Multi-source feature fusion network for grape berry instance segmentation

Mingcheng Yao[1†], Xiaoxia Yang[1†], Chengming Zhang[1*], Menxin Wu[2], Jinlong Fan[3], Feng Li[4*], Hongying Li[5], Jianguo Wei[6], Shujun Liu[1], Huake Zhang[1], Yanhui Jin[1]

(1. *College of Information Science and Engineering, Shandong Agricultural University, Tai'an 271018, Shandong, China*;
2. *The National Meteorological Center, Beijing 100081, China*;
3. *Faculty of Geographical Sciences, Beijing Normal University, Beijing 100875, China*;
4. *Shandong Provincial Climate Center, Jinan 250031, China*;
5. *Ningxia Institute of Meteorological Sciences, Yinchuan, Ningxia 750002, China*;
6. *Key Laboratory of Meteorological Disaster Monitoring, Early Warning, and Risk Management for Characteristic Agriculture in Arid Regions, China Meteorological Administration, Yinchuan, Ningxia 750002, China*)

Abstract: Accurate delineation of grape berry boundaries is essential for phenotypic measurement and growth assessment. This study proposes a multi-source feature fusion network (MFFNet) for instance segmentation of grape berries in dense clusters with frequent overlaps and blurred edges. MFFNet employs two parallel branches for feature extraction: a Swin Transformer backbone to capture hierarchical semantic features and an edge-detection branch that predicts an edge probability map to provide boundary cues. To address the substantial scale variation within a single image, the multilevel semantic features were enhanced using Adaptive Spatial Feature Fusion (ASFF). The edge probability map was introduced twice into the ASFF-enhanced multi-scale features. First, edge cues were injected into the highest-resolution fused feature map to strengthen global boundary awareness across the cluster. Second, during mask generation, edge cues were reintroduced within each candidate instance region to refine local contours and improve the separation in the adhered areas. Experiments on a custom dataset collected in Yinchuan, Ningxia, showed that MFFNet achieved an mAP_{50}^{box} of 93.4% and mAP_{50}^{mask} of 93.4%, outperforming representative baselines, including Mask2Former and HTC. The proposed model remained stable on images with severe berry overlap and indistinct edges, supporting practical grape growth monitoring.

Keywords: grape berry, instance segmentation, edge detection, multi-source feature fusion, precision agriculture
DOI: 10.25165/j.ijabe.20261902.10132

Citation: Yao M C, Yang X X, Zhang C M, Wu M X, Fan J L, Li F, et al. Multi-source feature fusion network for grape berry instance segmentation. Int J Agric & Biol Eng, 2026; 19(2): 294–302.

1 Introduction

Accurate morphological information of grape berries is essential for fine-grained growth monitoring, management, and phenotypic measurement[1,2]. Computer vision-based berry segmentation has become an important research direction in agricultural automation as it overcomes the inefficiency and subjectivity of manual observation and meets large-scale production requirements for automated and standardized phenotyping[3,4]. Although notable progress has been achieved under sparse fruit or relatively simple background conditions, real in-field vineyards exhibit dense clustering, leaf occlusion, strong illumination variations, and cluttered backgrounds, which hinder existing methods from precisely segmenting individual berries while preserving their complete outlines[5]. Early studies mainly relied on handcrafted features combined with traditional machine learning or classical image processing for fruit recognition and segmentation[6-10]. However, in practical vineyard scenarios with complex backgrounds, weak contrast, and adhered berries, such approaches lack robustness and thus cannot support high-precision morphological analysis[11,12].

With the development of deep learning, convolutional neural network (CNN)-based visual perception methods have been widely adopted for agricultural applications. Detection frameworks such as the YOLO family[13-15], SSD[16-18], and Faster R-CNN[19-22] have achieved strong performance in fruit localization tasks, and classic instance segmentation methods, such as Mask R-CNN[23], have also demonstrated high accuracy for pixel-level segmentation of crop fruits[24-27]. However, for grape berry clusters characterized by dense stacking, closely adjacent boundaries, and severe occlusion, CNNs are constrained by their limited receptive fields and insufficient long-range dependency modeling. Consequently, they often suffer from adhesion between neighboring berries, missed detection of small berries, and degraded contour details, leaving substantial scope for improvement[28,29].

To alleviate the limitations of CNNs in global modeling, Transformer-based vision models have attracted increasing attention owing to the global context modeling enabled by self-attention.

Received date: 2025-08-25 **Accepted date:** 2026-03-05
Biographies: **Mingcheng Yao**, Master candidate, research interest: image processing, Email: yaomingcheng@qq.com; **Xiaoxia Yang**, Master, research interest: intelligent agriculture, Email: yangxx@sdau.edu.cn; **Menxin Wu**, PhD, Email: wumx@cma.gov.cn; **Jinlong Fan**, PhD, Email: fanjl@bnu.edu.cn; **Hongying Li**, PhD, Email: hongyinglhy@126.com; **Jianguo Wei**, PhD, Email: weijian0269@163.com; **Shujun Liu**, Master candidate, Email: 19853838025@163.com; **Huake Zhang**, Master candidate, Email: 15621888796@163.com; **Yanhui Jin**, Master candidate, Email: 13623933329@163.com.
†These authors contributed equally to this work.
***Corresponding author: Chengming Zhang**, Professor, Research interest: intelligent agriculture, College of Information Science and Engineering, Shandong Agricultural University, Tai'an 271018, Shandong, China. Tel: +86-13953823659, Email: chming@sdau.edu.cn; **Feng Li**, Positive Senior Engineer, Research interest: Intelligent remote sensing. Shandong Provincial Climate Center, Jinan 250031, China. Tel: +86-15666976552, Email: lfeng1029@163.com.

Architectures such as ViT and Swin Transformer efficiently encode long-range dependencies and multi-scale semantic information by modeling the relationships between image patches or local windows[30,31]. Building on this foundation, Transformers have been introduced into detection and segmentation tasks, and CNN–Transformer hybrid designs have been explored to combine the local texture extraction of CNNs with the global context modeling of Transformers, thereby improving the perception and segmentation of morphologically diverse and densely overlapped targets[32-38]. However, for complex grape cluster scenes, systematic designs that coordinate local detail cues with global semantics, in particular to jointly enhance instance separability and contour fidelity, remain underexplored.

For grape clusters containing numerous small berries with tightly connected boundaries, which frequently lead to adhesion and missed instances[39], explicitly incorporating boundary information has proven to be effective in improving instance separability and contour accuracy[40-42]. Prior studies have enhanced boundary responses via edge-guidance modules[43,44] or constrained object contours through multitask learning[45,46]; some studies further verified that such strategies improved the mask boundary quality under complex backgrounds[47]. Berries within the same image often exhibit significant scale variations, rendering multiscale feature fusion essential for a robust performance on small and multiscale targets. Traditional feature pyramid networks (FPN) follow fixed fusion patterns and cannot adaptively adjust the contributions of features at different scales. By contrast, data-driven fusion weighting can improve sensitivity and representation capacity for small and densely distributed targets[48-50].

