Journal of Computing and Electronic Information Management
ISSN: 2413-1660 | Vol. 21, No. 1, 2026

SDAFormer: A Semantic-Guided and Detail-Aware Transformer for Apple Counting in Complex Orchards

Chenyu Zhu *

Henan Polytechnic University, Jiaozuo, 454000, China

* **Corresponding author**

Abstract: Accurate apple counting is crucial for orchard yield estimation and automated management. However, in complex natural agricultural settings, issues such as scale variations, fruit occlusion, and background interference pose significant challenges to existing counting methods. Current mainstream models often struggle to balance global contextual information with local fine-grained features, resulting in inaccurate counts in these areas and difficulty in effectively distinguishing fruits from complex backgrounds. To address the issues of easily disturbed shallow-level details and insufficient coordination between high-level semantics and local structure that apple targets face under varying scales and occlusion conditions in real orchard scenarios, this paper proposes a semantic-guided and detail-aware Transformer-based apple counting method, Named SDAFormer. This method uses the Semantic-Aware Detail Refinement Module (SADRM) to explicitly inject deep semantic information into shallow-level edge, texture, and local structural features, thereby enhancing the feature completeness and discriminative power of occluded apple regions; Through the Coordinate-Aware Multi-scale Module (CAMM), it enhances the position-aware capabilities and multi-scale context modeling during the density map regression stage, thereby improving the model's counting stability under varying scales and in partially occluded scenarios. Experimental results demonstrate that this method achieves superior counting performance on a self-built apple dataset, with a Mean Absolute Error (MAE) of 3.61 and a Mean Squared Error (MSE) 4.76.

Keywords: Apple counting; Density map estimation; Semantic guidance; Transformer.

1. Introduction

Fruit counting is a critical task in modern fruit industry management; particularly in apple orchards, information on fruit quantity directly or indirectly influences yield forecasts, management decisions, and economic returns [1-3]. Obtaining accurate data on apple quantity and density distribution is the foundation for optimizing fertilization, irrigation, and harvest planning [4-5]. In the absence of intelligent counting systems, traditional methods often rely on manual estimation, which is not only time-consuming, labor-intensive, and costly [6-7], but also prone to significant errors due to fruit displacement and limitations in observation angles, making it difficult to meet the demands of precision agriculture [8]. Therefore, developing non-destructive automatic counting models to accurately estimate apple counts has become a priority for advancing smart fruit farming and enhancing ecological benefits [9]. Compared to manual sampling and extrapolation, intelligent non-contact counting provides more timely and comprehensive yield information without interfering with tree growth, and is regarded as a foundational capability for digital twins and precision management in modern orchards [10]. In recent years, with the widespread adoption of drones, ground-based mobile platforms, and multimodal visual sensors, research on fruit detection and counting has advanced rapidly [11]; however, under complex canopies, variations in scale, occlusion by branches and leaves, clustering and overlapping, as well as intense non-uniform illumination, remain the primary sources of counting errors [12-13].

Apple counting research has evolved from early manual feature-based methods to a deep learning phase centered on detection, regression, and density mapping. Among these, detection methods excel in instance-level interpretability and engineering deployment; regression methods are more attractive for weak supervision and low annotation costs; and density mapping methods demonstrate greater advantages in high-density and heavily occluded scenarios. For apple counting research and applications in complex scenarios, the truly valuable research question is not simply choosing one of these methods, but rather how to construct more robust feature representations, more reasonable supervision methods, and counting workflows closer to real-world applications—all while addressing the multiple challenges of dense occlusion, scale variations, and background interference in actual orchard environments.

In real orchards, apple counting still faces three core challenges: First, scale variation: sparse single apples coexist with dense clusters, and targets exhibit a cross-scale distribution ranging from small-sized objects at a distance to large-sized objects at close range, placing higher demands on the model's multi-scale perception and feature fusion; Second, background interference: Apples share certain similarities with the foliage background, particularly during the pre-ripening stage when green apples closely resemble the background in both color and texture, making it difficult to distinguish the boundaries between apples and foliage and thereby weakening feature stability; Third, occlusion: Mutual occlusion between branches, leaves, and fruits leads to the loss of key texture and contour information, increasing counting uncertainty.

