335L camera (Orbbec Inc., Shenzhen, China); only the RGB data were used in this study. During acquisition, the camera was connected to a laptop, and the shooting distance and viewing angle were adjusted according to plant height, fruit distribution, and inter-row space to ensure that the fruits and their surrounding leaves, branches, and background were fully recorded.

The camera white balance was set to automatic, autofocus was enabled, and the image resolution was set to 1280 × 800 pixels. The camera was positioned approximately 0.5–0.8 m above the ground to obtain close-range field views. To increase the dataset's coverage of complex field conditions, the acquisition process retained representative scenes involving no occlusion, slight occlusion, moderate occlusion, severe occlusion, dusty fruit surfaces, diseased fruit surfaces, underexposure, and overexposure. After multiple rounds of collection and screening, 1500 original RGB images of field-grown peppers were retained, as illustrated in the image-acquisition component of Figure 1.

2.3. Dataset Preprocessing

2.3.1. Image Annotation

X-AnyLabeling (version 4.0.0-beta.7) was used to perform instance-level polygon annotation of the pepper fruits in each image. Each individual pepper fruit was treated as one instance. Polygon vertices were placed sequentially along the visible fruit contour

and any continuous boundary that could be inferred reliably, and the enclosed polygon was used as the ground-truth instance mask. All instances were assigned the class label pepper. After annotation, each annotation file recorded the class name and polygon-vertex coordinates of every pepper instance in a coordinate system whose origin was at the upper-left corner of the image. The polygon coordinates were then normalized by image width and height to values between 0 and 1 and converted to the label format required for YOLO instance segmentation. The converted labels were saved as .txt files, with each row representing one pepper instance and containing the class index followed by the normalized polygon-vertex coordinates for model training.

2.3.2. Dataset Partitioning

To prevent data leakage caused by augmented samples appearing across different subsets, the original data were split before augmentation. After image screening, cropping, and instance-level annotation, the images were divided at the image level into training, validation, and test sets. The training set was used to learn model parameters, the validation set was used to monitor training and select the best model, and the test set was reserved for independent evaluation after training. The validation and test sets contained only authentic field images and no offline-augmented samples, ensuring that the evaluation more objectively reflected generalization to natural field environments. The validation set ultimately contained 150 images and 528 pepper instances, whereas the test set contained 150 images and 501 instances.

2.3.3. Dataset Augmentation

To improve model robustness to illumination variation, noise, local occlusion, and image degradation, the augmentation methods were selected according to the major imaging variations and degradation patterns observed in the original field data. Two main principles guided the selection. First, the transformations should not alter the class identity of the pepper instances or invalidate the original annotations. Second, the simulated variations should correspond to realistic disturbances that may occur during field image acquisition. Based on these principles, brightness adjustment, HSV perturbation, and gamma transformation were used to represent image variations caused by different illumination and exposure conditions; Gaussian noise, salt-and-pepper noise, and JPEG compression were used to simulate sensor noise and image-quality degradation that may arise during image storage or transmission; Gaussian blur and motion blur were used to mimic image degradation caused by defocus, camera motion, or plant movement; random occlusion was introduced to increase the diversity of training samples under partial-occlusion conditions; and image sharpening was used to introduce variations in local edge contrast and image sharpness. The adopted augmentation methods are illustrated in the data-augmentation component of Figure 1. All offline augmentations were applied only to the training set after the original images had been divided at the image level into training, validation, and test subsets. The validation and test sets therefore consisted exclusively of authentic field images without offline augmentation, preventing augmented samples from appearing across different subsets and thereby avoiding data leakage and ensuring independent performance evaluation. After augmentation, the training set contained 7200 images and 25,380 pepper instances, while the validation and test sets contained 150 images with 528 instances and 150 images with 501 instances, respectively. The final dataset therefore comprised 7500 images and 26,409 pepper instances.

2.4. OccPepSeg-YOLO Model for Segmentation of Occluded Peppers

2.4.1. Overall Architecture of OccPepSeg-YOLO

Because field-pepper instance segmentation requires both high mask accuracy and practical inference efficiency, YOLO11n-seg was selected as the baseline model. YOLO11n-seg is a lightweight instance segmentation model in the Ultralytics YOLO family that simultaneously predicts bounding boxes, class confidence scores, and instance masks within a one-stage framework [25]. Compared with larger segmentation variants in the same family, YOLO11n-seg has fewer parameters and lower computational cost, making it more suitable for resource-constrained agricultural vision applications. Nevertheless, instance segmentation of occluded field-grown peppers imposes greater demands on spatial-detail representation and boundary discrimination. Pepper fruits are typically slender and curved, and their tips and narrow boundaries occupy relatively few pixels, making these details vulnerable to repeated downsampling. Leaf occlusion, fruit overlap, and similar-color backgrounds also weaken boundary responses between neighboring instances, preventing the original segmentation head from producing stable and complete fruit masks. To address these problems, OccPepSeg-YOLO was developed on the basis of YOLO11n-seg, as shown in Figure 2. The lightweight backbone of the baseline network was retained to preserve efficient feature extraction and deployment advantages, while the neck and segmentation head were modified specifically to improve boundary details, elongated-shape representation, inter-instance separation, and mask expression for occluded peppers.

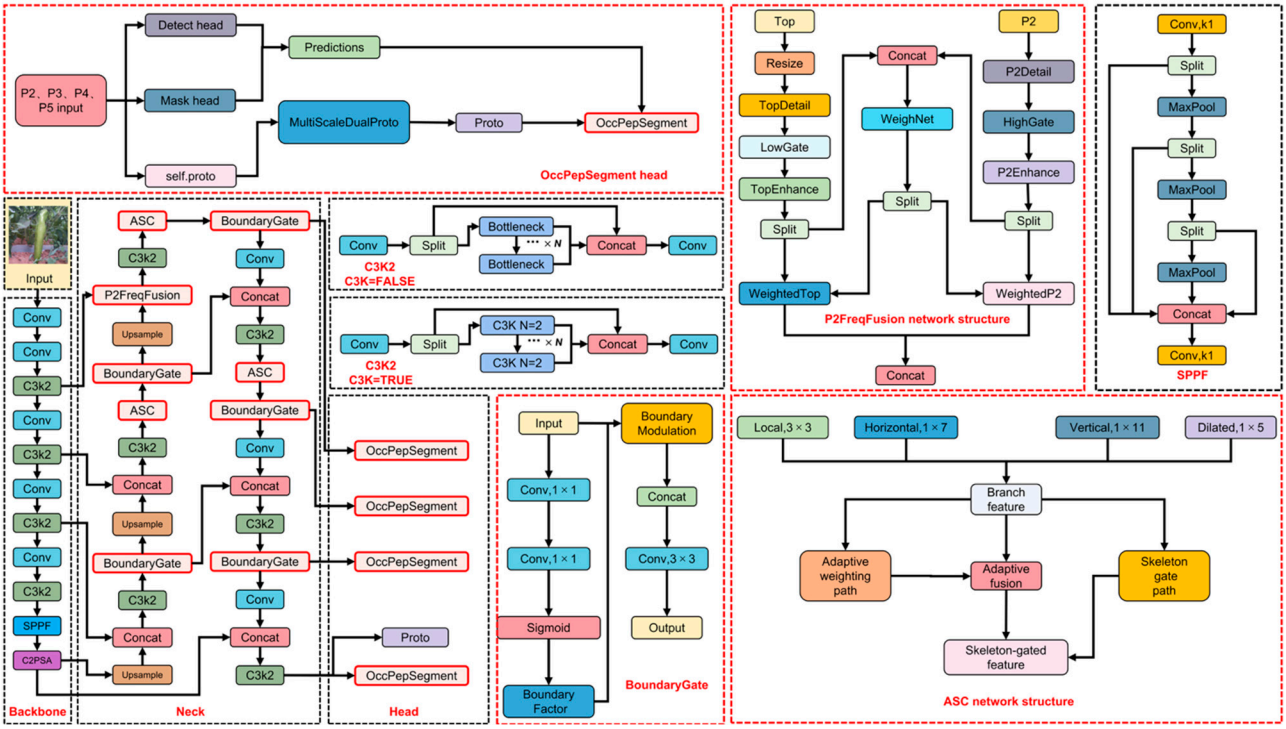

Figure 2. Overall architecture of OccPepSeg-YOLO and detailed structures of the proposed modules. P2FreqFusion, ASC, BoundaryGate, and OccPepSegment introduced into the Neck and Head are highlighted with red solid outlines in the main network. Their detailed architectures are enclosed by red dashed boxes, while black dashed boxes indicate baseline or supporting components retained from YOLO11n-seg.

2.4.2. P2FreqFusion Module

During feature fusion, YOLO11n-seg relies mainly on P3 and deeper features for mask prediction and makes limited use of high-resolution spatial details. For fruit tips, narrow boundaries, and locally occluded regions in field-pepper images, this feature representation

can weaken important details. Frequency-aware feature fusion improves cross-scale fusion for dense prediction by coordinating high-frequency boundary information with low-frequency semantic information [34]. To strengthen the complementarity between shallow spatial information and deep semantic information, a P2FreqFusion frequency-aware high-resolution fusion module was designed, as illustrated in Figure 3.

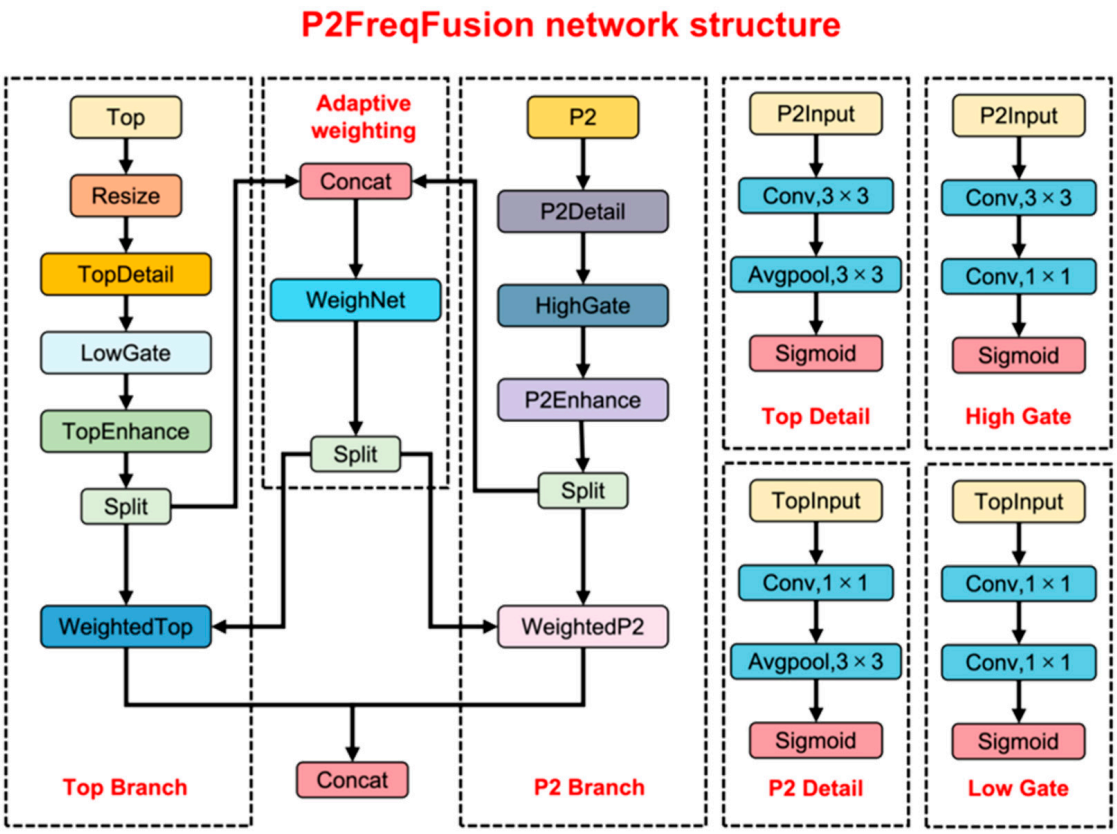

Figure 3. Network architecture of the P2FreqFusion module.