In summary, grape berry segmentation models require not only a global contextual understanding to distinguish between the background and the berry, but also a fine-grained local perception to characterize the berry outline and the ability to adaptively represent and fuse targets at different scales. Existing research often focuses on improving one aspect of global modeling, enhancing boundary information, or improving multiscale fusion. However, a multi-source feature fusion scheme specifically designed for the segmentation of grape berries, a typical example of dense occlusion, is required. This scheme should collaboratively integrate global semantics, explicit edges, and adaptive multiscale information within a unified framework.

To address the demand for high-precision berry segmentation in grape growth monitoring and nondestructive size measurements, we propose a multi-source feature fusion network (MFFNet) for instance segmentation. MFFNet leverages the hierarchical global semantic features extracted by a Swin Transformer backbone, introduces a parallel edge branch to generate explicit boundary cues, and employs Adaptive Spatial Feature Fusion (ASFF) for adaptive multiscale semantic fusion. Furthermore, a staged fusion strategy was designed to inject and exploit edge and multiscale information at different stages of the backbone and prediction head, thereby improving the accuracy and robustness of in-field grape berry instance segmentation. The main contributions of this study are as follows: 1) a parallel semantic-edge feature extraction architecture that preserves strong semantics while explicitly maintaining high-frequency boundary cues; 2) an ASFF-enhanced multi-scale semantic fusion mechanism to improve robustness to strong scale variations and small, dense targets; 3) a staged edge fusion strategy that strengthens global boundary awareness and refines RoI-level local contours, improving instance separation and contour fidelity for adhered berries; and 4) systematic experiments and ablation studies on a challenging in-field grape dataset to validate the effectiveness of each component.

2 Materials and methods

2.1 Study area and dataset

The eastern foothills of the Helan Mountains in Ningxia (Figure 1) constitute the premier wine-growing region in China[51]. Characterized by a temperate arid climate, abundant sunshine, and a stable diurnal temperature range during maturation, this area is ideally suited for high-quality grape production[52].

The image data for this study were collected at the Lilan Winery in Minning Town, Yinchuan, located in the core of the Helan Mountain wine region. This winery cultivates representative local varieties, such as Cabernet Sauvignon and Merlot, on a 100 hm² high-standard organic cultivation base. This base employs a standardized layout with 3-m row spacing, 0.5-m plant spacing, and a unilateral cordon trellis system, with production strictly limited to 4500 kg/hm². This fine-grained cultivation model provides an ideal and representative experimental site for this study.

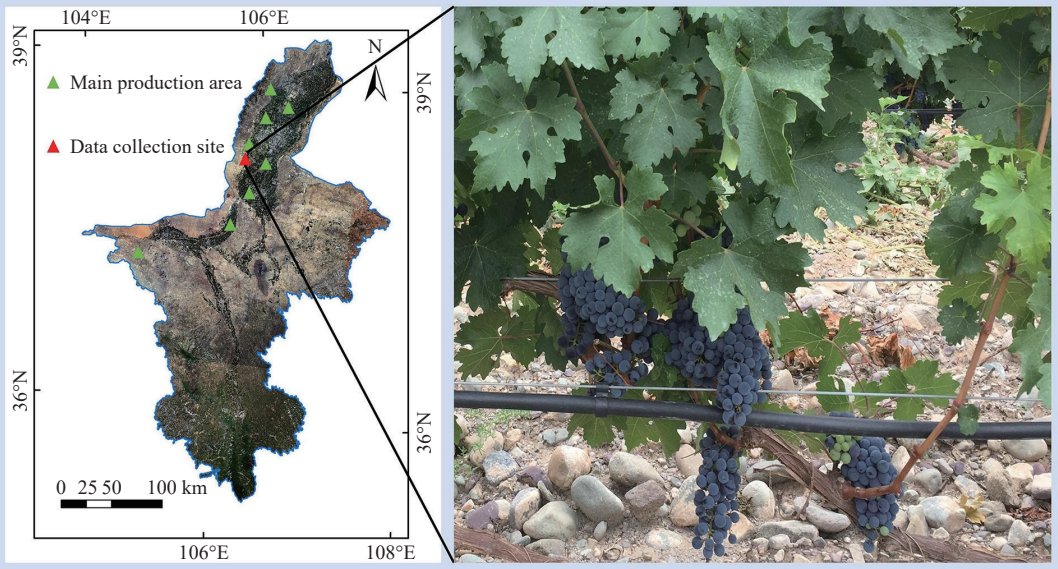

Figure 1 Main grape-producing regions in China

Image acquisition took place under clear weather conditions, focusing on the fruit maturation periods in 2024 and 2025. Images were captured using a Sony α6400 mirrorless camera equipped with a Sigma 30 mm F1.4 DC DN prime lens. All photographs were taken in manual mode, yielding a collection of 4000 high-resolution RGB images. To ensure that the dataset reflected real-world challenges, we captured images from multiple angles, encompassing complex backgrounds and varying degrees of occlusion under natural field conditions.

After screening the raw images to remove blurry or overexposed samples, 3600 high-quality images were selected to build the dataset. The contour of each visible grape berry was annotated using LabelMe. The dataset was randomly divided into training (2800 images; 130 534 instances), validation (400 images; 18 171 instances), and test (400 images; 17 964 instances) sets. All images were resized to 800×800 pixels to balance accuracy and efficiency. Offline augmentation was applied on the training set in this study, including horizontal/vertical flipping, color jittering (brightness±50%, contrast±50%, hue±20%), additive Gaussian noise (μ=0, σ=50), and random rotation (±25°), expanding the training set to 22 400 images.

2.2 Structure of MFFNet

The MFFNet architecture (Figure 2) comprises three main parts: 1) a parallel feature extraction module with an edge detection branch and a Swin Transformer backbone; 2) a feature fusion module that integrates multiscale semantic features with edge information; and 3) a segmentation head including the region proposal network, RoI alignment, and parallel prediction branches, where a fully convolutional network serves as the mask generation module and fully connected layers handle classification and box regression. The left panel of Figure 2 shows the data collection process and sample visualization, the middle panel presents the network architecture with detailed module interactions, and the right panel displays the instance segmentation results with detailed mask boundaries.

2.2.1 Parallel feature extraction module

The segmentation process is initiated using a dual-stream parallel feature extraction module designed to concurrently collect complementary visual cues. For this, the module comprises two distinct branches that operate in parallel on the input image: a semantic path and an edge detection branch. The semantic path utilizes a Swin Transformer backbone to capture the global context. Simultaneously, the independent edge detection branch employs an encoder–decoder architecture to generate an edge probability map, providing essential explicit boundary information.

The semantic path uses the Swin Transformer as the backbone network and processes the input RGB image through a hierarchical architecture (Figure 3). The image is initially partitioned into non-overlapping patches and mapped using linear embedding. The network is organized into four stages with Swin Transformer block counts of {2, 2, 4, 2}, respectively. Patch-merging layers positioned between the stages progressively halve the spatial resolution while doubling the channel dimensions, yielding a multiscale feature pyramid $\{F_1, F_2, F_3, F_4\}$. This hierarchical set effectively encapsulates information ranging from fine-grained textures to high-level semantic representations of grape clusters.