To address these challenges, we propose an apple counting method based on a semantic-guided and detail-aware Transformer. This method uses a Pyramid Vision Transformer as its backbone. By introducing a semantic-aware detail refinement module, it explicitly injects deep semantic information into shallow-layer edge, texture, and local structural features, thereby enhancing the features of occluded

apple regions. Additionally, a coordinate-aware multi-scale module is designed to introduce position-aware and multi-scale context modeling during the density map regression stage, improving the model's ability to distinguish the background and adapt to scale changes. This method achieves more reasonable density responses in scenarios where occlusion and scale variations are coupled, thereby improving the accuracy and stability of apple counting in complex scenes. The specific contributions of this work are as follows:

1. We propose a Semantic-Aware Detail Refinement Module (SADRM) that combines wavelet transformation enhancement with multi-scale consistency attention. By explicitly injecting deep semantic information into shallow-level edge, texture, and local structural features, this module enhances the features of occluded apple regions, thereby improving counting stability in scenes where objects are occluded.

2. We propose a Coordinate-Aware Multi-scale Module (CAMM) to address the challenges posed by scale variations during apple counting. This module introduces position-aware and multi-scale context modeling during the density map regression stage to improve the model's adaptability to spatial distribution variations and scale changes.

3. We conducted thorough qualitative and quantitative evaluations on our self-built real-world apple dataset. SDAFormer outperformed various mainstream counting methods on multiple key metrics, and these results strongly validate the effectiveness and robustness of the proposed model in complex agricultural scenarios.

2. Related Work

2.1. Counting Based on Traditional Methods

Before the widespread adoption of deep learning methods, research on fruit counting primarily relied on manually designed features and standardized image processing workflows. The core approach involved separating fruit regions from complex backgrounds using techniques such as color space transformation, threshold segmentation, texture analysis, shape priors, and geometric constraints, and subsequently performing fruit detection, quantity estimation, or yield prediction. These methods typically follow a basic technical workflow of preprocessing, feature extraction, object segmentation, and count estimation, and dominated early research on automation in orchard settings.

Researchers have conducted further explorations. Wang [14] et al. addressed the issue of overlapping fruits in apple-picking robot scenarios by utilizing the Lab color space, K-means clustering, morphological processing, and local extremum localization to achieve fruit separation and localization. Subsequently, Zhang [15] et al. proposed an apple image segmentation method that fuses color and texture features with machine learning to enhance the ability to extract apple regions in complex backgrounds; in follow-up research, Fan P [16] et al. proposed an apple segmentation method based on the grayscale-centered RGB color space to improve the separability between fruits and the background. The aforementioned work demonstrates that traditional methods have evolved beyond simple threshold segmentation and have gradually developed into image segmentation frameworks that integrate multiple features.

Overall, fruit counting research based on traditional methods has laid a crucial foundation for subsequent automated orchard perception. These methods offer advantages such as clear implementation pathways, strong interpretability, and low dependence on small-scale data, and they retain practical value in scenarios with regular tree structures, relatively simple backgrounds, or distinct color differences. However, their shortcomings are equally evident: on the one hand, model performance is highly dependent on manually designed color, texture, and shape features, making them sensitive to environmental changes and empirical parameters; on the other hand, under real-world orchard conditions—such as leaf and branch occlusion, fruit overlap, complex lighting, and similar backgrounds—the robustness and cross-scenario generalization capabilities of traditional methods are often limited.

2.2. Image counting methods based on detection

Detection-based image counting methods typically transform the counting problem into a question of how many objects are detected, and count those objects accordingly. The technical foundation of this class of methods stems from the development of convolutional neural networks and modern object detection frameworks, including feature extraction networks such as CNN [17], VGG [18], and ResNet[19]. In terms of detection frameworks, Faster R-CNN [20] and YOLO [21] have driven the maturation of the instance detection paradigm.