Let F_t denote the upsampled feature from the deep pathway and F_l denote the shallow P2 feature supplied by the lateral connection from the backbone. P2FreqFusion first performs frequency-aware enhancement on the two feature types:

$$\hat{F}_t = LP(F_t) + G_t(F_t)\cdot(F_t - LP(F_t)) \quad (1)$$

$$\hat{F}_l = F_l + G_l(F_l)\cdot(F_l - LP(F_l)) \quad (2)$$

where $LP(\cdot)$ denotes low-pass smoothing, and $G_t(\cdot)$ and $G_l(\cdot)$ are sigmoid-activated gating functions used to regulate the deep semantic feature and the shallow detail feature, respectively. The module then generates pixel-wise fusion weights from the enhanced features and produces the fused output:

$$F_{out} = Concat\left(w_t\hat{F}_t,\ w_l\hat{F}_l\right) \quad (3)$$

where w_t and w_l are the adaptive fusion weights for the deep and shallow features, respectively. In the implementation, a learnable detail-enhancement coefficient is also introduced to regulate the contribution of the shallow boundary branch. Through this process, P2FreqFusion selectively exploits shallow high-resolution information in complex backgrounds, reduces interference from leaf texture and background noise, and improves the spatial resolution of boundary regions for occluded peppers.

2.4.3. ASC Module

Pepper fruits exhibit a pronounced major-axis structure and substantial directional variation, whereas conventional square convolutions are limited in representing the morphological continuity of such slender and curved targets. Adaptive selection of different receptive fields has been used to enhance target-shape and contextual modeling [35]. Simply enlarging the convolution kernel, however, increases computation and introduces additional background responses. To improve representation of the directional and scale-related morphology of pepper fruits, an ASC direction- and scale-aware morphological convolution module was designed, as shown in Figure 4.

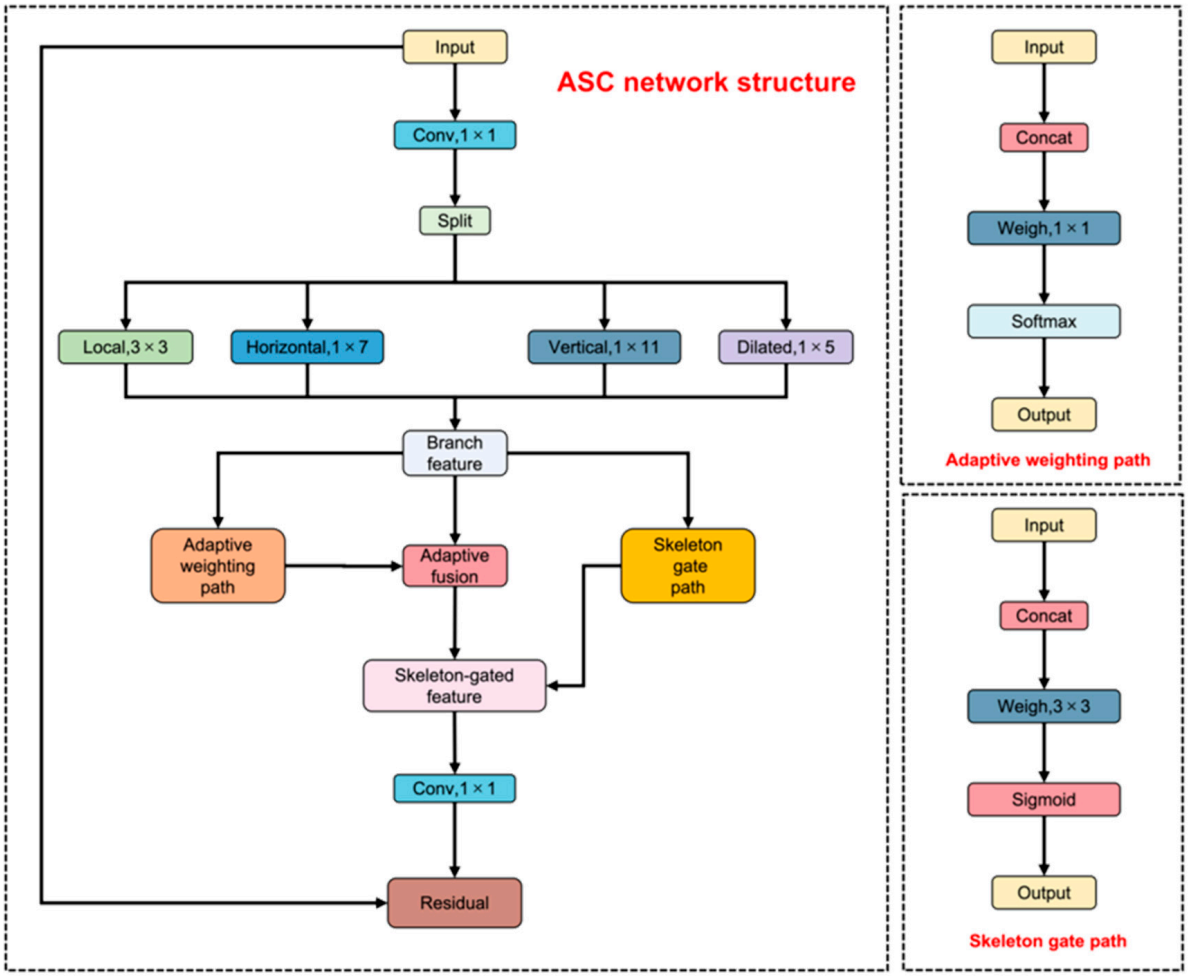

Figure 4. Network architecture of the ASC module.

Given an input feature X, ASC first performs channel compression using a 1×1 convolution and then constructs four convolution branches: local, horizontal, vertical, and dilated major-axis branches. The output of the kth branch is expressed as

$$F_k = B_k(Conv_{1\times1}(X)), \quad k = 1,2,3,4 \tag{4}$$

where $B_k(\cdot)$ represents a convolution branch with a specific direction or scale. The outputs of the four branches are adaptively fused using pixel-wise attention weights:

$$F_{out} = Conv_{1\times1}\left(\sum_{k=1}^{4} \alpha_k F_k\right) + S(X) \tag{5}$$

where α_k is the fusion weight of the kth branch and $S(X)$ is the shortcut branch. The local branch extracts short-range edge and texture information; the horizontal and vertical branches strengthen major-axis structures in different directions; and the dilated major-axis branch enlarges the receptive field while preserving long-range morphological continuity. In the implementation, a skeleton-guided gate is added after branch fusion to strengthen responses in the main regions of slender fruits. The ASC module therefore represents directional changes and major-axis continuity more effectively and reduces the risk of local structural breakage caused by occlusion or background interference.

2.4.4. BoundaryGate Module

In instance segmentation, boundary regions usually account for only a small fraction of target pixels, yet they are essential for preserving instance contours and separating adjacent targets. In field-pepper images, fruit edges are often close to leaves, branches, or neighboring fruits. Conventional feature-fusion operations tend to emphasize the main target region and respond less strongly to edges and contact regions between instances. To enhance sensitivity to boundary-related features, a BoundaryGate boundary-gating module was designed, as illustrated in Figure 5.

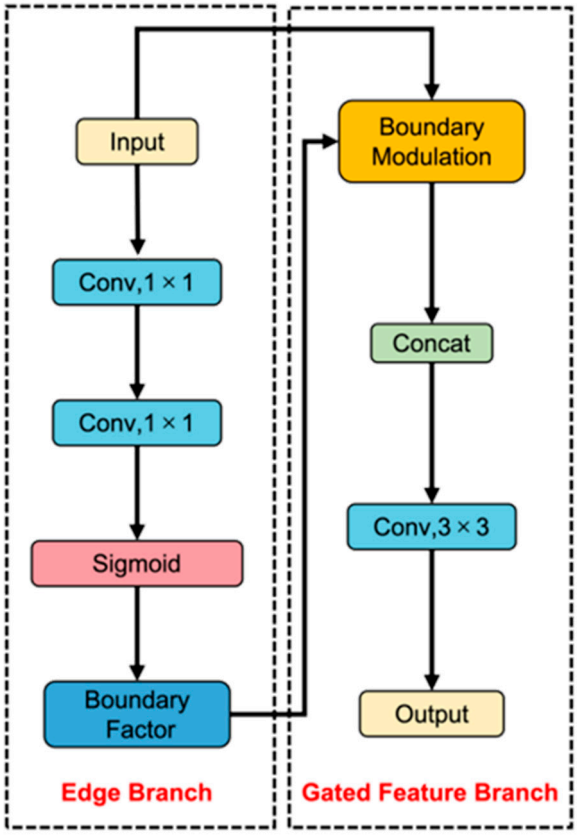

Figure 5. Network architecture of the BoundaryGate module.

The central idea of BoundaryGate is to adaptively generate a boundary-likelihood map from the input feature and use this map to gate and enhance the original feature. Given an input feature X, BoundaryGate first generates a single-channel boundary-likelihood map through convolution:

$$E = \sigma(G_e(X)) \quad (6)$$

where $G_e(\cdot)$ denotes the boundary-response generation function, σ denotes the sigmoid activation, and E represents the likelihood that a given location belongs to a fruit edge or an interface between instances. The boundary-likelihood map is then used as a spatial gating

factor for the input feature, strengthening boundary-prone regions while preserving the representation of the main target region:

$$F_{out} = G_r(X \cdot (1 + \beta E)) \quad (7)$$

where β is a learnable boundary-enhancement coefficient and $G_r(\cdot)$ is the feature-refinement function. This module requires no additional boundary annotation and learns boundary-related responses directly from the instance-segmentation supervision. Compared with simply adding convolutional layers, BoundaryGate concentrates enhancement on likely boundary regions, thereby improving feature discrimination at fruit edges and inter-instance interfaces with only a small additional computational cost.

2.4.5. OccPepSegment Head

YOLO-series instance segmentation models generally generate instance masks by linearly combining shared prototype masks with instance-specific coefficients [36]. Although this mechanism is computationally efficient, the original segmentation head provides limited prototype diversity and has difficulty simultaneously preserving region completeness, boundary alignment, and continuity of elongated structures. To improve mask representation for occluded peppers, an OccPepSegment multi-scale prototype segmentation head was constructed, as shown in Figure 6. Let *Pproto* denote the prototype masks and C_i denote the instance coefficients. The predicted mask of the ith target is expressed as