Each Swin Transformer block is composed of two successive sublayers (Figure 4). For an input feature map Z^{l-1} to the l-th block, the process begins with layer normalization (LN). The first sub-layer applies a windowed multi-head self-attention (W-MSA) mechanism that partitions the feature map into non-overlapping windows and computes self-attention independently within each. The output is combined with the input Z^{l-1} via a residual connection. This result is passed through another normalization layer and a multi-layer perceptron (MLP) with a final residual connection, producing the output feature map Z^l, as expressed in Equations (1) and (2).

$$\hat{Z}^l = W - \mathrm{MSA}(\mathrm{LN}(Z^{l-1})) + Z^{l-1} \quad (1)$$

$$Z^l = \mathrm{MLP}(\mathrm{LN}(\hat{Z}^l)) + \hat{Z}^l. \quad (2)$$

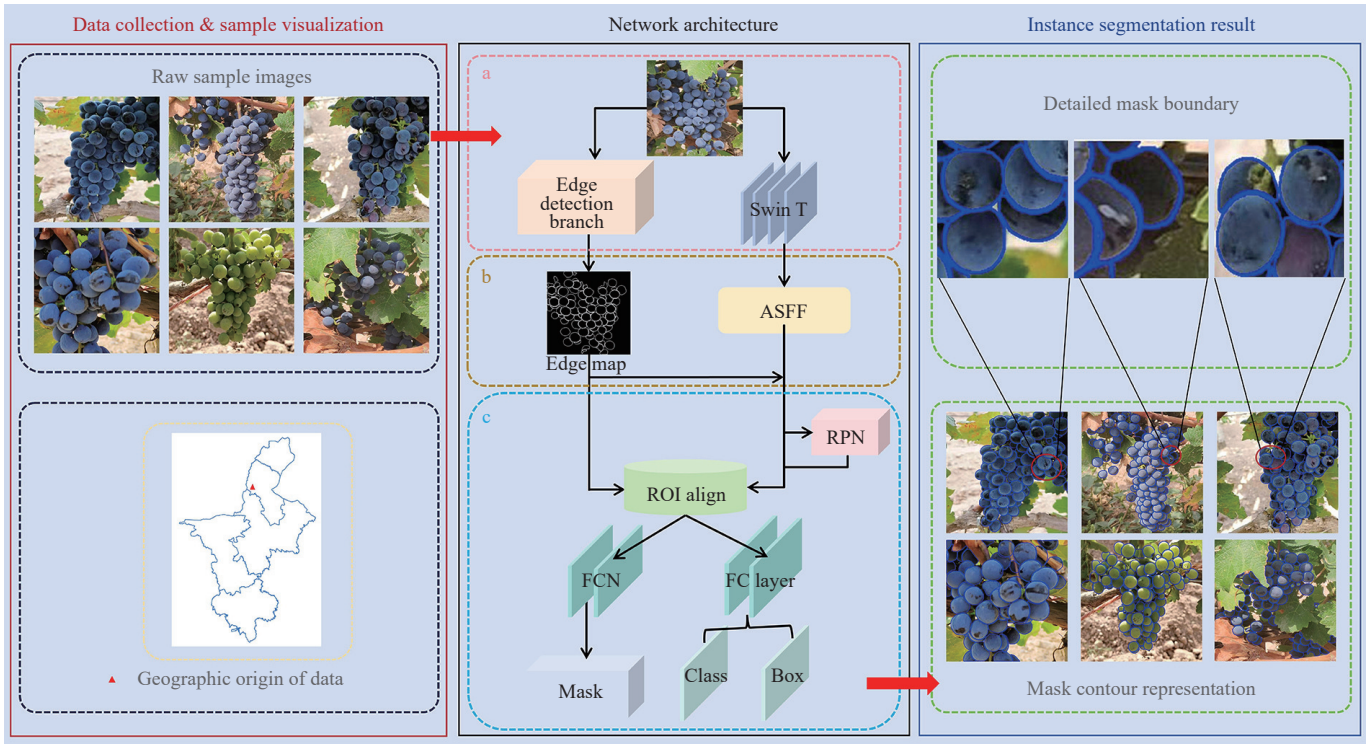

Figure 2 Overview of MFFNet

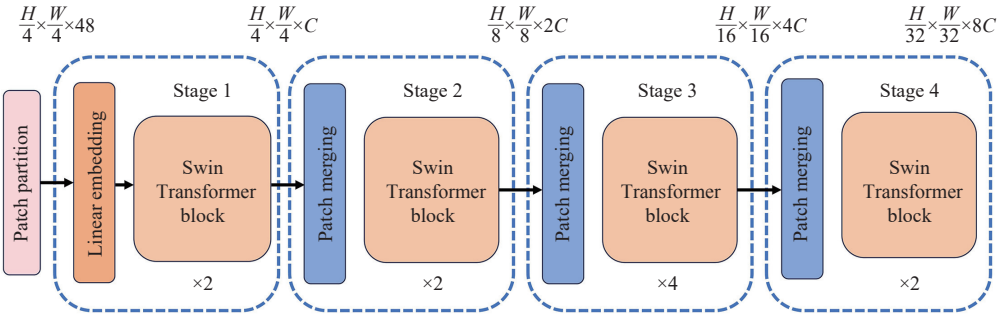

Figure 3 Architecture of the Swin Transformer

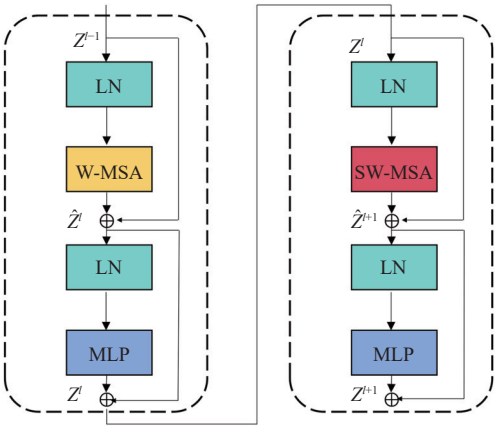

Figure 4 Structure of the Swin Transformer block

The feature map Z^l is then processed by the second sublayer. This sub-layer follows a similar structure but replaces the W-MSA with a shifted window multi-head self-attention (SW-MSA) mechanism. Before partitioning the map, the SW-MSA applies a cyclic shift to it, ensuring that patches from different windows in the previous layer are grouped in the new windows. Self-attention is computed within these new shifted windows. This process enables information exchange across window boundaries, which is crucial for modeling the global context and accurately distinguishing grape clusters from background elements, such as leaves and stems. The operations of the second sublayer are given in Equations (3) and (4).

$$\hat{Z}^{l+1} = \mathrm{SW-MSA}(\mathrm{LN}(Z^l)) + Z^l \tag{3}$$

$$Z^{l+1} = \mathrm{MLP}(\mathrm{LN}(\hat{Z}^{l+1})) + \hat{Z}^{l+1} \tag{4}$$

By alternating between the W-MSA and SW-MSA, the architecture efficiently captures the global context while maintaining computational viability.