In apple-related scenarios, research on detection-based counting has accelerated significantly over the past five years. Gao [22] et al. employed YOLOv4-tiny for detection, Kalman filtering for prediction, and Hungarian matching to achieve apple detection and counting in orchard videos. Zhao [23] et al. focused on apple recognition in complex orchard environments. Hu [24] et al. combined an improved YOLOv7 with multi-object tracking methods for detection and counting in apple orchards. Abeyrathna et al. [25] combined 3D cameras, the YOLO series, and Deep SORT for apple recognition and counting under dynamic conditions. In the past two years, this approach has further evolved toward integrating detection with tracking and multi-view methods. For example, Yang et al. [26] proposed AD-YOLO and MR-SORT for automatic apple detection and counting. Matos [27] et al. proposed an apple tracking and counting method that combines spatio-temporal and geometric constraints to address intermittent occlusion and low frame rates. Jin et al. [28] integrated detection, localization, and counting into complex robotic harvesting scenarios in orchards. Cao [29] et al. further utilized an improved YOLOv7-Tiny-PDE and tracking mechanisms to handle apple detection and counting in complex scenarios.

Overall, detection-based image counting methods offer the advantage of relatively strong instance-level interpretability, providing fruit location, class, and quantity. However, under conditions of severe occlusion, dense clustering, and extremely small fruit sizes, missed detections and incorrect segmentation due to overlap remain prevalent.

2.3. Counting Based on Regression Methods

Regression-based image counting methods do not require the explicit detection of each individual fruit instance; instead, they directly learn the mapping relationship between images or image regions and the corresponding count. The rationale behind this approach is that when there are a large number of fruits, severe occlusion, or when instance boundaries are difficult to label clearly, directly regressing the total count is

often easier to train than detecting instances one by one, and it also reduces annotation costs. Wang et al. [30] proposed DeepPhenology, which uses deep learning to estimate the distribution of apple blossoms during the flowering season. Although this task is not entirely equivalent to counting ripe fruits, it is a significant representative of methods that use overall image features to regress the distribution of crop states. Bhattarai and Karkee [31] proposed the weakly supervised regression-based CountNet, which can count apple blossoms and apples using only image-level count labels, without requiring precise detection or segmentation;

Overall, the primary advantages of regression-based image counting methods lie in their straightforward training objectives and low requirements for annotation formats, making them particularly suitable for scenarios with unclear boundaries, occlusions, or a large number of objects. However, the shortcomings of these methods are equally evident: their interpretability is typically weaker than that of detection methods, and they struggle to naturally provide clear boundaries and instance identities for each fruit; when scene distributions vary significantly, the model may learn background cues that are correlated with the actual number of fruits but are unstable.

2.4. Counting Based on Density Maps

Image counting methods based on density maps typically use point annotations as supervision signals, representing the pixel contributions near each object as continuous density distributions, and then calculating the total count by integrating the density across the entire image. This approach was initially systematically developed for crowd counting. Lempitsky and Zisserman [32] proposed the fundamental idea of object counting via density integration. Zhang [33] et al. introduced MCNN, which employs a multi-column convolutional architecture to adapt to different object scales. Li [34] et al. proposed CSRNet, utilizing dilated convolutions to preserve feature resolution while expanding the receptive field. Ma [35] et al. introduced Bayesian Loss to improve counting training under point supervision. Wang [36] et al. introduced DM-Count, noting that point annotations should not simply be subjected to fixed Gaussian smoothing, but rather should learn more reasonable density representations through distribution matching.

Building on this foundation, density map methods have gradually evolved into a technical framework comprising point annotations, density estimation, and integral counting. Image counting methods based on density map estimation are particularly well-suited for scenarios involving dense, small, or heavily occluded objects. A major advantage is that training typically requires only point annotations, significantly reducing annotation costs. At the same time, by learning density distributions, the model can avoid the difficulties associated with manually bounding each instance; however, its shortcomings must also be acknowledged: spatial peaks in density maps do not correspond to strict instance boundaries, so localization accuracy and instance-level interpretability are generally inferior to detection methods. Furthermore, the quality of density maps can be affected by noise in point annotations, scale changes, and locally dense distributions.