$$M_i = \sigma(C_i P_{proto}) \quad (8)$$

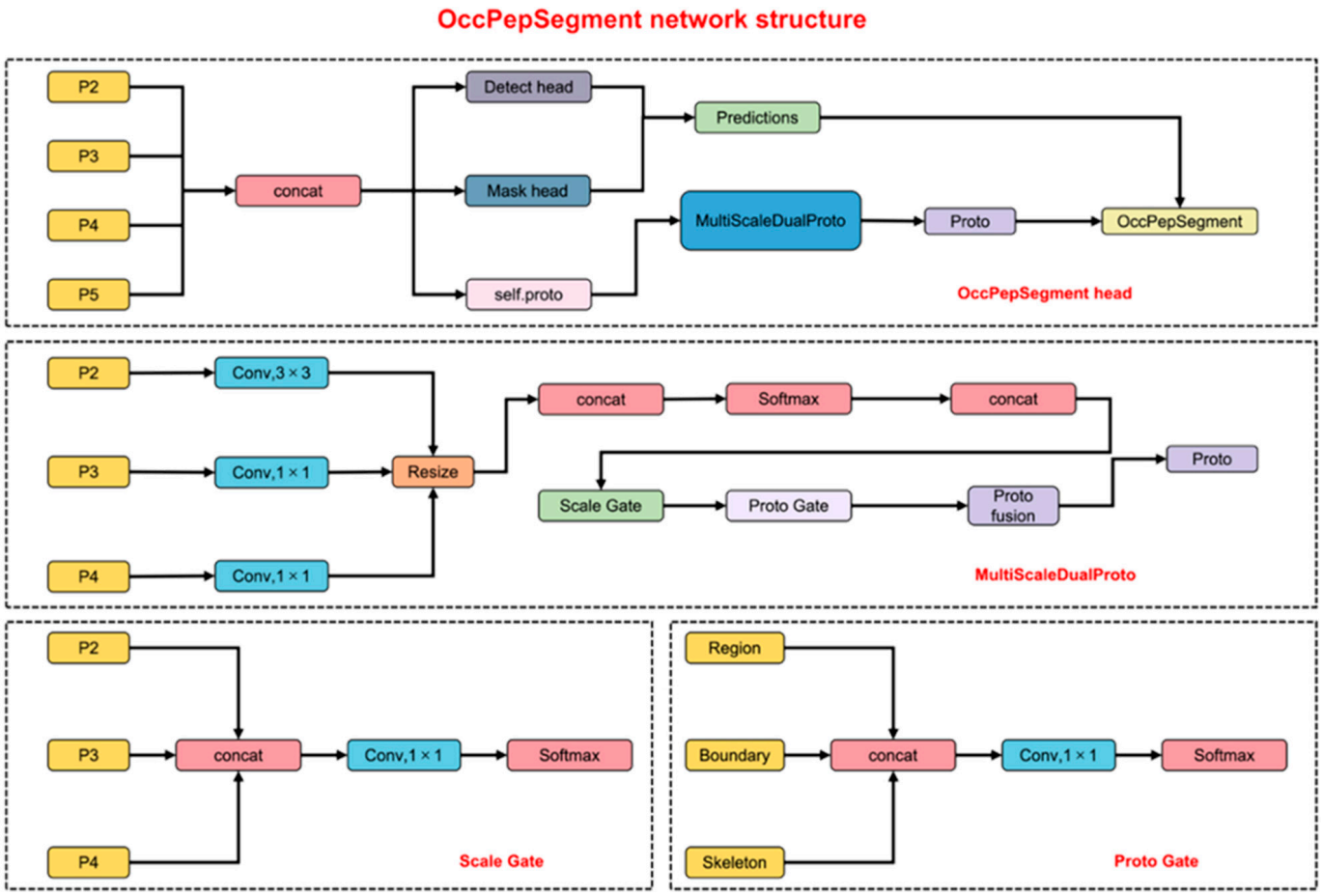

Figure 6. Network architecture of the OccPepSegment head.

The original Segment Head generates shared prototypes primarily from a single scale or a limited number of feature scales. Although efficient, this mechanism remains inadequate for occluded-pepper segmentation. A single-scale prototype cannot simultaneously preserve high-resolution boundaries and middle- to high-level semantic information. In addition, conventional prototypes mainly emphasize the main target region and do not explicitly represent boundary contours or the major-axis skeleton, which can lead to broken masks for slender fruits or fusion between neighboring instances.

To improve instance-mask quality for occluded peppers, the original Segment Head was replaced with OccPepSegment, which contains a MultiScaleDualProto multi-scale prototype-generation module, as shown in Figure 6. Let X_2, X_3, and X_4 denote the features at the P2, P3, and P4 scales, respectively. The module first fuses the features from the different scales to obtain a multi-scale prototype feature:

$$F_m = Fuse(X_2, Up(X_3), Up(X_4)) \tag{9}$$

MultiScaleDualProto then constructs three branches for region, boundary, and skeleton prototypes and generates the final prototype through adaptive gating:

$$P_{proto} = G_o(Concat(\delta_r R, \delta_b B, \delta_s S)) \tag{10}$$

where R, B, and S denote the region, boundary, and skeleton prototype features, respectively; δ_r, δ_b, and δ_s are their adaptive fusion weights; and $G_o(\cdot)$ is the prototype-output function. The region prototype describes the completeness of the main pepper region, the boundary prototype strengthens fruit edges, tips, and occlusion interfaces, and the skeleton prototype preserves the central structure and major-axis continuity of slender fruits. Accordingly, OccPepSegment generates instance masks using not only responses from the main region but also boundary and skeleton information, making it better suited to slender, occluded, and adherent pepper instances.

2.4.6. Improved Loss Function

The BCE mask loss commonly used for instance segmentation constrains agreement between predicted and ground-truth masks from a pixel-classification perspective and can be dominated by the large number of pixels in the main target region. For small but critical areas such as occlusion interfaces, fruit tips, and narrow boundaries, BCE alone may not adequately penalize boundary displacement. Such errors can truncate fruit tips, connect adjacent instances, or produce incomplete local contours. To strengthen boundary-error supervision during training, Boundary Difference over Union (BDoU) loss was introduced in addition to the original mask loss.

The basic BCE mask loss is expressed as

$$L_{BCE} = -\frac{1}{N}\sum_{i=1}^{N}[y_i log(\hat{y}_i) + (1 - y_i) log(1 - \hat{y}_i)] \tag{11}$$

where $\hat{y}_i$ is the predicted probability of the ith pixel, y_i is the corresponding ground-truth label, and N is the number of pixels. Local dilation and erosion are then applied to the predicted and ground-truth masks to extract boundary responses:

$$E(M) = Dilate(M) - Erode(M) \tag{12}$$

$$E(Y) = Dilate(Y) - Erode(Y) \tag{13}$$

A local boundary band is constructed from the predicted and ground-truth boundaries:

$$\Omega_b = Edge(max(E(M), E(Y))) \tag{14}$$

Within this boundary band, BDoU loss is defined as

$$L_{BDoU} = \frac{\sum_{\Omega b}|M_b - Y_b|}{\sum_{\Omega b} max(M_b, Y_b) + \sum_{\Omega b} min(M_b, Y_b) + 1} \tag{15}$$

where M_b and Y_b are the predicted and ground-truth masks within the boundary band, respectively, and the constant 1 is included for numerical stability. The final loss for the instance-mask branch is

$$L_{mask} = L_{BCE} + \lambda L_{BDoU} \tag{16}$$

In the implementation, λ was set to 0.35. This loss is applied only to the instance-mask branch; the bounding-box loss, classification loss, and distribution focal loss (DFL) retain the original settings of the YOLO framework. The proposed loss causes the model to focus on prediction errors within the boundary band while optimizing the main fruit region, thereby placing greater emphasis on fruit contours, occlusion interfaces, and adhesion regions during training. It complements BoundaryGate and the multi-scale boundary prototypes: the network modules strengthen boundary-feature representation, whereas the loss function increases the weight of boundary consistency in the optimization objective.

3. Results

3.1. Experimental Environment and Evaluation Metrics

Experiments were conducted using the Ultralytics deep learning framework. The computational platform was equipped with an NVIDIA GeForce RTX 3090 GPU with 24 GB of video memory and a rated power of 350 W. The software environment included NVIDIA Driver 570.124.04, Python 3.10.14, PyTorch 2.4.1+cu121, CUDA 12.1, and cuDNN 9.1.0 (version 90100). All models were trained and evaluated using the same field-pepper instance segmentation dataset and identical training, validation, and test splits, and were initialized with pretrained weights. To reduce the influence of random factors on model comparison, the random seed was fixed at 0 for all experiments. The validation set was used to monitor the training process and select the best-performing checkpoint, whereas the test set was reserved exclusively for final performance evaluation and was not involved in parameter selection during training.

To ensure a fair comparison among different network architectures, a unified basic training protocol was adopted for all comparative and ablation experiments, without model-specific hyperparameter retuning. The main training hyperparameters are summarized in Table 1. SGD was used as the optimizer, with an initial learning rate lr0 of 0.01, a final learning-rate factor lrf of 0.01, a momentum of 0.937, and a weight decay of 0.0005. The input size was set to 640 × 640 pixels as a practical compromise between preserving spatial details, particularly pepper tips and narrow boundaries, and computational cost. A batch size of 16 was adopted to accommodate the available memory of the RTX 3090 while maintaining stable training. The maximum number of training epochs was set to 2000, together with an early-stopping patience of 50 epochs. Training was terminated when the validation performance showed no further improvement for 50 consecutive epochs, and the checkpoint with the best validation performance was retained for subsequent evaluation. Therefore, 2000 epochs represents the upper limit of the training duration rather than a fixed number of epochs for every model. These basic training settings were kept unchanged across all comparative and ablation experiments to minimize the influence of differences in training configuration on the model performance comparison.

Table 1. Main training hyperparameters.

Hyperparameter	Value
epochs	2000
patience	50
batch_size	16
img_size	640
lr0	0.01
lrf	0.01
momentum	0.937
weight_decay	0.0005
optimizer	SGD

Model performance was evaluated at both the bounding-box and instance-mask levels. Bounding-box metrics included B-P, B-R, B-mAP50, and B-mAP50–95, whereas instance-mask metrics included M-P, M-R, M-mAP50, and M-mAP50–95. Here, P, R, mAP50, and mAP50–95 denote precision, recall, average precision at an IoU threshold of 0.5, and average precision averaged over IoU thresholds from 0.50 to 0.95 in increments of 0.05, respectively. Because this study involved a single pepper class, mAP was equivalent to AP for that class. Compared with mAP50, mAP50–95 imposes stricter overlap requirements between predicted and ground-truth regions and more effectively reflects mask quality at fruit boundaries, slender tips, and interfaces between adjacent instances. M-mAP50–95 was therefore used as the primary metric for evaluating instance segmentation of occluded peppers.

In addition to segmentation accuracy, the computational characteristics of the evaluated models were assessed from three perspectives: model size, theoretical computational complexity, and measured inference efficiency. Parameter count and weight-file size were used to characterize model scale, FLOPs were used to quantify theoretical computational complexity at a fixed input resolution, and practical inference efficiency was evaluated using mean latency, latency standard deviation, P95 latency, and frames per second (FPS). Because FLOPs reflect only the theoretical operation count, actual inference speed can also be affected by operator type, GPU parallelism, memory-access patterns, CUDA kernel scheduling, and post-processing overhead. Therefore, all models were repeatedly benchmarked under the same hardware and inference configuration. Specifically, all saved model checkpoints were evaluated on an NVIDIA GeForce RTX 3090 GPU using the CUDA backend and FP32 precision, with a batch size of 1 and an input size of 640 $\times$ 640 pixels. Each model first underwent 100 warm-up predictions to reduce the influence of CUDA context initialization, memory allocation, and operator caching on the timing measurements. This was followed by five independent repeated runs, each consisting of 200 timed predictions, yielding a total of 1000 timing measurements per model. The measured latency included image preprocessing, model forward inference, and post-processing/NMS, but excluded model loading, weight reading, and disk image decoding. The mean latency, standard deviation, and P95 latency were recorded, and FPS was calculated from the mean latency; the corresponding comparative results are presented in Section 3.6.

3.2. Model Training Results

OccPepSeg-YOLO was trained and validated on the self-constructed field-pepper instance segmentation dataset. During training, the bounding-box, segmentation, classification, and distribution focal losses gradually decreased and stabilized. The validation losses followed trends similar to those of the training losses, with no evident divergence between them, indicating stable optimization and convergence.

On the independent test set, OccPepSeg-YOLO achieved M-P, M-R, M-mAP50, and M-mAP50–95 values of 93.87%, 92.09%, 97.17%, and 82.31%, respectively. Among these

metrics, M-mAP50–95 evaluates the overlap between predicted and ground-truth masks over a range of strict IoU thresholds. Compared with M-mAP50, it imposes more stringent requirements on contour localization and mask completeness and was therefore used as the primary instance segmentation metric in this study to characterize mask quality at fruit tips, narrow boundaries, and interfaces between adjacent instances.

3.3. Comparative Experiment

To evaluate the overall performance of OccPepSeg-YOLO for occluded-pepper instance segmentation, it was compared with YOLOv8n-seg, YOLOv9c-seg, YOLO11n-seg, YOLO12n-seg, and YOLOv26n-seg. All models were trained and tested using the same dataset split. The quantitative results are presented in Table 2, and the performance comparison is visualized in Figure 7.

Table 2. Performance comparison among different YOLO-based instance segmentation models.