To address challenges such as blurred contours and adhesion between berries, the auxiliary edge detection branch extracts explicit boundary information (Figure 5). This branch utilizes a ResNet-34 encoder coupled with a decoder to produce a high-resolution EdgeMap (E_{map}), where the pixel values represent the probability of belonging to a berry contour. Unlike the semantic path, this branch employs a CNN-based ResNet-34 encoder, rather than sharing the Transformer backbone. Although Transformers excel at modeling long-range semantic contexts, accurate edge localization depends heavily on local textures and low-level cues. ResNet-34 is inherently suitable for capturing high-frequency details. Decoupling the encoders preserves fine boundary information that might otherwise be attenuated by the global pooling operations of the Transformer.

The encoder progressively reduces the resolution while increasing the channel depth, thereby yielding feature maps $\{E_1, E_2, E_3, E_4\}$. To capture the multiscale context without further reducing the spatial resolution, we apply a dilated convolution block to the output of the final encoder layer, E_4. This module consists of four dilated convolution layers with the dilation rates set to {1, 2, 4, 8}, respectively. The module also features a residual design, in which the outputs of the four dilated layers $\{A_1, A_2, A_3, A_4\}$ are combined with the original input E_4 via element-wise addition to produce the final output. The generation of the intermediate features A_k is expressed in Equation (5).

$$A_k = ReLU\left(BN\left(Conv_{3\times3}, dilation = 2^{k-1}(A_{k-1})\right)\right), \tag{5}$$

where A_0=E_4, k = 1, 2, 3, 4, and the dilation rate is set to 2^{k-1}, corresponding to 1, 2, 4, and 8 for the four dilated convolution layers, respectively.

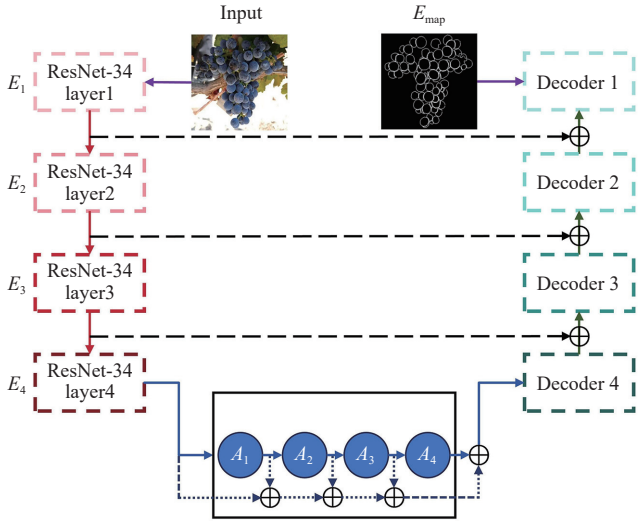

Figure 5 Structure of the edge detection branch

Corresponding to the encoder, the decoder consists of four decoding units. The input of each decoding unit is the sum of the outputs of the previous decoding unit and the corresponding encoding unit. Each decoding unit first uses a 1×1 convolution to reduce its input channel count to one-quarter of its original value, and then uses a 3×3 transposed convolution to upsample the features by a factor of two. The last convolutional layer generates a single-channel edge probability map E_{map}, whose calculation process is shown in Equation (6).

$$O_k = D_k(O_{k+1}) + \Pi(k > 1)E_k(k{=}1,2,3,4), \tag{6}$$

where $\Pi(k > 1)$ is an indicator function that equals 1 when $k > 1$ and 0 otherwise. O_k denotes the output of the k-th decoder, D_k denotes the operation of the k-th decoder, and E_k denotes the feature map output of the k-th encoder stage.

2.2.2 Adaptive spatial feature fusion module

Grape berries often exhibit diverse scales and are subject to mutual occlusion, requiring the model to have a powerful multiscale feature representation capability to identify berries of various sizes accurately. Introducing the ASFF module (Figure 6) enables the learning of adaptive fusion weights, facilitating more effective utilization of multiscale feature maps. This generates more discriminative fused features $\{Y_1, Y_2, Y_3, Y_4\}$ that are more sensitive to grape berries of different scales.

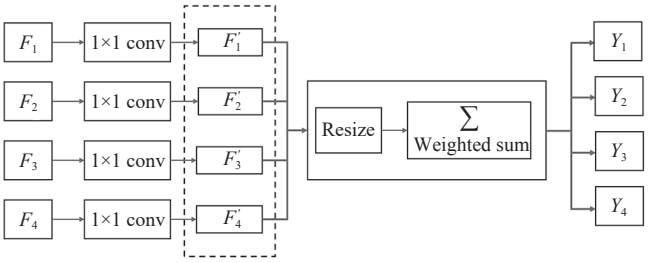

Figure 6 Structure of the ASFF module

The ASFF process begins by obtaining multiscale feature maps $\{F_1, F_2, F_3, F_4\}$ from the backbone. For each feature map F_i, a 1×1 convolution adjusts the channel count to a uniform dimension. Next, for the target fusion level i, every other feature map is resized to match the spatial dimensions of the target level using bilinear interpolation. This yields a set of spatially aligned feature maps, $R_{l\to i}$. The module introduces a set of learnable weight parameters $\alpha_i=\{\alpha_{i1}, \alpha_{i2}, \alpha_{i3}, \alpha_{i4}\}$, representing the weight vector used to generate the fused feature map Y_i at level i. The weight vector α_i is normalized via a Softmax function to obtain $w_i = Softmax(\alpha_i) = \mathrm{w}_{i1}, w_{i2}, w_{i3}, w_{i4}\}$, where w_{il} is the normalized weight for level l and for a fixed i. The calculation process is expressed in Equation (7).

$$Y_i = \sum_{l=1}^{4} w_{il} \times R_{l\to i} \tag{7}$$

The four fused feature maps output by the ASFF module integrate adaptively weighted information from all input levels. They maintain the same spatial resolution as that of the original corresponding levels, providing a high-quality multiscale feature foundation for subsequent edge fusion and instance prediction stages.

2.2.3 Segmentation head with edge fusion

Following feature extraction and fusion, the enriched feature maps are passed to the segmentation head for the final instance prediction.

To embed the global boundary constraints into the prediction pipeline, the edge probability map from the edge detection branch is fused with the ASFF-enhanced feature maps. The EdgeMap is first processed using a 3×3 convolution to align its channel dimensions and extract features, resulting in E_p. To avoid down-sampling and preserve sharp boundary details, E_p is fused exclusively with Y_1, which is the feature map with the highest resolution. Fusion is achieved by concatenating Y_1 and E_p along the channel axis, followed by a 1×1 convolution to integrate the information and restore the original channel count. This process yields the edge-enhanced feature map Y_1', as expressed in Equation (8). The original feature map Y_1 is then replaced by the enhanced Y_1', forming the final feature set $\{Y_1', Y_2, Y_3, Y_4\}$. This early fusion sharpens the berry contours within backbone features, thereby providing a robust foundation for subsequent prediction tasks.

$$Y_1' = \mathrm{Conv}_{1\times1}\left(\mathrm{Concat}([Y_1, E_p])\right) \tag{8}$$

The segmentation prediction head consists of the region proposal network (RPN), multi-scale RoI alignment, edge RoI alignment, bounding box detection branch, and mask prediction branch. The RPN generates candidate boxes that may contain grape berries from the input feature maps. It configures the anchor boxes with base sizes of 8×8, 16×16, 32×32, and 64×64 on four feature maps of different sizes. According to the actual conditions of the grape berries, three different aspect ratios (1:1, 1:2, and 2:1) are applied to cover diverse berry morphologies. A convolutional layer then predicts the objectness confidence and bounding-box regression offsets for each anchor box. After threshold filtering and non-maximum suppression (NMS) processing, high-quality candidate boxes are generated.