3. Method

3.1. Overall Architecture

To address counting errors caused by significant occlusions and pronounced scale variations in complex orchard scenarios. This paper proposes an apple counting method based on a semantic-guided and detail-aware Transformer, referred to as SDAFormer. The method uses PVT as the backbone network and generally follows a design approach involving multi-level feature extraction, semantic-guided detail refinement, and coordinate-aware multi-scale regression. For the input raw image Image, the backbone network first extracts hierarchical features across four stages, denoted as F1, F2, F3, and F4. The spatial resolutions of F1 to F4 are 1/4, 1/8, 1/16, and 1/32 of the original image, respectively. Shallow-level features retain more edge, texture, and local structural information, making them more sensitive to small-scale apples and the boundaries of dense clusters; deep-level features possess stronger semantic expression capabilities and larger receptive fields, which are advantageous for providing stable target discrimination criteria in complex scenes. Based on the current network implementation, the second-stage feature F2 has a higher resolution and is suitable as the detail representation branch; the fourth-stage feature F4 has the most stable semantics and is suitable as the high-level semantic guidance branch; the third-stage feature F3 combines spatial structure with mid-level semantic information and is suitable as an intermediate input for subsequent density map regression. The fourth-stage feature F4 provides more stable high-level semantic constraints for subsequent feature updates.

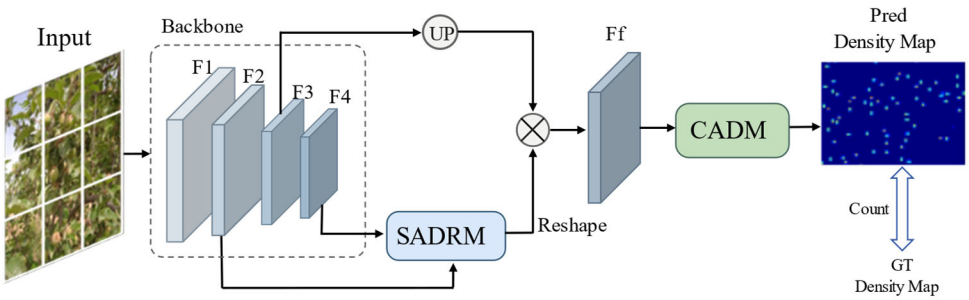

Figure 1. SDAFormer Overall Framework

Given that apple counting in complex orchard scenarios faces challenges related to both occlusion and scale variation, this paper does not directly use single-layer features for density map regression. Instead, it introduces two modules—the Semantic-Aware Detail Refinement Module (SADRM) and the Coordinate-Aware Multi-Scale Module (CAMM)—to further model the backbone features. By feeding F2 and F4 into the Semantically Aware Detail Refinement Module, we obtain the semantically guided detail refinement feature F_{sadrm}, where F_{sadrm} maintains the same spatial resolution as F2 but offers greater discriminative power in semantic representation. The introduction of the Semantically Aware

Detail Refinement Module enables the model to explicitly incorporate high-level semantic constraints into shallow-level detail features, thereby enhancing the model's ability to represent apple features. The third-stage feature F3 is upsampled and fused with F_{sadrm} , and the result is fed into the Coordinate-Aware Multi-Scale Module. Based on position-aware and multi-scale modeling, the final density map is generated through regression. The overall structural diagram is shown in Figure 1.

3.2. Semantic-Aware Detail Refinement Module

The design philosophy of SADRM is inspired by the collaborative modeling approach of encoders and decoders in DETR. In DETR, the encoder models global relationships among input features, while the decoder learns target-level features through the interaction between the query sequence and the memory. This paper draws on the decoder-based interaction concept from DETR, treating shallow-level fine-grained features as query sequences that need to be updated, and deep-level high-level semantic features as a memory that provides semantic constraints, thereby enabling directed interaction between features at different levels. Unlike DETR, this paper does not introduce an additional independent encoder but retains only the decoder-based update path. This is because the PVT backbone network adopted in this paper has already achieved sufficient contextual aggregation during the hierarchical feature extraction process, particularly as the deepest-layer features possess strong global semantic expression capabilities. Adding an independent encoder would likely result in redundant modeling of high-level semantics and increase computational overhead. Directly using the deepest-layer features as semantic memory and interacting with shallow-layer detailed features through decoding better aligns with the task characteristics of apple counting in complex scenes, where local visible information is incomplete but overall semantic information remains exploitable. The structure of SADRM is shown in Figure 2.