Model	M-P/%	M-R/%	M-mAP50/%	M-mAP50–95/%	B-P/%	B-R/%	B-mAP50/%	B-mAP50–95/%
YOLOv8n-seg	88.49	87.41	93.35	72.18	87.58	86.28	93.20	77.61
YOLOv9c-seg	87.07	87.41	92.65	73.96	86.85	86.84	92.25	78.92
YOLO11n-seg	88.28	88.91	93.34	72.79	87.91	88.53	93.63	78.07
YOLO12n-seg	88.38	88.61	92.91	72.87	88.19	88.42	92.71	77.47
YOLOv26n-seg	86.15	87.70	91.15	71.86	85.97	87.52	91.12	75.87
Ours	93.87	92.09	97.17	82.31	93.68	91.91	97.17	87.48

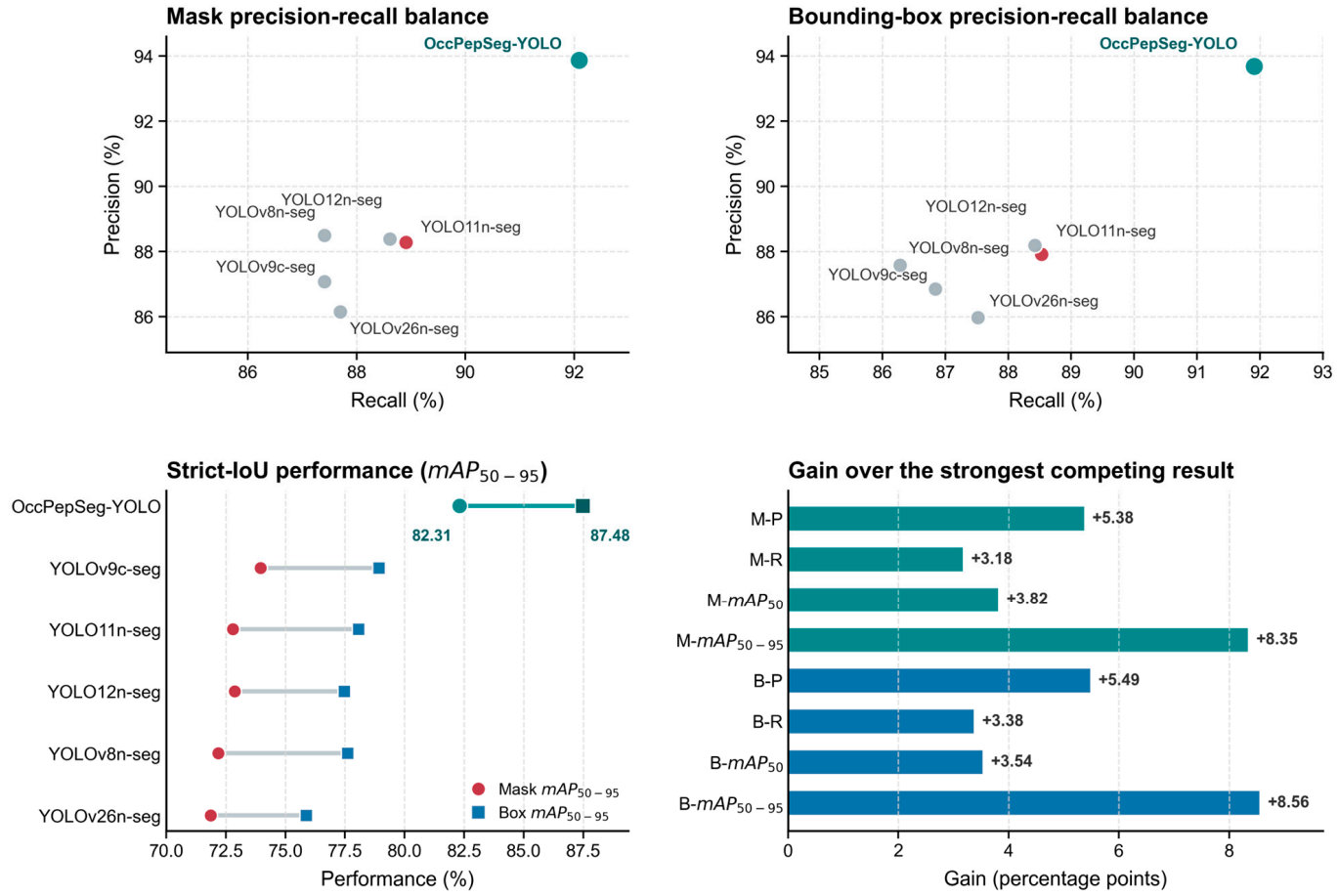

Figure 7. Performance comparison between OccPepSeg-YOLO and other YOLO-based instance segmentation models. Green circles and squares denote the mask and box M-mAP50–95 values, respectively, for OccPepSeg-YOLO.

As shown in Table 2, all five comparison models achieved M-mAP50 values above 91%, indicating that YOLO-series instance segmentation models can identify the main pepper regions effectively. Under the stricter M-mAP50–95 metric, however, the competing models achieved M-mAP50–95 values ranging from 71.86% to 73.96%, revealing insufficient mask quality at occlusion boundaries, slender tips, and interfaces between adjacent instances.

OccPepSeg-YOLO achieved M-mAP50 and M-mAP50–95 values of 97.17% and 82.31%, respectively, both of which were the highest among all models. Compared with the direct baseline YOLO11n-seg, OccPepSeg-YOLO improved M-P, M-R, M-mAP50, and M-mAP50–95 by 5.59, 3.18, 3.83, and 9.52 percentage points, respectively. Bounding-box performance also improved: B-P, B-R, B-mAP50, and B-mAP50–95 increased by 5.77, 3.38, 3.54, and 9.41 percentage points, respectively. These results show that OccPepSeg-YOLO improved not only target localization but, more importantly, instance-mask contour quality under strict IoU thresholds.

3.4. Adjustment of BDoU Loss Weight

BDoU loss was used to improve agreement between predicted and ground-truth masks within the boundary band. To determine an appropriate contribution of BDoU to the total mask loss, comparative experiments were conducted with different BDoU weight coefficients. Let λ denote the BDoU loss weight. When λ = 0, BDoU loss was disabled; as λ increased, the boundary-error constraint during training became stronger. Quantitative performance under different λ values is listed in Table 3 and visualized in Figure 8.

Table 3. Effects of different BDoU loss weights on model performance.

λ	M-P/%	M-R/%	M-mAP50/%	M-mAP50–95/%
0	93.12	91.59	96.36	81.83
0.15	93.46	91.74	96.79	82.14
0.35	93.87	92.09	97.17	82.31
0.50	93.71	91.87	96.82	82.19

Table 3 shows that the BDoU loss weight had a clear influence on instance segmentation performance. When λ increased from 0 to 0.15, M-mAP50 and M-mAP50–95 improved from 96.36% and 81.83% to 96.79% and 82.14%, respectively, indicating that moderate boundary supervision can improve mask prediction for occluded peppers. The best overall performance was achieved at λ = 0.35, where M-P, M-R, M-mAP50, and M-mAP50–95 reached 93.87%, 92.09%, 97.17%, and 82.31%, respectively, corresponding to improvements of 0.75, 0.50, 0.81, and 0.48 percentage points compared with λ = 0. These results suggest that an appropriate boundary-band constraint can improve the segmentation of fruit contours, occlusion interfaces, and adhesion regions between adjacent instances. However, when λ was further increased to 0.50, M-mAP50 and M-mAP50–95 decreased to 96.82% and 82.19%, respectively. This result suggests that excessive emphasis on boundary errors may affect the balance between segmentation of the main fruit regions and refinement of local boundary details, thereby limiting further performance gains. Therefore, λ = 0.35 was selected as the BDoU loss weight in OccPepSeg-YOLO.

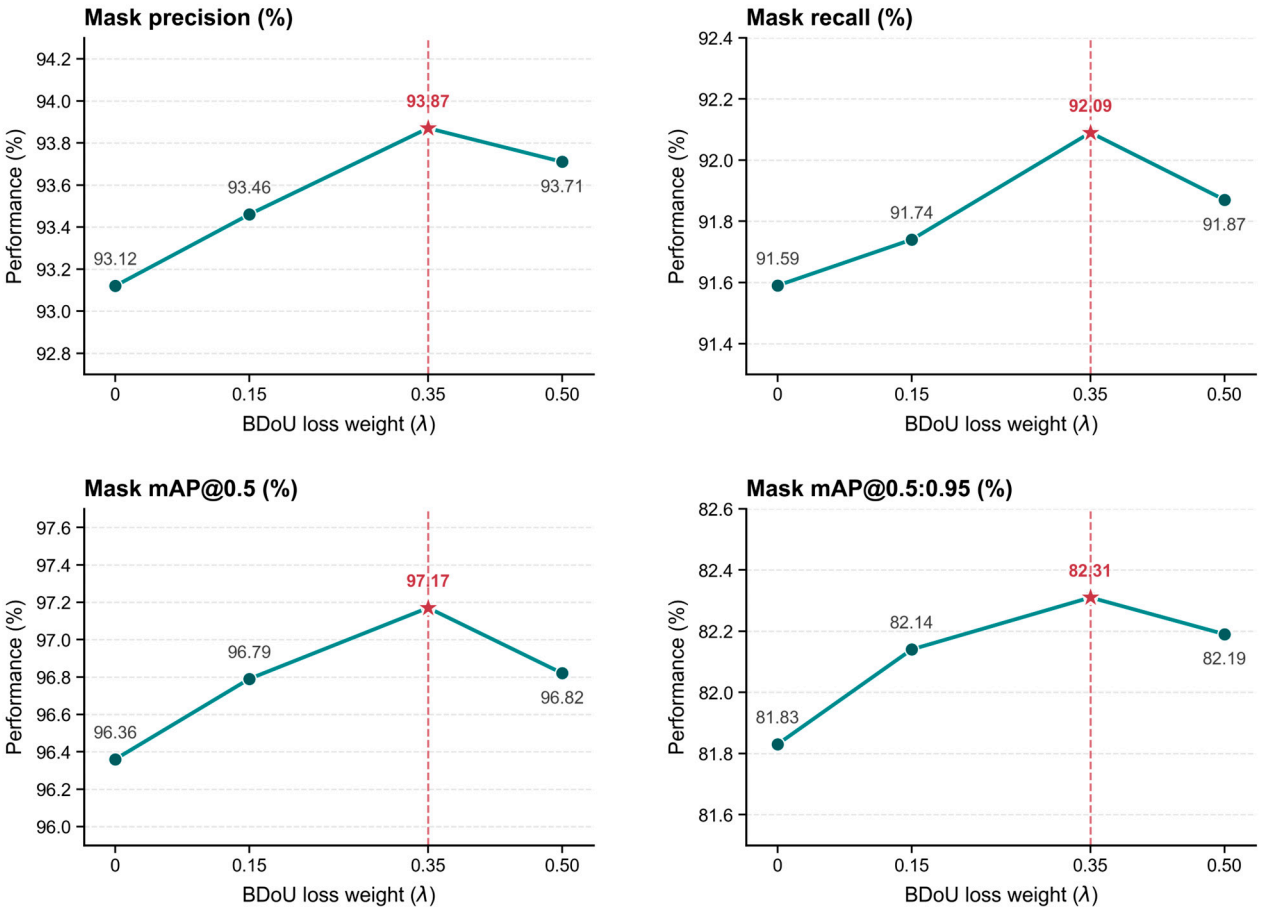

Figure 8. Model performance under different BDoU loss weights. The red star marks the optimal setting (λ = 0.35).

3.5. Ablation Experiment

To evaluate the contribution of each proposed component, stepwise ablation experiments were conducted using YOLO11n-seg as the baseline. A0 denotes the original YOLO11n-seg; A1 adds P2FreqFusion to A0; A2 further introduces ASC; A3 additionally incorporates BoundaryGate; A4 replaces the original segmentation head with OccPepSegment; and A5 adds BDoU loss to A4, forming the complete OccPepSeg-YOLO model. The quantitative results are presented in Table 4, and the ablation results are visualized in Figure 9.

Table 4. Ablation results for the proposed modules and loss function.