The multiscale RoI alignment receives the candidate boxes output by the RPN. For each candidate box, it selects an appropriate scale from the feature pyramid and performs RoI alignment to obtain instance-aligned semantic features, denoted as $\mathrm{M_{roi}}$. These features are fed into the bounding box detection branch, which consists of a classifier and a regressor, both of which use fully connected layers. The flattened RoI features are used to predict class probabilities and refine the bounding box regression offsets, ultimately outputting the target class and bounding box.

Simultaneously, the edge probability map E_{map} output from the edge detection module, together with the candidate boxes, is input into the edge RoI alignment module, which extracts the instance boundary-aligned edge features E_{roi}. Edge priors are introduced into mask prediction by concatenating M_{roi} and E_{roi} along the channel dimension and fused through a 1×1 convolution to generate edge-refined features M'_{roi}. The fusion process is expressed by Equation (9).

$$M'_{\mathrm{roi}} = \mathrm{Conv}_{1\times1}\left(\mathrm{Concat}([E_{\mathrm{roi}}, M_{\mathrm{roi}}])\right), \tag{9}$$

where M'_{roi} is fed into the mask prediction branch, which employs a fully convolutional network structure to generate pixel-wise binary masks for grape berries and restores them to the original image resolution through upsampling. Ultimately, the model determines the detection results based on the classification confidence and regressed bounding boxes. It simultaneously outputs segmentation masks corresponding to each detection box and obtains fine-grained grape berry instance segmentation results.

2.2.4 Loss function

The loss function consists of instance segmentation and boundary auxiliary losses, as expressed in Equation (10).

$$L_{\mathrm{total}} = L_{\mathrm{RPN}} + L_{\mathrm{cls}} + L_{\mathrm{reg}} + L_{\mathrm{mask}} + \lambda_b L_b \tag{10}$$

where L_{total}, L_{RPN}, L_{cls}, L_{reg}, L_{mask}, and L_b represent the total, RPN, classification, bounding box regression, mask segmentation, and boundary auxiliary loss, respectively. λ_b is the weight parameter indicating the proportion of boundary auxiliary loss in the total loss. Its value is determined through comparative experiments using a validation dataset. Candidate values {0.05, 0.10, 0.20, 0.50, 1.00} were evaluated on the validation set, and the optimal value was selected; consequently, λ_b was set to 0.2.

The calculation formula for L_{RPN} is expressed in Equation (11).

$$L_{\mathrm{RPN}} = -\frac{1}{N_a}\sum_{i=1}^{N_a}\left[y_i \cdot \log(p_i) + (1-y_i)\cdot\log(1-p_i)\right] + \lambda_{\mathrm{rpn\_reg}}\frac{1}{N_{\mathrm{pos}}}\sum_{i\in\rho_a} smooth_{L1}(t_i - t_i^*) \tag{11}$$

where N_a is the total number of anchor boxes, y_i is the ground-truth label of the i-th anchor, p_i is the model's predicted probability that the i-th anchor is a grape berry, ρ_a is the set of positive (grape berry) anchor boxes, N_{pos} is the total number of positive anchor boxes, and t_i and t_i^* are the predicted and ground-truth parameters for the i-th anchor box, respectively.

The formula for calculating L_{cls} is presented in Equation (12).

$$L_{\text{cls}} = -\frac{1}{M_r}\sum_{j=1}^{M_r}\log\left(p_{j,c_j}\right), \tag{12}$$

where M_r is the total number of RoIs and p_{j,c_j} is the predicted probability of the j-th RoI belonging to its ground-truth class c_j.

The formula for calculating L_{reg} is presented in Equation (13).

$$L_{reg} = \frac{1}{M_{pos}}\sum_{j\in\rho_{roi}}\sum_{k=1}^{4} smooth_{L1}\left(t_{j,k} - t_{j,k}^*\right), \tag{13}$$

where ρ_{roi} is the set of positive RoIs, M_{pos} is the number of positive (grape berry) RoIs, and $t_{j,k}$ and $t_{j,k}^*$ are the predicted and ground-truth values for the k-th bounding box parameter of the j-th RoI, respectively.

The formula for calculating L_{mask} is expressed in Equation (14).

$$L_{\text{mask}} = -\frac{1}{M_{\text{pos}}}\sum_{j\in\rho_{\text{roi}}}\frac{1}{H\times W}\sum_{h=1}^{H}\sum_{w=1}^{W}\left[m_{j,h,w}^*\cdot\log\left(m_{j,h,w}\right) + \left(1-m_{j,h,w}^*\right)\cdot\log\left(1-m_{j,h,w}\right)\right] \tag{14}$$

where $m_{j,h,w}$ and $m_{j,h,w}^*$ are the predicted and ground-truth mask values, respectively, at position (h, w) for the j-th RoI.

The formula for calculating L_b is expressed in Equation (15).

$$L_b = -\frac{1}{N}\sum_{i=1}^{N}\left[w_p\cdot q_i^*\cdot\log(q_i) + w_n\left(1-q_i^*\right)\cdot\log(1-q_i)\right] \tag{15}$$

Here, N is the total number of pixels, q_i and q_i^* are the predicted edge probability and ground-truth label for the i-th pixel, respectively, and w_p and w_n are the weights for the positive and negative samples, respectively. Generally, w_p=0.9 and w_n=0.1. The ground-truth edge map is generated by extracting contours from the polygon annotations in the labeling files.

3 Results and Discussion

3.1 Experimental setup

All the models were implemented in PyTorch and trained on a workstation with an Intel Xeon Silver 4210R CPU and an NVIDIA GeForce RTX 3090 GPU. We used AdamW with an initial learning rate of 0.0001. Automatic mixed precision (AMP) was used to accelerate training and reduce GPU memory usage. The training stability was improved by applying gradient clipping with a maximum L2 norm of 5.0.

This study compared MFFNet with representative instance segmentation baselines covering different paradigms: Mask R-CNN as a widely used two-stage benchmark, Cascade Mask R-CNN and HTC as stronger cascaded variants, SOLOv2 as a single-stage, anchor-free method designed for dense scenes, and Mask2Former as a recent Transformer-based mask-classification framework. This set of baselines enabled us to evaluate MFFNet against both classic CNN pipelines and the newer Transformer-style segmentation models.

3.2 Evaluation metrics

The standard mean average precision (mAP) metric was used. The mAP was reported for both bounding box detection and mask segmentation, calculated across a range of intersection over union (IoU) thresholds.