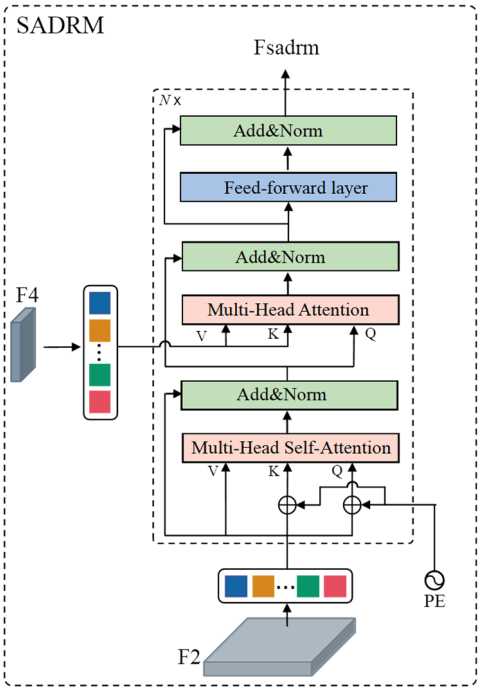

Figure 2. SADRM Structure Diagram

In terms of the feature hierarchy, after the input image passes through the backbone network, the features output in the second stage are denoted as $F_2 \in \mathbb{R}^{\frac{H}{8}\times\frac{W}{8}\times C_2}$, The output features of the fourth stage are denoted as $F_4 \in \mathbb{R}^{\frac{H}{32}\times\frac{W}{32}\times C_4}$. In particular, F2 has a higher spatial resolution and primarily captures detailed information such as the edges, contours, and local textures of the apple. F4, on the other hand, undergoes deeper feature extraction and possesses stronger semantic generalization capabilities. Due to differences in the number of channels and semantic levels between the two, F2 is first convoluted and projected to align its channel dimension with that of F4, and then flattened into a sequence of detailed features:

$$X = Flatten(v_1(F_2)) \quad (1)$$

Where $v_1(\cdot)$ Indicates channel mapping operations.After mapping, the dimensions of $v_1(F_2)$ can be expressed as $v_1(F_2) \in \mathbb{R}^{\frac{H}{8}\times\frac{W}{8}\times C_4}$ Therefore, the sequence X can be denoted as $X \in \mathbb{R}^{\frac{HW}{64}\times C_4}$. The purpose of this step is to map the second-stage features into a representation space consistent with deep-level semantics, while preserving as much as possible the local contours, edges, and visible textures of the apple, thereby providing a fine-grained foundation for subsequent feature updates under occlusion conditions. For the fourth-stage features F4 , they are flattened into a semantic sequence and combined with positional encoding to construct the semantically guided sequence M:

$$M = Flatten(F_4) + P_{pos} \quad (2)$$

Since the spatial resolution of F4 is 1/32 that of the original image, the semantic memory size corresponding to the flattened sequence length can be expressed as $M \in \mathbb{R}^{\frac{HW}{1024}\times C_4}$。

Here, P_{pos} represents the position encoding. With the addition of positional encoding, the high-level semantic sequence not only contains discriminative information between the apple region and the background region but also preserves the relative distribution relationships between different spatial positions. This configuration helps the model establish a correspondence between locally visible fragments and the overall target semantics during interaction, thereby improving the recognizability of occluded apples. After obtaining the detail sequence X and the semantic guidance sequence M , SADRM establishes global dependencies within the detail sequence through a self-attention mechanism.

$$X_1 = \text{LayerNorm}\big(X + \text{SelfAttn}(X)\big) \quad (3)$$

This process allows shallow detail features to transcend the limitations of local convolutional neighborhoods, enabling the integration of contextual information across a broader scope. For the apple counting task in occlusion scenarios, this step connects visible fragments of the same apple that are scattered across different local regions, providing a more consistent structural representation of fragmented edges and local textures, and laying a more reliable foundation of detail for subsequent semantic constraints. After the detail sequence completes its internal context modeling, SADRM updates it via cross-attention using the high-level semantic sequence M:

$$X_2 = \text{LayerNorm}\big(X_1 + \text{CrossAttn}(X_1, M)\big) \quad (4)$$

The key to this process lies in directly incorporating high-level semantic information into the update of low-level features. For apple counting, this update mechanism enables the model to explicitly enhance responses related to the apple target while maintaining fine-grained resolution, ensuring that areas obscured by leaves, blocked by adjacent fruits, or

only partially exposed can still obtain consistent target representations within the constraints of a broader semantic context. In other words, SADRM does not focus solely on suppressing the background, but rather on using high-level semantic information to filter, supplement, and reorganize incomplete local visual information, thereby mitigating the problem of feature loss caused by occlusion. After cross-attention updates, the module performs nonlinear transformations and channel reorganization via a feedforward network, restoring the output sequence to a spatial feature map:

$$F_{sadrm} = \text{Reshape}\left(\text{LayerNorm}\left(X_2 + \text{FFN}(X_2)\right)\right) \quad (5)$$