Group	P2	ASC	BoundaryGate	OccPepSegment	BDoU	M-P/%	M-R/%	M-mAP50/%	M-mAP50–95/%	FPS
A0	-	-	-	-	-	88.28	88.91	93.34	72.79	106.49
A1	√	-	-	-	-	90.61	89.76	94.29	76.72	76.81
A2	√	√	-	-	-	91.04	90.38	95.15	78.37	74.39
A3	√	√	√	-	-	92.37	91.03	95.83	80.47	68.67
A4	√	√	√	√	-	93.12	91.59	96.36	81.83	63.29
A5	√	√	√	√	√	93.87	92.09	97.17	82.31	63.29

Note: A check mark indicates that the module or loss term was included; - indicates that it was not included. FPS values were measured using the same repeated-inference protocol described in Section 3.1.

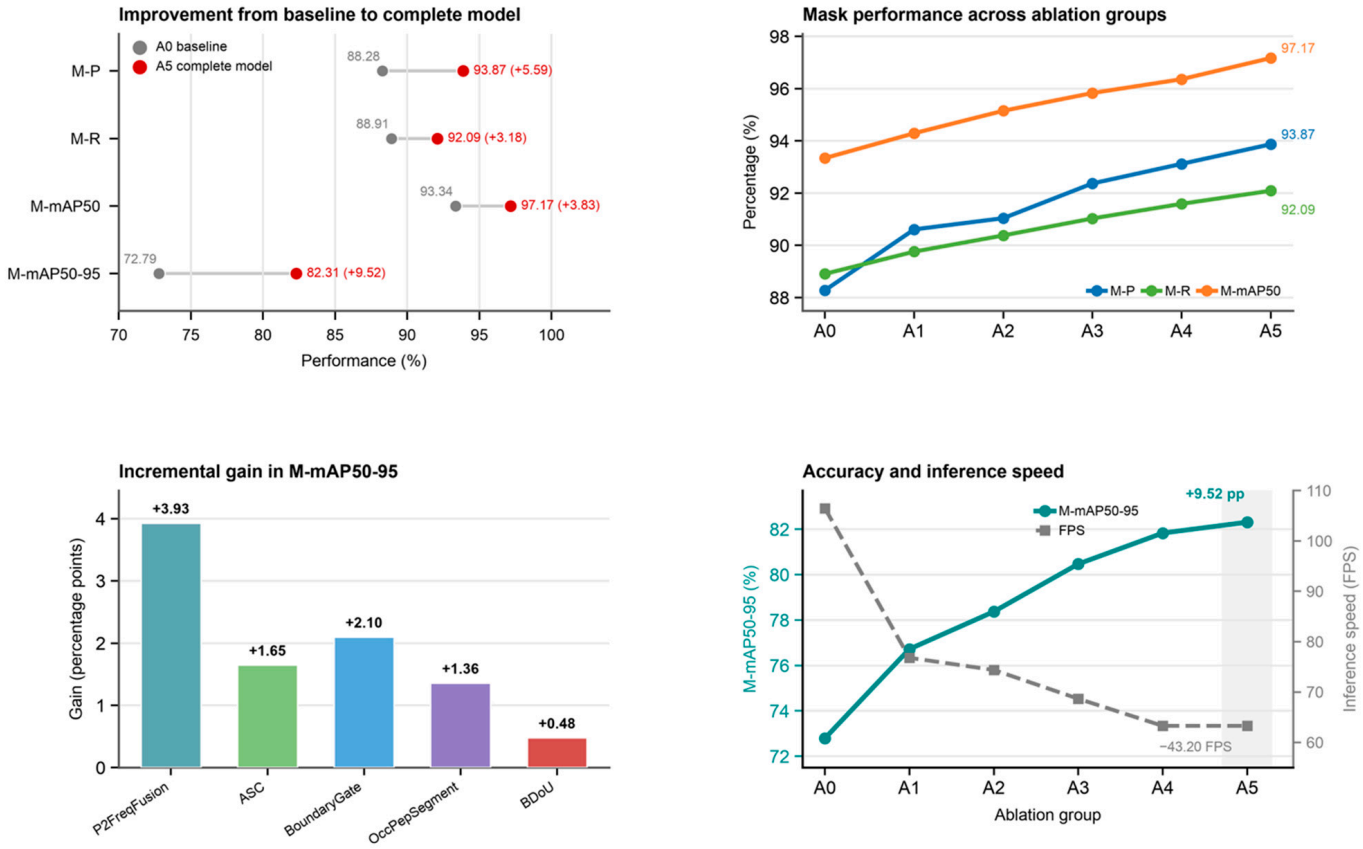

Figure 9. Ablation results for the proposed components of OccPepSeg-YOLO. The light-gray shaded area denotes the final ablation configuration (A5).

The baseline model A0 achieved M-mAP50 and M-mAP50–95 values of 93.34% and 72.79%, respectively, indicating effective segmentation of the main pepper regions but considerable room for improvement under stricter IoU thresholds. After adding P2FreqFusion, the M-mAP50–95 of A1 increased to 76.72%, an improvement of 3.93 percentage points over A0, supporting the effectiveness of high-resolution feature fusion for representing fruit-edge details. Adding ASC increased the M-mAP50–95 of A2 to 78.37%, a further gain of 1.65 percentage points, indicating that directional and scale-aware morphology modeling helps represent slender, curved peppers.

After BoundaryGate was introduced, the M-mAP50–95 of A3 reached 80.47%, an increase of 2.10 percentage points over A2, demonstrating the beneficial effect of boundary enhancement on occlusion interfaces and contact regions between adjacent instances. Replacing the segmentation head with OccPepSegment further increased the M-mAP50–95 of A4 to 81.83%, a gain of 1.36 percentage points, indicating that multi-scale region, boundary, and skeleton prototypes improve instance-mask representation. Adding BDoU loss produced the complete A5 model, whose M-mAP50–95 reached 82.31%, 0.48 percentage points higher than that of A4. Overall, A5 improved M-mAP50–95 by 9.52 percentage points relative to the A0 baseline, confirming that each component made a positive incremental contribution to occluded-pepper instance segmentation.

The A0 and A5 predictions in Figure 10 illustrate the overall improvement of the complete OccPepSeg-YOLO framework over the YOLO11n-seg baseline. The independent contribution of OccPepSegment is further quantified by the A3–A4 ablation in Table 4, where M-mAP50–95 increased from 80.47% to 81.83%.

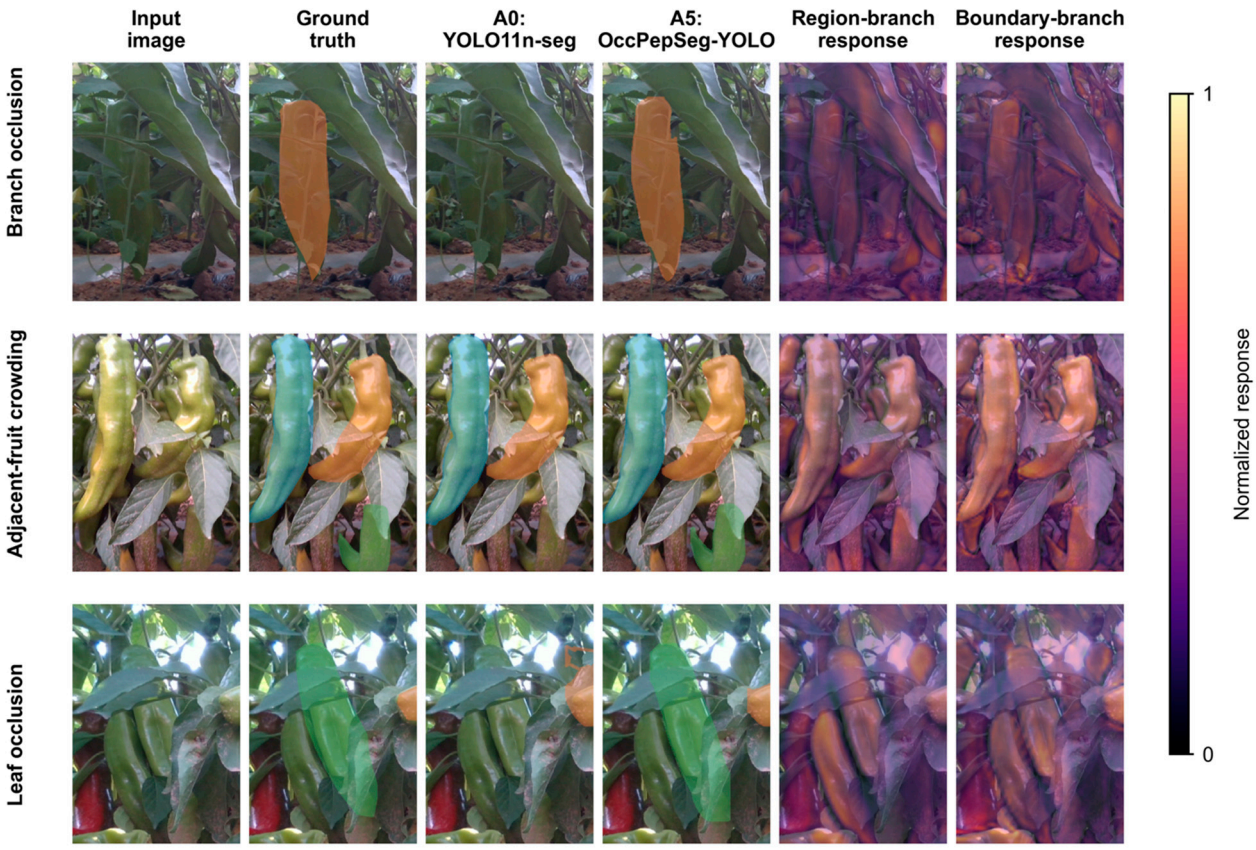

Figure 10. Mechanism-oriented visualization of OccPepSegment in challenging field scenes. Each row presents a representative case involving branch occlusion, adjacent-fruit crowding, or leaf occlusion. The input image, ground-truth mask, A0 prediction from YOLO11n-seg, A5 prediction from the complete OccPepSeg-YOLO, and normalized region- and boundary-branch responses extracted from OccPepSegment are shown.

The internal responses provide further insight into the behavior of OccPepSegment. Stronger region-branch responses were mainly distributed over the visible fruit bodies, whereas the boundary branch responded more prominently around fruit contours, occlusion interfaces, and regions where adjacent instances were close to one another. These complementary patterns are consistent with the structural design of OccPepSegment: the region prototype supports complete representation of the main fruit area, the boundary prototype refines fruit contours and inter-instance interfaces, and the skeleton prototype preserves the central structure and major-axis continuity of slender fruits. Figure 10 focuses on the region- and boundary-branch responses to provide an intuitive view of how the segmentation head represents fruit regions and boundaries. Together with the quantitative improvement from A3 to A4, these observations indicate that the coordinated fusion of multi-scale prototype information contributes to improved region completeness, boundary localization, and separation of adjacent instances under occlusion.

3.6. Multi-Objective Analysis of Segmentation Accuracy, Model Complexity, and Inference Efficiency

To provide a comprehensive evaluation of the compared instance segmentation models, segmentation accuracy, model scale, theoretical computational complexity, and measured inference efficiency were considered jointly. M-mAP50–95 was used to evaluate instance-mask quality across IoU thresholds ranging from 0.50 to 0.95, with greater sensitivity to mask quality under stricter overlap criteria. Parameter count and weight-file size were used to characterize model scale, FLOPs were used to describe theoretical computational

complexity at a fixed input resolution, and mean latency, latency standard deviation, P95 latency, and FPS were used to characterize measured inference efficiency. Because OccPepSeg-YOLO was developed from YOLO11n-seg, the cross-generation comparison included the nominally compact n-scale variants YOLOv8n-seg, YOLO11n-seg, YOLO12n-seg, and YOLOv26n-seg, while their actual differences in parameter count, FLOPs, and weight-file size were explicitly reported. In addition, YOLOv9c-seg was retained as a substantially larger and more computationally demanding competitor, allowing the evaluation to span a broader range of model sizes and computational demands rather than being restricted to only the smallest variants. The quantitative results are summarized in Table 5, and the corresponding multi-objective relationships are visualized in Figure 11.