Average precision (AP) represents the area under the precision–recall (P-R) curve. The final mAP score is the mean of the AP values calculated across all object classes. The definitions are expressed in Equations (16) and (17).

$$\text{AP} = \int_0^1 P\text{d}R \tag{16}$$

$$\text{mAP} = \frac{1}{N_{\text{classes}}}\sum_{i=1}^{N_{\text{classes}}}\text{AP}_i \tag{17}$$

We evaluated MFFNet against baseline models on our test set. Following the standard COCO evaluation protocols, we reported the mAP for both the bounding boxes (mAP^{box}) and segmentation masks (mAP^{mask}). We also determined the primary mAP (averaged over IoU thresholds from 0.50 to 0.95) along with the AP_{50} and AP_{75}. The detailed results are listed in Table 1, with the visual representations displayed in Figure 7.

Figure 7 Performance comparison of different instance segmentation models on grapes

Table 1 quantitatively compares the performance of the models on the full test set. MFFNet achieved an mAP^{box} of 66.5% and an mAP^{mask} of 67.9%. Its $\text{mAP}_{75}^{\text{mask}}$ reached 79.9%, indicating a clear advantage when strict contour alignment is required. In Table 1, "-" indicates that the method did not output bounding boxes; thus, the box metrics are not reported.

Among the two-stage baselines, HTC performed the best with an mAP^{mask} of 63.3% and an $\text{mAP}_{75}^{\text{mask}}$ of 73.4%. Its cascaded design helps to refine masks, but boundaries can still become ambiguous when the berries are tightly adhered. Cascade Mask R-

CNN shows a similar pattern; its mAP^{mask} reached 63.0%, benefiting from stronger localization but remaining limited to fine boundary delineation. Mask R-CNN yielded the lowest performance among the two-stage methods; the low performance is attributed to its reliance on convolutional features and a fixed-pyramid fusion scheme when facing dense, small, and visually similar berries.

Table 1 Performance comparison with other models (%)

Model	mAP^{box}	mAP^{box}_{50}	mAP^{box}_{75}	mAP^{mask}	mAP^{mask}_{50}	mAP^{mask}_{75}
Mask R-CNN	60.3	91.0	70.5	61.0	90.1	70.7
Cascade Mask R-CNN	61.9	91.8	72.2	63.0	89.9	72.1
HTC	63.8	92.3	73.6	63.3	90.5	73.4
SOLOv2	-	-	-	58.8	86.3	67.6
Mask2Former	-	-	-	62.7	91.1	73.3
MFFNet	66.5	93.4	79.6	67.9	93.4	79.9

SOLOv2 exhibited lower performance on this dataset with an mAP^{mask} of 58.8%. In dense clusters, multiple berries can fall into the same grid region, rendering the instance assignment unstable and increasing the number of missed detections. Mask2Former obtained a competitive mAP^{mask}_{50} of 91.1%, but its overall mAP^{mask} reached 62.7%, which was below that of HTC. This suggests that query-based matching becomes less reliable when instances heavily overlap and edges are indistinct. Overall, the results indicate that MFFNet is more robust in separating adjacent berries and maintaining tight contours, which is also reflected in the larger margin of mAP^{mask}_{75}.

The results indicate that the performance gains of MFFNet were most pronounced at the stricter mAP_{75} metric. This improvement is directly attributed to the fusion of edge information. The fusion of edge information provides strong geometric priors that guide contour delineation, thereby directly improving the fidelity of the final masks. High mask fidelity is critical for achieving high IoU scores. Accurate boundary separation reduces the errors for adhered berries, leading to more true positives under strict precision requirements. Furthermore, for the irregular morphology of grape berries, a precise mask is a more accurate representation of the true size and shape than a simple bounding box.

The effectiveness of MFFNet stems from its complementary integration of semantic features, multiscale information, and edge priors. The Swin Transformer backbone network captures the overall structure of the grape berries and distinguishes them from the background by modeling long-range dependencies. The ASFF module handles scale variations through adaptive fusion, enabling the model to match the most suitable feature representations for berries of different sizes. The edge detection branch introduces geometric constraints, providing independent boundary signals to guide semantic features and achieve accurate contour separation in closely packed regions.

3.3 Performance on heavily occluded scenes

We further evaluated all the methods on a challenging subset of 100 test images with dense clusters, severe occlusion, and leaf interference (Table 2). Compared with the results in Table 1, the performance of all models decreased, although the magnitude of degradation varied across methods. MFFNet remained the top performer, with an mAP^{mask}_{50} of 86.9% and an mAP^{mask}_{75} of 72.6%; it showed the smallest decrease in mAP^{mask}_{50} among all the compared methods.

Mask2Former exhibited the largest performance drop on this subset, where mAP^{mask}_{50} decreased from 91.1% to 76.8%. In highly cluttered scenes, where berries have similar appearances and the boundaries are weak, query-to-instance assignments are more likely to confuse adjacent instances, leading to missed or merged masks.

Table 2 Performance of different models in complex scenarios (%)

Model	mAP^{box}	mAP^{box}_{50}	mAP^{box}_{75}	mAP^{mask}	mAP^{mask}_{50}	mAP^{mask}_{75}
Mask R-CNN	57.3	81.3	66.5	58.8	78.1	67.4
Cascade Mask R-CNN	58.1	79.0	69.2	58.4	79.5	68.0
HTC	62.4	80.8	70.9	62.0	80.9	71.5
SOLOv2	-	-	-	56.8	75.3	65.7
Mask2Former	-	-	-	53.7	76.8	66.2
MFFNet	63.8	85.3	72.0	63.5	86.9	72.6

HTC remained relatively stable and ranked second on this subset, but its masks still showed imperfect boundary fitting when the berries overlapped heavily (Figure 8). Cascade Mask R-CNN dropped more noticeably, and Figure 8 shows that it often produced fragmented masks with irregular contours on partially occluded berries. Mask R-CNN and SOLOv2 exhibited clearer limitations under this setting; therefore, we omitted them from the detailed visualization in Figure 8 and focused on more competitive methods. On this subset, MFFNet showed a larger margin over HTC in mAP^{mask}_{50} than on the full test set. As the occlusion increased and boundaries became less distinct, MFFNet maintained clearer instance separation and tighter contours.

MFFNet Cascade mask R-CNN HTC Mask2Former

Figure 8 Model performance in complex contexts

Overall, the results on the heavily occluded subset suggested that MFFNet was less sensitive to overlap and boundary ambiguity. This robustness is attributed to its joint use of hierarchical semantics, scale-aware features, and explicit boundary cues, which together reduce instance merging and preserve contour fidelity in cluttered clusters.

3.4 Ablation study

A series of ablation studies systematically validated the contributions of the key components of MFFNet, including its ASFF module, edge detection branch, auxiliary boundary loss (L_b), and staged feature fusion strategy. All the ablated models were trained using the same hyperparameters. The results are listed in Table 3.