The dimensions of the reconstructed feature map can be expressed as $F_{sadrm} \in \mathbb{R}^{\frac{H}{8}\times\frac{W}{8}\times C_4}$. As can be seen, the output of SADRM maintains the same spatial resolution as the second-stage features but has been unified with deep semantic features in terms of channel dimensions. Therefore, it is no longer a simple shallow-level detail feature but rather a refined detail feature that integrates high-level semantic constraints.

In terms of its functional role, SADRM primarily addresses the issues of incomplete local structures and a lack of semantic support for shallow-level details in occluded apples. Its contribution lies in enhancing the quality of intermediate features for occluded apples prior to density map regression, enabling the model to generate a relatively stable discriminative representation for partially visible objects. For this reason, SADRM is better suited to serve as a key module in the feature representation stage rather than in the multiscale regression stage. This provides reliable input for CAMM to further enhance the model's adaptability during the location-aware and multi-scale density regression stages.

3.3. Coordinate-Aware Multi-scale Module

The Coordinate-Aware Multi-scale Module, CAMM, takes as input the stage-three feature F3 and the SADRM-refined feature F_{sadrm}, as shown in Figure 3-3.

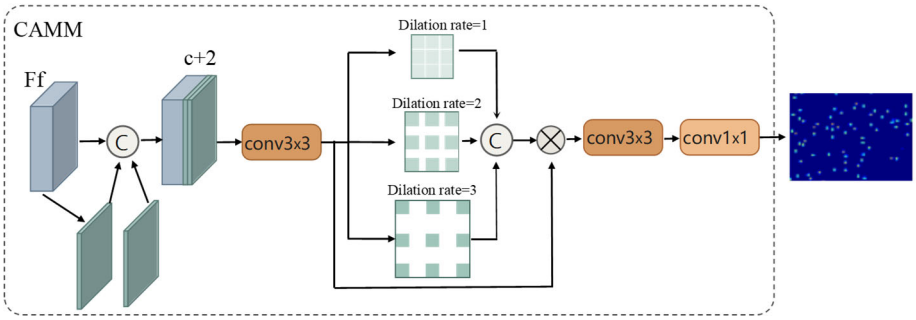

Figure 3. CAMM Structure Diagram

The stage-three feature can be expressed as $F_3 \in \mathbb{R}^{\frac{H}{16}\times\frac{W}{16}\times C_3}$, while the output of SADRM is $F_{sadrm} \in \mathbb{R}^{\frac{H}{8}\times\frac{W}{8}\times C_4}$, As can be seen, the two images differ in both spatial resolution and the number of channels; therefore, alignment must be performed before feature fusion. Upsample F3 to change its spatial resolution from $\frac{H}{16}\times\frac{W}{16}$ to $\frac{H}{8}\times\frac{W}{8}$, and then use convolution mapping to unify the number of channels. At the same time, apply convolution projection to F_{sadrm} to ensure its channel dimension matches that of the former.

$$F_3' = \text{Conv}\left(\text{Up}(F_3)\right), F_m = \text{Conv}(F_{sadrm}) \quad (6)$$

Where, $F_3', F_m \in \mathbb{R}^{\frac{H}{8}\times\frac{W}{8}\times C_m}$。

Therefore, before performing element-wise addition, the two images are already fully aligned in terms of spatial dimensions and the number of channels, satisfying the conditions for fusion. The fusion feature is expressed as:

$$F_f = F_3' + F_m \quad (7)$$

Thus, $F_f \in \mathbb{R}^{\frac{H}{8}\times\frac{W}{8}\times C_m}$. This step achieves the complementary fusion of mid-level structural information and semantically refined detailed information, enabling the regression module to both preserve the apple's boundaries and local structural differences, and to enhance the response of the target region by leveraging the semantic filtering results obtained in the previous stage.