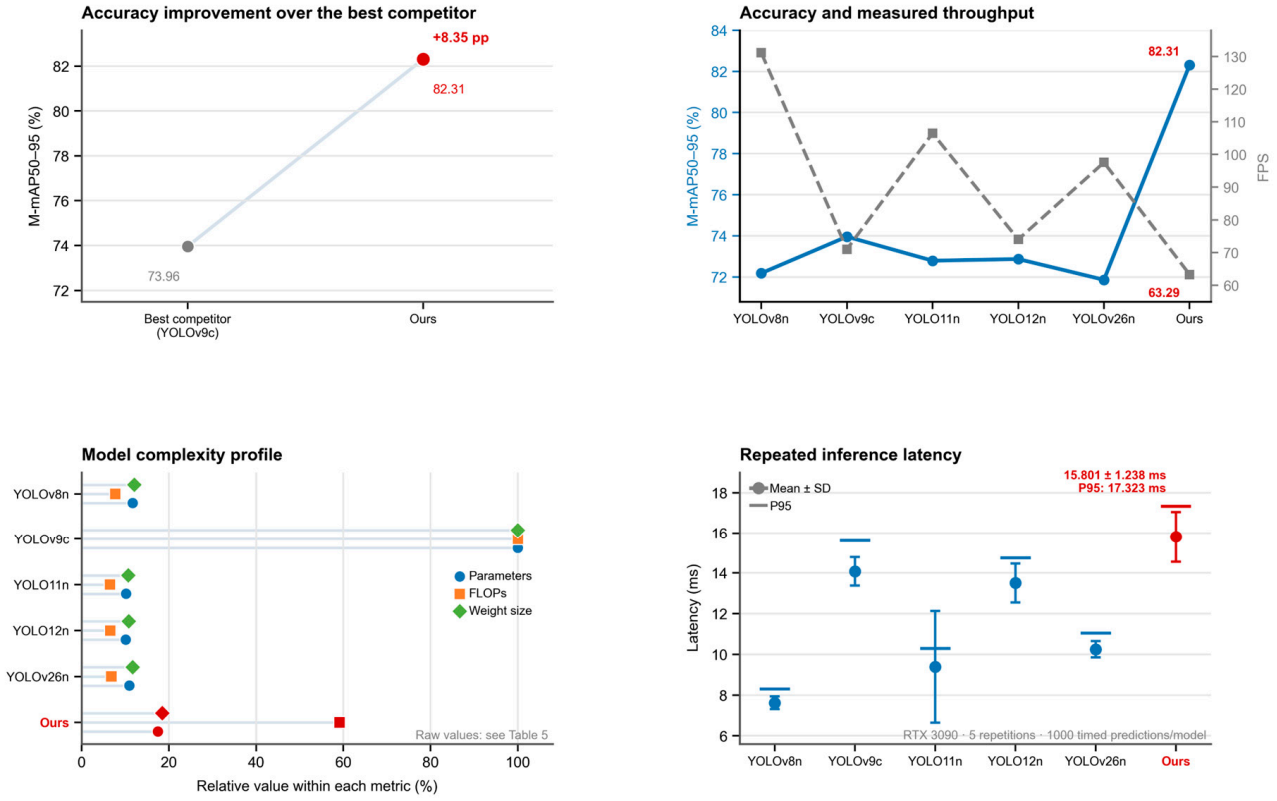

Figure 11. Multi-objective comparison of segmentation accuracy, model complexity, and inference efficiency among the evaluated models. The figure summarizes the improvement in M-mAP50–95 over the best competing model, the relationship between segmentation accuracy and measured throughput, the normalized model-complexity profile, and repeated inference latency on an NVIDIA RTX 3090 GPU. Red markers indicate the values of OccPepSeg-YOLO.

Table 5. Multi-objective comparison of segmentation accuracy, model complexity, and repeated inference efficiency on an RTX 3090.

Model	M-mAP50–95/%	Parameters/M	FLOPs/G	Weight Size/MB	Latency/ms (Mean ± SD)	P95 Latency/ms	FPS
YOLOv8n-seg	72.18	3.264	11.48	6.49	7.625 ± 0.317	8.293	131.16
YOLOv9c-seg	73.96	27.836	149.01	53.64	14.081 ± 0.704	15.601	71.02
YOLO11n-seg	72.79	2.843	9.73	5.75	9.390 ± 2.739	10.284	106.49
YOLO12n-seg	72.87	2.821	9.77	5.81	13.511 ± 0.953	14.743	74.01
YOLOv26n-seg	71.86	3.053	10.16	6.29	10.250 ± 0.403	11.049	97.56
Ours	82.31	4.873	88.08	9.93	15.801 ± 1.238	17.323	63.29

Note: Inference efficiency was measured using the unified repeated-inference protocol described in Section 3.1. The table reports mean latency, latency standard deviation, and P95 latency, while FPS was calculated from the mean latency.

As shown in Table 5 and Figure 11, the evaluated models exhibited clear trade-offs between segmentation accuracy and measured inference throughput. YOLOv8n-seg achieved the highest throughput at 131.16 FPS, but its M-mAP50–95 was 72.18%. YOLO11n-seg and YOLOv26n-seg achieved throughputs of 106.49 and 97.56 FPS, respectively, with corresponding M-mAP50–95 values of 72.79% and 71.86%. YOLO12n-seg achieved 74.01 FPS with an M-mAP50–95 of 72.87%. Among the competing models, YOLOv9c-seg obtained the highest M-mAP50–95 of 73.96% while operating at 71.02 FPS. In comparison, OccPepSeg-YOLO achieved an M-mAP50–95 of 82.31%, exceeding the best competing result by 8.35 percentage points. In the repeated inference benchmark, OccPepSeg-YOLO achieved a mean latency of 15.801 $\pm$ 1.238 ms and a P95 latency of 17.323 ms, corresponding to 63.29 FPS. These results indicate that OccPepSeg-YOLO favors instance-mask accuracy under strict IoU criteria and accepts some reduction in inference throughput in exchange for substantially improved segmentation quality.

The model-scale comparison provides further context for this accuracy–efficiency trade-off. OccPepSeg-YOLO contains 4.873 M parameters and has a weight-file size of 9.93 MB. Although these values are higher than those of the compact n-scale competing models, they remain substantially lower than the 27.836 M parameters and 53.64 MB weight file of YOLOv9c-seg. Meanwhile, OccPepSeg-YOLO exceeds YOLOv9c-seg in M-mAP50–95 by 8.35 percentage points. Therefore, the improvement in strict mask accuracy cannot be explained simply by a substantial increase in parameter count or model-storage size. Instead, the proposed architecture achieves higher segmentation accuracy while remaining considerably smaller than the largest competing model in terms of parameter count and weight-file size. On the other hand, OccPepSeg-YOLO requires 88.08 G FLOPs, substantially higher than the approximately 9–11 G FLOPs required by the compact n-scale models, indicating that the proposed architecture introduces additional computation while maintaining a relatively modest parameter count and storage requirement.

From an architectural perspective, the increased computational cost of OccPepSeg-YOLO mainly arises from high-resolution feature processing and multi-branch feature modeling. P2FreqFusion introduces additional upsampling, gating, and adaptive fusion operations at the relatively high-resolution P2 level, allowing more fine-grained spatial details to participate in subsequent feature representation. ASC employs multiple parallel branches for local, horizontal, vertical, and dilated major-axis modeling to capture morphological information at different directions and scales, thereby introducing additional convolutional computation. BoundaryGate further adds boundary-response generation and feature-refinement operations to enhance the representation of occlusion interfaces and boundaries between adjacent instances. Meanwhile, OccPepSegment aligns and fuses P2-, P3-, and P4-scale features and constructs region, boundary, and skeleton prototype branches before generating the final prototype through adaptive gating, which further increases the computational load of mask generation. In contrast, BDoU is involved only in mask-loss optimization during training and does not modify the inference-time network structure or computational pathway. Overall, the higher FLOPs of OccPepSeg-YOLO are mainly associated with high-resolution feature fusion, multi-branch morphological modeling, boundary enhancement, and multi-scale prototype generation rather than simply with an increase in parameter count.

The results further show that theoretical computational complexity does not correspond directly to measured inference latency. For example, YOLO11n-seg has fewer parameters and lower FLOPs than YOLOv8n-seg, yet its mean latency was higher, at 9.390 $\pm$ 2.739 ms compared with 7.625 $\pm$ 0.317 ms. Similarly, YOLOv9c-seg requires 149.01 G FLOPs, compared with 88.08 G for OccPepSeg-YOLO, but its mean latency of 14.081 $\pm$ 0.704 ms was slightly lower than the 15.801 $\pm$ 1.238 ms measured for OccPepSeg-

YOLO. This occurs because FLOPs primarily characterize theoretical operation counts, whereas measured runtime is also affected by operator composition, feature-map memory access, GPU parallel execution, CUDA kernel scheduling, and post-processing overhead. Therefore, segmentation accuracy, parameter count, FLOPs, weight-file size, and measured latency should be considered jointly rather than using any single indicator to characterize model effectiveness.

Overall, OccPepSeg-YOLO is positioned toward the accuracy-oriented end of the observed accuracy–scale–efficiency trade-off. Under the unified RTX 3090 benchmark, it achieved the highest M-mAP50–95 among all evaluated models while maintaining a parameter count of 4.873 M, a weight-file size of 9.93 MB, and a measured throughput of 63.29 FPS. Table 5 and Figure 11 therefore provide complementary quantitative and visual assessments of the trade-offs among segmentation accuracy, model scale, theoretical computational complexity, and measured inference efficiency under identical benchmarking conditions. Future work will further examine how these accuracy–efficiency characteristics transfer to resource-constrained agricultural vision hardware and complete field-vision processing pipelines.

3.7. Recognition Experiment of Occluded Peppers in the Field

To compare segmentation performance in complex field scenes, five representative groups of test images were selected for qualitative analysis, as shown in Figure 12. Columns (a)–(e) show scenes with (a) branch and leaf occlusion, (b) foreground interference, (c) pronounced differences in fruit scale, (d) adhesion between adjacent fruit instances, and (e) complex backgrounds. Rows, from top to bottom, show the predictions of YOLOv8n-seg, YOLOv9c-seg, YOLO11n-seg, YOLO12n-seg, YOLOv26n-seg, and OccPepSeg-YOLO (Ours), respectively.

As shown in Figure 12, all models recognized the main pepper instances, but their segmentation differed at occluded regions, fruit tips, and interfaces between neighboring instances. Some comparison models produced truncated contours and local missed regions under leaf or branch occlusion. Where adjacent fruits touched or overlapped, some predictions showed displaced instance boundaries, incomplete mask coverage, or insufficient separation between neighboring instances. Complex backgrounds and dense fruit distributions further increased the difficulty of recognizing small-scale fruits and determining instance boundaries.

In contrast, OccPepSeg-YOLO produced relatively complete fruit contours in the displayed samples and clearer instance separation at occlusion interfaces, slender fruit tips, and contact regions between adjacent instances. It also reduced local omissions, false detections, contour breakage, and boundary displacement. These qualitative observations are consistent with the highest M-mAP50–95 reported for OccPepSeg-YOLO in Table 2, indicating that its improvement is reflected not only in aggregate metrics but also in mask completeness and boundary localization in representative field images. Figure 12 is intended only to illustrate prediction differences on selected samples; model performance should be judged primarily from the quantitative results on the complete test set.