Table 3 Ablation experiment results (%)

ASFF	L_b	E_{Fusion}	mAP^{box}	mAP^{box}_{50}	mAP^{box}_{75}	mAP^{mask}	mAP^{mask}_{50}	mAP^{mask}_{75}
×	×	×	61.8	91.7	73.1	62.3	90.3	72.0
×	√	√	62.5	92.0	75.5	62.9	91.3	74.4
√	×	×	62.3	91.8	72.6	62.4	90.8	73.3
√	√	×	63.6	91.9	73.7	62.5	91.4	74.2
√	√	√	66.5	93.4	79.6	67.9	93.4	79.9

The baseline coupled a Swin Transformer backbone with a standard prediction head and excluded the ASFF, auxiliary boundary loss L_b, and edge fusion E_{Fusion}. L_b represents using only

the edge loss function to supervise the model without feature fusion, while E_{Fusion} represents using feature fusion. The baseline achieved an mAP^{mask} of 62.3%. Adding ASFF improved the high-IoU mask quality, with $\text{mAP}^{\text{mask}}_{75}$ rising to 73.3%, as adaptive multi-scale re-weighting helped the model handle strong scale variation. Building on this, we added an edge branch with L_b but kept the edge map out of the main feature stream; $\text{mAP}^{\text{mask}}_{75}$ further increased to 74.2%, indicating that explicit boundary supervision promotes sharper contour learning. The largest gain was obtained when staged fusion was enabled, injecting edge cues into the highest-resolution backbone feature and RoI-aligned head features; $\text{mAP}^{\text{mask}}_{75}$ reached 79.9%. A control setting showed that fusing edges into the baseline without ASFF resulted in only limited improvement. This suggests that ASFF strengthens the scale-adaptive semantic foundation, whereas fused edge cues provide the geometric corrections that matter most when separating adhered or mutually occluded berries under strict IoU criteria.

4 Conclusions

This study addresses the challenge of extracting precise individual grape berry contours from field images where berries cluster densely with frequent occlusions and ambiguous boundaries. We propose MFFNet, which integrates semantic understanding with boundary guidance. The network employs separate encoders for semantic and edge feature extraction to preserve complementary information. Hierarchical features from the Swin Transformer capture the global context and instance-level semantics, while the parallel edge branch generates boundary probability maps that highlight contour locations. Adaptive multiscale feature fusion handles substantial size variations across berries in the same image. Edge information enters the segmentation pipeline at two stages to strengthen boundary delineation. MFFNet achieved competitive performance on standard metrics and showed clear advantages when evaluated under stricter IoU thresholds.

Future work will incorporate multimodal information such as depth and multiview data to alleviate 3D occlusion, design lightweight edge branches to reduce computational overhead, and explore self-supervised boundary learning to reduce reliance on high-quality contour annotations.

Acknowledgements

This research was funded by the Fengyun Satellite Application Pioneer Program (Phase Ⅲ) (Grant No. FY-APP-2024.0301), the Science Foundation of Shandong (Grant No. ZR2021MD097), the Natural Science Foundation of Ningxia (Grant No. 2024AAC03419), and Shandong Provincial Meteorological Bureau Innovation Team Special Project (Grant No. 2024sdcxtd04).

[References]

[1] Palacios F, Diago M P, Melo-Pinto P, Tardaguila J. Early yield prediction in different grapevine varieties using computer vision and machine learning. Precision Agriculture, 2023; 24(2): 407–435.

[2] Lanza B, Botturi D, Gnutti A, Lancini M, Nuzzi C, Pasinetti S. A stride toward wine yield estimation from images: Metrological validation of grape berry number, radius, and volume estimation. Sensors, 2024; 24(22): 7305.

[3] Liu S, Zeng X, Whitty M. A vision-based robust grape berry counting algorithm for fast calibration-free bunch weight estimation in the field. Computers and Electronics in Agriculture, 2020; 173: 105360.

[4] Coviello L, Cristoforetti M, Jurman G, Furlanello C. GBCNet: In-field grape berries counting for yield estimation by dilated CNNs. Applied Sciences, 2020; 10(14): 4870.

[5] Buayai P, Yok-In K, Inoue D, Nishizaki H, Makino K, Mao X Y. Supporting table grape berry thinning with deep neural network and augmented reality technologies. Computers and Electronics in Agriculture, 2023; 213: 108194.

[6] Blasco J, Aleixos N, Cubero S, Gomez-Sanchis J, Molto E. Automatic sorting of satsuma (*Citrus unshiu*) segments using computer vision and morphological features. Computers and Electronics in Agriculture, 2009; 66(1): 1–8.

[7] Bhargava A, Bansal A. Automatic detection and grading of multiple fruits by machine learning. Food Analytical Methods, 2020; 13(3): 751–761.

[8] Bhargava A, Bansal A, Goyal V. Machine learning–based detection and sorting of multiple vegetables and fruits. Food Analytical Methods, 2022; 15(1): 228–242.

[9] Gongal A, Amatya S, Karkee M, Zhang Q, Lewis K. Sensors and systems for fruit detection and localization: A review. Computers and Electronics in Agriculture, 2015; 116: 8–19.

[10] Ji W, Zhao D A, Cheng F Y, Xu B, Zhang Y, Wang J J. Automatic recognition vision system guided for apple harvesting robot. Computers and Electrical Engineering, 2012; 38(5): 1186–1195.

[11] Zabawa L, Kicherer A, Klingbeil L, et al. Detection of single grapevine berries in images using fully convolutional neural networks. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Long Beach: IEEE, 2019; pp.2571–2579. doi: 10.1109/CVPRW.2019.00313.

[12] Feng Q H, Xu X Z, Wang Z X. Deep learning-based small object detection: A survey. Mathematical Biosciences and Engineering, 2023; 20(4): 6551–6590.

[13] Koirala A, Walsh K B, Wang Z, McCarthy C. Deep learning for real-time fruit detection and orchard fruit load estimation: Benchmarking of 'MangoYOLO'. Precision Agriculture, 2019; 20(6): 1107–1135.

[14] Shen Q, Zhang X, Shen M, et al. Multi-scale adaptive YOLO for instance segmentation of grape pedicels. Computers and Electronics in Agriculture, 2025; 229: 109712.

[15] Chen Y M, Li X, Jia M, Li J L, Hu T Y, Luo J. Instance segmentation and number counting of grape berry images based on deep learning. Applied Sciences, 2023; 13(11): 6751.

[16] Zhang X, Gao Q M, Pan D, Cao P C, Huang D H. Research on spatial positioning system of fruits to be picked in field based on binocular vision and SSD model. In: 2020 International Conference on Computer, Information and Telecommunication Systems (CITS), Jiaxing, China: IOP Publishing, 2021; 1748: 042011.

[17] Liang Q K, Zhu W, Long J Y, Wang Y N, Sun W, Wu W N. A real-time detection framework for on-tree mango based on SSD network. In: 11th International Conference on Intelligent Robotics and Applications (ICIRA 2018), Singapore: Springer, 2018; pp.423–436. doi: 10.1007/978-3-319-97589-4_36.

[18] Sa I, Ge Z Y, Dayoub F, Upcroft B, Perez T, McCool C. Deepfruits: A fruit detection system using deep neural networks. Sensors, 2016; 16(8): 1222.

[19] Wan S, Goudos S. Faster R-CNN for multi-class fruit detection using a robotic vision system. Computer Networks, 2020; 168: 107036.

[20] Tu S Q, Pang J, Liu H F, Zhuang N, Chen Y, Zheng C, et al. Passion fruit detection and counting based on multiple scale faster R-CNN using RGB-D images. Precision Agriculture, 2020; 21: 1072–1091.