Based on the fused feature F_f, CAMM further explicitly incorporates coordinate information. For the fused feature of size $\frac{H}{8}\times\frac{W}{8}$, horizontal and vertical coordinate maps G_x and $G_y$can be generated, respectively. After concatenating these with F_f along the channel dimension, we obtain the position−enhanced feature:

$$F_c = \text{Concat}(F_f, G_x, G_y) \quad (8)$$

Since G_x and G_y each have one additional channel, if the number of channels in F_f is C_m, then $F_c \in \mathbb{R}^{\frac{H}{8}\times\frac{W}{8}\times(C_m+2)}$, Through this operation, the model is able to explicitly perceive the spatial locations of different feature responses during subsequent convolutional regression. For the apple counting task, this step enables the network to better establish the relationship between spatial locations and apple distribution, thereby enhancing its adaptability to scenarios with non-uniform distributions.

After obtaining the location-enhanced features, CAMM derives the stem features through a basic convolutional branch:$F_s = \text{Conv}(F_c)$. Let the stem output channel be C_s; then, $F_s \in \mathbb{R}^{\frac{H}{8}\times\frac{W}{8}\times C_s}$.Subsequently, the module uses three convolutional branches with different void ratios to extract multi-scale contextual information:

$$F_{d1} = \text{Conv}_{r=1}(F_s), F_{d2} = \text{Conv}_{r=2}(F_s), F_{d3} = \text{Conv}_{r=3}(F_s) \quad (9)$$

Since all three branches maintain the same spatial resolution through appropriate padding, and assuming that each branch has C_d output channels, we have $F_{d1}, F_{d2}, F_{d3} \in \mathbb{R}^{\frac{H}{8}\times\frac{W}{8}\times C_d}$. By concatenating the outputs of the three branches along the channel dimension, we obtain a multi-scale feature representation:

$$F_{ms} = \text{Concat}(F_{d1}, F_{d2}, F_{d3}) \quad (10)$$

Therefore, $F_{ms} \in \mathbb{R}^{\frac{H}{8}\times\frac{W}{8}\times 3C_d}$. To enable element-wise fusion with the structural features from the stem, CAMM is designed such that $C_s = 3C_d$, ensuring that F_{ms} and F_s are

fully consistent in terms of the number of channels and spatial dimensions. After dimensional alignment is complete, the module performs element−wise multiplicative fusion of the multiscale features with the stem features: After alignment, the module performs element-wise multiplication between the multi-scale feature and the stem feature:

$$F_g = F_{ms} \odot F_s \tag{11}$$

The purpose of this step is to modulate the multiscale response using structural features, thereby further enhancing the effective scale response associated with the apple target. Finally, the enhanced features are passed through a convolutional regression head to produce a single-channel density map $D \in \mathbb{R}^{\frac{H}{8}\times\frac{W}{8}\times 1}$, as shown in the formula:

$$D = \mathrm{Conv}_{1\times1}\left(\mathrm{Conv}_{3\times3}(F_g)\right) \tag{12}$$

Consequently, the model generates the final density map based on features enhanced by location-based information and multiscale modeling, and calculates the apple count by integrating the density map.

In terms of module responsibilities, the primary role of CAMM is not to further supplement semantic information in occluded regions, but to address response shifts caused by significant variations in apple size and uneven spatial distribution during the regression stage. Explicit coordinate encoding enhances the model's ability to perceive spatial position differences, while the multi-void-rate volume integration branch enhances the model's adaptability to apples of different scales. Together, these two components improve the rationality of the density map structure and the stability of the regression. Therefore, CAMM is better suited to serve as the key module in the density regression stage. SADRM is responsible for making the features of occluded apples more prominent, while CAMM is responsible for ensuring more stable regression of apples of different scales. Overall, CAMM is not merely a simple convolutional regression head, but a location-aware and multi-scale density estimation module specifically designed for complex orchard scenarios.