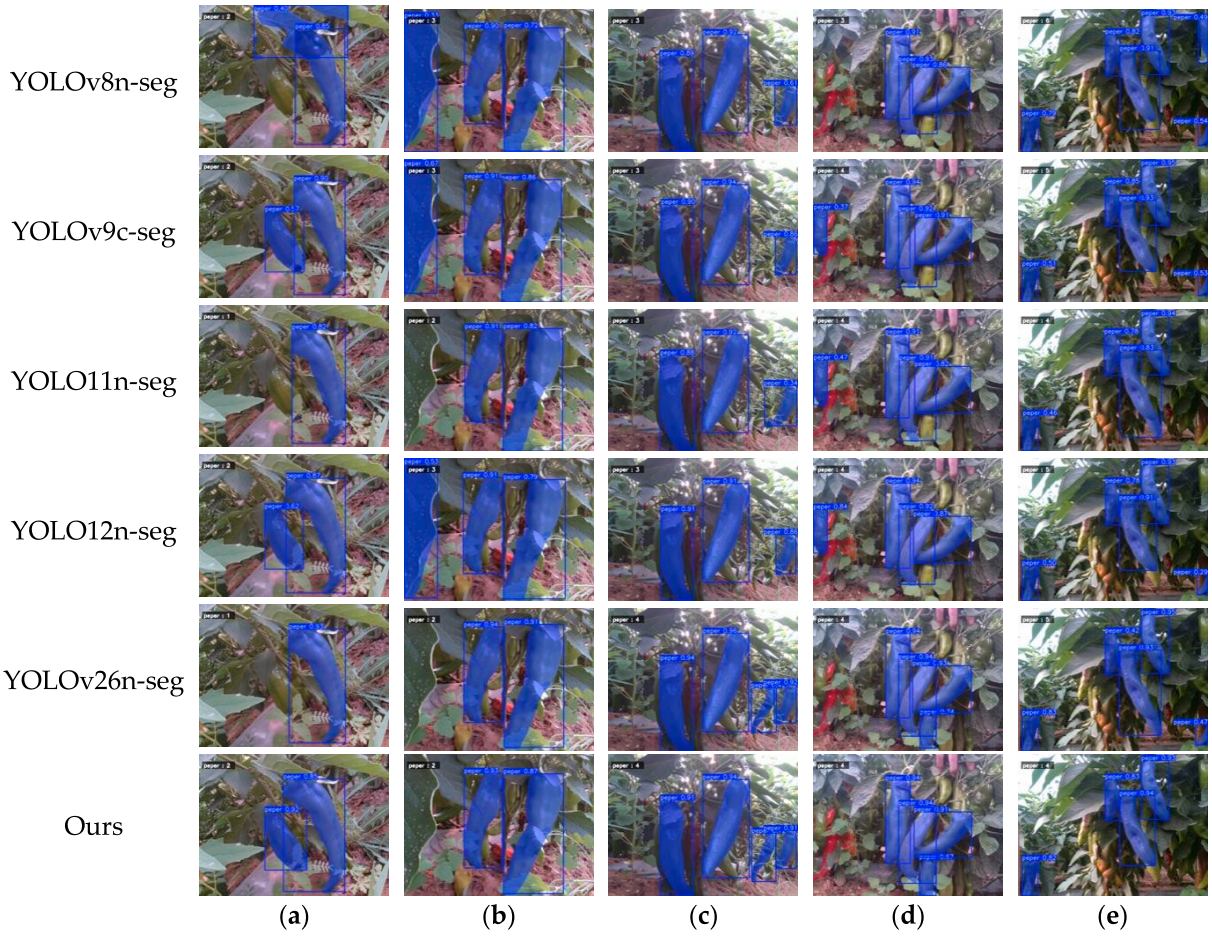

Figure 12. Qualitative comparison of instance segmentation results in complex field scenes. Columns (**a**)–(**e**) show representative test images with (**a**) branch and leaf occlusion, (**b**) foreground interference, (**c**) pronounced differences in fruit scale, (**d**) adhesion between adjacent fruit instances, and (**e**) complex backgrounds. Rows, from top to bottom, show predictions from YOLOv8n-seg, YOLOv9c-seg, YOLO11n-seg, YOLO12n-seg, YOLOv26n-seg, and OccPepSeg-YOLO (Ours), respectively.

4. Discussion

The comparative experiments demonstrated that OccPepSeg-YOLO consistently outperformed the YOLO11n-seg baseline in both pepper detection and instance segmentation. Relative to YOLO11n-seg, B-mAP50 and B-mAP50–95 increased by 3.54 and 9.41 percentage points, respectively, while M-mAP50 and M-mAP50–95 increased by 3.83 and 9.52 percentage points. Notably, the improvements in mAP50–95 were substantially greater than those in mAP50. Because mAP50–95 evaluates prediction quality over multiple and increasingly strict IoU thresholds ranging from 0.50 to 0.95, this result indicates that the proposed improvements contributed not only to target recognition but also to more accurate localization and instance-mask alignment. This advantage is particularly relevant for field-grown peppers, whose slender and curved morphology, narrow fruit tips, partial occlusion, and complex spatial relationships with leaves, branches, and neighboring fruits make precise contour prediction more difficult than simple target detection. The results therefore suggest that structural optimization for high-resolution details, elongated-shape representation, and occlusion-boundary discrimination is effective for improving the completeness and spatial accuracy of pepper instance masks.

The ablation experiments further showed that the proposed components provide complementary contributions to the overall improvement. P2FreqFusion introduces shallow high-resolution information into the feature-fusion process, helping to preserve spatial

details at fruit tips, narrow edges, and locally occluded regions. ASC models morphological information over different directions and receptive-field scales, which is beneficial for maintaining the structural continuity of slender and curved pepper fruits. BoundaryGate enhances responses around fruit contours, occlusion interfaces, and contact regions between neighboring fruits. On this basis, OccPepSegment further improves mask representation by integrating multi-scale region, boundary, and skeleton prototype information, while BDoU provides additional supervision for boundary discrepancies during training. The progressive increase in M-mAP50–95 as these components were introduced indicates that the overall performance gain results from the coordinated enhancement of spatial detail, morphological structure, boundary discrimination, and prototype-mask representation rather than from a single isolated modification. The mechanism-oriented visualization of OccPepSegment is also consistent with this interpretation, with the region branch responding mainly to the visible fruit bodies and the boundary branch showing more concentrated responses around fruit contours and occlusion interfaces.

The qualitative field-recognition results were consistent with the quantitative evaluation. Under challenging conditions involving branch and leaf occlusion, fruit overlap, similar-color backgrounds, scale variation, and dense fruit distributions, OccPepSeg-YOLO produced relatively complete instance masks in the displayed samples and reduced local omissions, truncated fruit tips, mask adhesion between neighboring instances, and displacement of occlusion boundaries. These improvements were accompanied by additional computational cost. Under the unified repeated-inference benchmark on the RTX 3090, OccPepSeg-YOLO achieved a mean inference latency of 15.801 $\pm$ 1.238 ms, a P95 latency of 17.323 ms, and a throughput of 63.29 FPS. Although its inference throughput was lower than that of several compact comparison models, it achieved substantially higher M-mAP50–95, reflecting an accuracy–efficiency trade-off in which additional high-resolution and multi-branch computation was exchanged for improved mask quality. The model therefore retains relatively high inference throughput under the tested desktop-GPU environment, although its practical performance on resource-constrained agricultural hardware still requires dedicated evaluation.

The significance of field-pepper instance segmentation extends beyond improving image-level recognition accuracy, because instance masks provide fruit information with explicit instance identities and pixel-level spatial extents for subsequent agricultural vision tasks. Compared with bounding-box detection, an instance mask describes the visible contour of each pepper more precisely and can distinguish neighboring, partially overlapping, or occluded fruits as separate instances. High-quality instance masks can therefore provide a more reliable basis for fruit counting and yield estimation by reducing errors associated with missed fruits, duplicate counting, and instance confusion in densely distributed and occluded scenes. Pixel-level fruit contours can also support the extraction of two-dimensional phenotypic characteristics, such as fruit length, width, projected area, shape, and major-axis orientation, thereby providing useful information for field phenotyping, growth assessment, and yield-related analysis.

For downstream operations such as robotic harvesting, instance segmentation provides more precise target regions and clearer spatial boundaries between fruits, leaves, branches, and neighboring fruits than rectangular detections. Such information can facilitate subsequent estimation of fruit centers, major-axis orientations, occlusion status, and candidate operational regions, providing visual cues for determining manipulator approach directions and harvesting regions. When further combined with depth information, pixel-level instance masks could also support three-dimensional fruit localization, spatial-pose estimation, approach-path planning, and picking-point determination. Reliable instance segmentation of peppers under complex occlusion can therefore serve as an

important front-end perception step linking field-image analysis with downstream tasks such as fruit counting, phenotypic measurement, yield estimation, and robotic harvesting. It should be noted that the present study primarily evaluates two-dimensional instance segmentation from RGB images; three-dimensional localization and robotic manipulation represent downstream applications that can build upon the segmentation results rather than system-level functions directly validated in this study.

The results also indicate that segmentation accuracy alone is insufficient for evaluating the practical applicability of an agricultural vision model. OccPepSeg-YOLO contains 4.873 M parameters and has a weight-file size of 9.93 MB, while its theoretical computational demand reaches 88.08 G FLOPs. This computational cost mainly results from the additional high-resolution P2 feature fusion, multi-branch morphological modeling, boundary enhancement, and multi-scale prototype generation introduced by the proposed architecture. The increase in measured latency relative to YOLO11n-seg also confirms that these operations introduce practical runtime overhead. However, FLOPs and measured latency are not strictly proportional across different models because practical GPU inference is also affected by operator composition, feature-map memory access, parallel execution efficiency, CUDA kernel scheduling, and post-processing. Therefore, the practical performance of OccPepSeg-YOLO should be interpreted by jointly considering instance-mask accuracy, parameter count, theoretical computational complexity, and measured inference efficiency. In the present study, the model is positioned toward the accuracy-oriented side of this trade-off, which is relevant for applications in which accurate instance boundaries and separation of occluded fruits are more important than maximizing throughput alone.

Several limitations remain. First, the current dataset was collected primarily within a specific region, season, and cultivation environment and therefore does not yet provide a comprehensive evaluation of generalization across pepper cultivars, growth stages, geographic regions, and imaging platforms. Second, under extreme illumination, extensive leaf occlusion, or severe overlap among multiple fruits, the model may still produce local omissions, contour errors, or boundary displacement. Third, the present experiments were conducted mainly on offline RGB images. Although OccPepSeg-YOLO achieved a throughput of 63.29 FPS under the unified RTX 3090 benchmark, its operational stability during continuous field work, robustness to camera motion and dynamic illumination changes, and deployment performance on edge-computing devices have not yet been systematically evaluated. In addition, this study focuses on two-dimensional instance segmentation and does not directly validate downstream tasks such as three-dimensional fruit localization, robotic approach planning, or picking-point determination. Future work will therefore expand the dataset across regions, seasons, cultivars, and imaging conditions, further reduce computational complexity and improve deployment efficiency, and integrate instance segmentation with depth sensing and robotic perception systems to evaluate its practical value for fruit counting, phenotypic measurement, yield estimation, and autonomous harvesting.

5. Conclusions

To address the slender and curved morphology, partial occlusion, ambiguous boundaries, and adhesion between adjacent pepper instances in complex field environments, this study developed OccPepSeg-YOLO on the basis of YOLO11n-seg. By improving high-resolution feature fusion, morphological feature modeling, boundary responses, prototype-mask generation, and the loss function, the model enhanced the boundary quality and structural completeness of occluded-pepper instance masks. The main conclusions are as follows:

(1) P2FreqFusion, ASC, and BoundaryGate enhance the representation of shallow spatial details, slender and curved structures, and inter-instance interfaces, respectively. After these three modules were added sequentially, M-mAP50–95 increased from 72.79% for the baseline to 80.47%, confirming the importance of high-resolution information, directional and scale-related features, and boundary responses for instance segmentation of occluded peppers.

(2) By fusing region, boundary, and skeleton prototypes, OccPepSegment strengthens the representation of the pepper body, contour, and major-axis structure, increasing M-mAP50–95 from 80.47% to 81.83%. After BDoU loss was further introduced, M-mAP50–95 reached 82.31%, showing that multi-scale prototype representation and boundary supervision further improve mask completeness and boundary alignment.