[21] Mai X C, Zhang H, Meng M Q H. Faster R-CNN with classifier fusion for small fruit detection. In: 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia: IEEE, 2018; pp.7166–7172. doi: 10.1109/ICRA.2018.8461130.

[22] Santos T T, de Souza L L, dos Santos A A, Avila S. Grape detection, segmentation, and tracking using deep neural networks and three-dimensional association. Computers and Electronics in Agriculture, 2020; 170: 105247.

[23] He K M, Gkioxari G, Dollár P, Girshick R. Mask R-CNN. In: 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy: IEEE, 2017; pp.2980–2988. doi: 10.1109/ICCV.2017.322.

[24] Tian Y N, Yang G D, Wang Z, Li E, Liang Z Z. Instance segmentation of apple flowers using the improved mask R-CNN model. Biosystems Engineering, 2020; 193: 264–278.

[25] Shen L, Su J Y, Huang R, Quan W M, Song Y Y, Fang Y L, et al. Fusing attention mechanism with Mask R-CNN for instance segmentation of grape cluster in the field. Frontiers in Plant Science, 2022; 13: 934450.

[26] Jia W K, Tian Y Y, Luo R, Zhang Z H, Lian J, Zheng Y J, et al. Detection and segmentation of overlapped fruits based on optimized mask R-CNN

application in apple harvesting robot. Computers and Electronics in Agriculture, 2020; 172: 105380.
[27] Huang Y P, Wang T H, Basanta H. Using fuzzy mask R-CNN model to automatically identify tomato ripeness. IEEE Access, 2020; 8: 207672–207682.
[28] Liu Y, Sun P, Wergeles N, Shang Y. A survey and performance evaluation of deep learning methods for small object detection. Expert Systems with Applications, 2021; 172: 114602.
[29] Huang X, Peng D D, Qi H N, Zhou L, Zhang C. Detection and instance segmentation of grape clusters in orchard environments using an improved mask R-CNN model. Agriculture, 2024; 14(6): 918.
[30] Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv, arXiv: 2010.11929. doi: 10.48550/arXiv.2010.11929.
[31] Liu Z, Lin Y T, Cao Y, Hu H, Wei Y X, Zhang Z. Swin transformer: Hierarchical vision transformer using shifted windows. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada: IEEE, 2021; pp.9992–10002. doi: 10.1109/ICCV48922.2021.00986.
[32] Liu H, Wang X X, Zhao F Y, Yu F Y, Lin P, Gan Y, et al. Upgrading swin-B transformer-based model for accurately identifying ripe strawberries by coupling task-aligned one-stage object detection mechanism. Computers and Electronics in Agriculture, 2024; 218: 108674.
[33] Xiao B J, Nguyen M, Yan W Q. Fruit ripeness identification using transformers. Applied Intelligence, 2023; 53(19): 22488–22499.
[34] Song K, Yang J, Wang G. A Swin transformer and MLP based method for identifying cherry ripeness and decay. Frontiers in Physics, 2023; 11: 1278898.
[35] Zheng H, Wang G H, Li X C. Swin-MLP: A strawberry appearance quality identification method by Swin Transformer and multi-layer perceptron. Journal of Food Measurement and Characterization, 2022; 16(4): 2789–2800.
[36] Wang J H, Zhang Z Y, Luo L F, Wei H L, Wang W, Chen M Y, et al. DualSeg: Fusing transformer and CNN structure for image segmentation in complex vineyard environment. Computers and Electronics in Agriculture, 2023; 206: 107682.
[37] Lu S L, Liu X Y, He Z X, Zhang X, Li W B, Karkee M. Swin-Transformer-YOLOv5 for real-time wine grape bunch detection. Remote Sensing, 2022; 14(22): 5853.
[38] Liu G X, Zhang Y H, Liu J, Liu D Y, Chen C L, Li Y J, et al. An improved YOLOv7 model based on Swin Transformer and Trident Pyramid Networks for accurate tomato detection. Frontiers in Plant Science, 2024; 15: 1452821.
[39] Charisis C, Argyropoulos D. Deep learning-based instance segmentation architectures in agriculture: A review of the scopes and challenges. Smart Agricultural Technology, 2024; 8: 100448.
[40] Zhang Z J, Fu H Z, Dai H, Shen J B, Pang Y W, Shao L. ET-Net: A generic edge-attention guidance network for medical image segmentation. In: 22nd International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI 2019), Shenzhen, China: Springer, 2019; pp.442–450. doi: 10.1007/978-3-030-32239-7_49.
[41] Zhao J X, Liu J J, Fan D P, Cao Y, Yang J F, Cheng M M. EGNet: Edge guidance network for salient object detection. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea (South): IEEE, 2019; pp.8778–8787. doi: 10.1109/ICCV.2019.00887.
[42] Yang J Y, Jiao L C, Shang R H, Liu X, Li R Y, Xu L C. EPT-Net: Edge perception transformer for 3D medical image segmentation. IEEE Transactions on Medical Imaging, 2023; 42(11): 3229–3243.
[43] He C, Li S L, Xiong D H, Fang P Z, Liao M S. Remote sensing image semantic segmentation based on edge information guidance. Remote Sensing, 2020; 12(9): 1501.
[44] Sheng X, Kang C M, Zheng J Y, Lyu C. An edge-guided method to fruit segmentation in complex environments. Computers and Electronics in Agriculture, 2023; 208: 107788.
[45] Jia W, Cao K, Liu M Y, Lu Y Q, Ji Z, Liu G L, et al. FCOS-EAM: An accurate segmentation method for overlapping green fruits. Computers and Electronics in Agriculture, 2024; 226: 109392.
[46] Zimmermann R S, Siems J N. Faster training of Mask R-CNN by focusing on instance boundaries. Computer Vision and Image Understanding, 2019; 188: 102795.
[47] Yang C C, Geng T Y, Peng J, Xu C, Song Z C. Mask-GK: An efficient method based on mask Gaussian kernel for segmentation and counting of grape berries in field. Computers and Electronics in Agriculture, 2025; 234: 110286.
[48] Liu S T, Huang D, Wang Y H. Learning spatial fusion for single-shot object detection. arXiv, 2019; arXiv: 1911.09516. doi: 10.48550/arXiv.1911.09516.
[49] Du W S, Liu P. Instance segmentation and berry counting of table grape before thinning based on AS-SwinT. Plant Phenomics, 2023; 5: 0085.
[50] Li H, Xiong X R, Liu C X, Ma Y, Zeng S, Li Y Q. SFFNet: Staged feature fusion network of connecting convolutional neural networks and graph convolutional neural networks for hyperspectral image classification. Applied Sciences, 2024; 14(6): 2327.
[51] Li S, Bitsch L, Hanf J. Grape supply chain: Vertical coordination in Ningxia, China. Asian Journal of Agriculture, 2021; 5(1): 050103.
[52] Chen R W, Zhang X Y, Yang Y, Yang Y E, Wang J, Li H Y. Analyses of vineyard microclimate in the eastern foothills of the Helan Mountains in Ningxia region, China. Sustainability, 2023; 15(17): 12740.