(3) The complete model achieved M-P, M-R, M-mAP50, and M-mAP50–95 values of 93.87%, 92.09%, 97.17%, and 82.31%, respectively, representing improvements of 5.59, 3.18, 3.83, and 9.52 percentage points over YOLO11n-seg. In field images, the model adapted well to leaf occlusion, fruit overlap, similar-color backgrounds, and scale variation. Under the unified repeated-inference benchmark on the RTX 3090, the model achieved a measured throughput of 63.29 FPS.

Overall, OccPepSeg-YOLO effectively improves the instance segmentation quality of occluded peppers in complex field environments. Its generalization across regions, cultivars, and devices, as well as its stability in practical deployment, requires further validation. Future research will expand diverse field data and reduce computational complexity to improve applicability to agricultural robots and edge-computing platforms.

Author Contributions: Conceptualization, S.X. and X.Y.; methodology, X.Y. and M.J.; software, M.J.; validation, X.Y., M.J., F.G. and Y.F.; formal analysis, X.Y. and M.J.; investigation, X.Y., F.G., Y.F., Y.B., Z.P., Q.L. (Qi Lu) and Q.L. (Qian Liu); resources, X.Y., Z.P., Q.L. (Qi Lu) and Q.L. (Qian Liu); data curation, X.Y., M.J. and Y.B.; writing—original draft preparation, X.Y. and M.J.; writing—review and editing, S.X., F.G., Y.F., Y.B., Z.P., Q.L. (Qi Lu) and Q.L. (Qian Liu); visualization, S.X.; supervision, S.X.; project administration, X.Y.; funding acquisition, S.X. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Ordos Municipal Bureau of Agriculture and Animal Husbandry through the 2024 Agriculture and Animal Husbandry Sector "Open Bidding for Selecting the Best Candidates" Project (grant number BBDS-2024-JG-NJ-01), and by the Science and Technology Department of Xizang Autonomous Region through the Xizang Autonomous Region Science and Technology Plan Project (grant number XZ202601ZY0120).

Data Availability Statement: The data and code supporting the findings of this study are openly available in Zenodo at https://doi.org/10.5281/zenodo.21907482 (accessed on 12 August 2026).

Acknowledgments: The authors thank all members of the research team for their contributions to this work.

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

References

1. Todorova, V.; Nankar, A.N.; Yankova, V.; Tringovska, I.; Markova, D. Assessment of Balkan Pepper (Capsicum annuum L.) Accessions for Agronomic, Fruit Quality, and Pest Resistance Traits. *Horticulturae* **2024**, *10*, 389. [CrossRef]
2. Duan, Y.; Li, J.; Zou, C. Research on Detection Method of Chaotian Pepper in Complex Field Environments Based on YOLOv8. *Sensors* **2024**, *24*, 5632. [CrossRef] [PubMed]
3. Chen, H.; Zhang, R.; Peng, J.; Peng, H.; Hu, W.; Wang, Y.; Jiang, P. YOLO-Chili: An Efficient Lightweight Network Model for Localization of Pepper Picking in Complex Environments. *Appl. Sci.* **2024**, *14*, 5524. [CrossRef]
4. Huang, Y.; Zhong, Y.; Zhong, D.; Yang, C.; Wei, L.; Zou, Z.; Chen, R. Pepper-YOLO: An Lightweight Model for Green Pepper Detection and Picking Point Localization in Complex Environments. *Front. Plant Sci.* **2024**, *15*, 1508258. [CrossRef] [PubMed]

5. Nan, Y.; Zhang, H.; Zeng, Y.; Zheng, J.; Ge, Y. Faster and Accurate Green Pepper Detection Using NSGA-II-Based Pruned YOLOv5l in the Field Environment. *Comput. Electron. Agric.* **2023**, *205*, 107563. [CrossRef]
6. Li, H.; Huang, J.; Gu, Z.; He, D.; Huang, J.; Wang, C. Positioning of Mango Picking Point Using an Improved YOLOv8 Architecture with Object Detection and Instance Segmentation. *Biosyst. Eng.* **2024**, *247*, 202–220. [CrossRef]
7. Wang, W.; Shan, Y.; Hu, T.; Gu, J.; Zhu, Y.; Gao, Y. Locating Apple Picking Points Using Semantic Segmentation of Target Region. *Trans. Chin. Soc. Agric. Eng.* **2024**, *40*, 172–178. [CrossRef]
8. Zhang, F.; Sun, H.; Xie, S.; Dong, C.; Li, Y.; Xu, Y.; Zhang, Z.; Chen, F. A Tea Bud Segmentation, Detection and Picking Point Localization Based on the MDY7-3PTB Model. *Front. Plant Sci.* **2023**, *14*, 1199473. [CrossRef] [PubMed]
9. Song, G.; Wang, J.; Ma, R.; Shi, Y.; Wang, Y. Study on the Fusion of Improved YOLOv8 and Depth Camera for Bunch Tomato Stem Picking Point Recognition and Localization. *Front. Plant Sci.* **2024**, *15*, 1447855. [CrossRef] [PubMed]
10. Cong, P.; Li, S.; Zhou, J.; Lv, K.; Feng, H. Research on Instance Segmentation Algorithm of Greenhouse Sweet Pepper Detection Based on Improved Mask RCNN. *Agronomy* **2023**, *13*, 196. [CrossRef]
11. Paul, A.; Machavaram, R.; Ambuj; Kumar, D.; Nagar, H. Smart Solutions for Capsicum Harvesting: Unleashing the Power of YOLO for Detection, Segmentation, Growth Stage Classification, Counting, and Real-Time Mobile Identification. *Comput. Electron. Agric.* **2024**, *219*, 108832. [CrossRef]
12. Li, Y.; Wang, W.; Guo, X.; Wang, X.; Liu, Y.; Wang, D. Recognition and Positioning of Strawberries Based on Improved YOLOv7 and RGB-D Sensing. *Agriculture* **2024**, *14*, 624. [CrossRef]
13. Jia, W.; Liu, J.; Lu, Y.; Liu, Q.; Zhang, T.; Dong, X. Polar-Net: Green Fruit Instance Segmentation in Complex Orchard Environment. *Front. Plant Sci.* **2022**, *13*, 1054007. [CrossRef] [PubMed]
14. Peng, H.; Zhong, J.; Liu, H.; Li, J.; Yao, M.; Zhang, X. ResDense-Focal-DeepLabV3+ Enabled Litchi Branch Semantic Segmentation for Robotic Harvesting. *Comput. Electron. Agric.* **2023**, *206*, 107691. [CrossRef]
15. Xie, J.; Jing, T.; Chen, B.; Peng, J.; Zhang, X.; He, P.; Yin, H.; Sun, D.; Wang, W.; Xiao, A.; et al. Method for Segmentation of Litchi Branches Based on the Improved DeepLabv3+. *Agronomy* **2022**, *12*, 2812. [CrossRef]
16. Huang, X.; Peng, D.; Qi, H.; Zhou, L.; Zhang, C. Detection and Instance Segmentation of Grape Clusters in Orchard Environments Using an Improved Mask R-CNN Model. *Agriculture* **2024**, *14*, 918. [CrossRef]
17. Shui, Y.; Yuan, K.; Wu, M.; Zhao, Z. Improved Multi-Size, Multi-Target and 3D Position Detection Network for Flowering Chinese Cabbage Based on YOLOv8. *Plants* **2024**, *13*, 2808. [CrossRef] [PubMed]
18. Zhang, D.; Lu, R.; Guo, Z.; Yang, Z.; Wang, S.; Hu, X. Algorithm for Locating Apical Meristematic Tissue of Weeds Based on YOLO Instance Segmentation. *Agronomy* **2024**, *14*, 2121. [CrossRef]
19. Rong, Q.; Hu, C.; Hu, X.; Xu, M. Picking Point Recognition for Ripe Tomatoes Using Semantic Segmentation and Morphological Processing. *Comput. Electron. Agric.* **2023**, *210*, 107923. [CrossRef]
20. Lei, J.; Yu, J.; Han, K.; Li, M.; Jin, X.; Yin, H. YOLO-CornSeg: A Lightweight Segmentation Model for Corn Seedlings with an Indirect Weed Detection Strategy. *Agronomy* **2026**, *16*, 1091. [CrossRef]
21. Zhang, H.; Ge, Y.; Xia, H.; Sun, C. Safflower Picking Points Localization Method during the Full Harvest Period Based on SBP-YOLOv8s-Seg Network. *Comput. Electron. Agric.* **2024**, *227*, 109646. [CrossRef]
22. Yang, J.; Li, X.; Wang, X.; Fu, L.; Li, S. Vision-Based Localization Method for Picking Points in Tea-Harvesting Robots. *Sensors* **2024**, *24*, 6777. [CrossRef] [PubMed]
23. Li, H.; Yin, Z.; Zuo, Z.; Pan, L.; Zhang, J. Precision Citrus Segmentation and Stem Picking Point Localization Using Improved YOLOv8n-Seg Algorithm. *Front. Plant Sci.* **2025**, *16*, 1655093. [CrossRef] [PubMed]
24. Wang, C.; Ding, F.; Wang, Y.; Wu, R.; Yao, X.; Jiang, C.; Ling, L. Real-Time Detection and Instance Segmentation of Strawberry in Unstructured Environment. *Comput. Mater. Contin.* **2024**, *78*, 1481–1501. [CrossRef]
25. Xie, Y.; Chen, L. CBLN-YOLO: An Improved YOLO11n-Seg Network for Cotton Topping in Fields. *Agronomy* **2025**, *15*, 996. [CrossRef]
26. Cardellicchio, A.; Renò, V.; Cellini, F.; Summerer, S.; Petrozza, A.; Milella, A. Incremental Learning with Domain Adaption for Tomato Plant Phenotyping. *Smart Agric. Technol.* **2025**, *12*, 101324. [CrossRef]
27. Khan, A.T.; Jensen, S.M. LEAF-Net: A Unified Framework for Leaf Extraction and Analysis in Multi-Crop Phenotyping Using YOLOv11. *Agriculture* **2025**, *15*, 196. [CrossRef]
28. Niu, J.; Bi, M.; Yu, Q. Apple Pose Estimation Based on SCH-YOLO11s Segmentation. *Agronomy* **2025**, *15*, 900. [CrossRef]
29. Luo, R.; Zhao, R.; Yi, B. Enhanced YOLO11n-Seg with Attention Mechanism and Geometric Metric Optimization for Instance Segmentation of Ripe Blueberries in Complex Greenhouse Environments. *Agriculture* **2025**, *15*, 1697. [CrossRef]
30. Sapkota, R.; Ahmed, D.; Karkee, M. Comparing YOLOv8 and Mask R-CNN for Instance Segmentation in Complex Orchard Environments. *Artif. Intell. Agric.* **2024**, *13*, 84–99. [CrossRef]
31. Liu, Q.; Lv, J.; Zhang, C. MAE-YOLOv8-Based Small Object Detection of Green Crisp Plum in Real Complex Orchard Environments. *Comput. Electron. Agric.* **2024**, *226*, 109458. [CrossRef]

32. Liu, M.; Jia, W.; Wang, Z.; Niu, Y.; Yang, X.; Ruan, C. An Accurate Detection and Segmentation Model of Obscured Green Fruits. *Comput. Electron. Agric.* **2022**, *197*, 106984. [CrossRef]
33. Wang, D.; He, D. Apple Detection and Instance Segmentation in Natural Environments Using an Improved Mask Scoring R-CNN Model. *Front. Plant Sci.* **2022**, *13*, 1016470. [CrossRef] [PubMed]
34. Chen, L.; Fu, Y.; Gu, L.; Yan, C.; Harada, T.; Huang, G. Frequency-Aware Feature Fusion for Dense Image Prediction. *IEEE Trans. Pattern Anal. Mach. Intell.* **2024**, *46*, 10763–10780. [CrossRef] [PubMed]
35. Li, Y.; Hou, Q.; Zheng, Z.; Cheng, M.-M.; Yang, J.; Li, X. Large Selective Kernel Network for Remote Sensing Object Detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 1–6 October 2023; pp. 16748–16759. [CrossRef]
36. Bolya, D.; Zhou, C.; Xiao, F.; Lee, Y.J. YOLACT++: Better Real-Time Instance Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 1108–1121. [CrossRef] [PubMed]