4. Experiment

4.1. Dataset Description

To validate the effectiveness of the proposed method in complex real-world scenarios, this paper constructed an apple counting dataset. The data was collected from apples in the early stages of thinning, when the fruits are generally unripe and appear greenish-yellow. Their color is similar to that of adjacent fruits and leaves, making them difficult to distinguish visually. At the same time, apples typically grow in clusters, resulting in frequent occlusion between targets, and interference from the foliage background is also significant. These factors all have a certain impact on apple feature extraction and accurate counting. Therefore, the dataset constructed in this paper exhibits strong characteristics of complex scenes and can realistically reflect the actual challenges of apple counting tasks in natural orchard environments.

Data collection took place at the Huaniu Orchard in Yuanlong Town, Maji District, Tianshui City, Gansu Province, during the pre-thinning stage. At this stage, most apples are greenish-yellow, and the fruits are often clustered, exhibiting high similarity in color and texture to the surrounding leaves. Consequently, distinguishing fruit targets from the background is challenging, which better aligns with the practical requirements of apple counting in complex scenes. Compared to images of ripe red apples, images from this stage are more susceptible to occlusion, confusion caused by similar backgrounds, and the absence of fine-grained features, thereby providing a more comprehensive test of the model's robustness.

During data collection, a smartphone was used to capture images, resulting in a total of 1,668 raw images, all with a resolution of 1920×1080. To ensure the representativeness and validity of the data, images of poor quality, those with redundant information, or those with high scene repetition were excluded during subsequent screening. Ultimately, 862 representative sample images were retained for this experiment. For the experiment, the dataset was divided into a training set and a test set. The training set consisted of 689 images and 689 corresponding annotation files, while the test set comprised 173 images and 173 corresponding annotation files. This division ensures the data volume required for model training while also allowing for a relatively objective evaluation of the method's generalization performance on unseen samples.

4.2. Evaluation Criteria and Experimental Setup

To comprehensively evaluate model performance, this paper adopts error analysis of counting results as the criterion for judging the accuracy of the proposed method. To clearly reflect the superiority and inferiority of different methods in counting performance and prediction quality, mean absolute error, MAE, and root mean square error, RMSE, are used as the primary evaluation metrics. In practical algorithm evaluation, MAE and RMSE are typically used together to assess predictive accuracy. Smaller MAE and RMSE values indicate more accurate predictions, whereas larger MAE and RMSE values indicate poorer performance and suggest that the model choice or parameter settings should be reconsidered.

Based on this principle, the present study uses MAE and RMSE to evaluate model performance. They are defined as follows:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|C_i - C_i^{GT}\right| \tag{13}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left|Ci - C_i^{GT}\right|^2} \tag{14}$$

Specifically, MAE reflects the accuracy of the model's predictions regarding the number of apples; a smaller value indicates that the model's predictions are closer to the actual values. RMSE is more sensitive to larger errors and reflects the dispersion and stability of the model's prediction errors.

Regarding the training settings, this paper uses the AdamW optimizer for parameter updates. During the training phase, input images undergo data augmentation processes such as random cropping and random flipping before being fed into the network. Considering the correspondence between the length of the positional encoding for high-level semantic features in the current implementation and the length of the feature sequence in the fourth stage, the training input cropping size is set to 512 in this chapter. This setting ensures consistency between the current model implementation and the feature dimensions, while also facilitating the stable completion of the Transformer encoding and subsequent density map regression processes. All other training

parameters are kept fixed under uniform experimental settings to ensure comparability among different models.

Our loss function design draws inspiration from a popular loss function widely used in the field of crowd counting [36], which consists of a weighted sum of the counting loss L_{count}, the optimal transport loss L_{OT}, and the total variance loss L_{TV}. Although L_{OT}performs well in dense regions, its performance in sparse regions is less than satisfactory. To address this issue, we introduce L_{TV}, which promotes smoother density maps by penalizing the differences between adjacent pixels. In low−density regions where the apple distribution is sparse, L_{OT}may be susceptible to outliers, leading to overfitting and producing unreasonable density estimates. The smoothing penalty of L_{TV} can effectively mitigate this overfitting, making the model's density estimates in low−density regions more stable. The total loss function for the predicted density map D and its corresponding ground truth D' is defined as shown in Equation (15):

$$L_{total} = L_{count}(P, G) + \lambda_1 L_{OT} + \lambda_2 L_{TV}(D, D') \quad (15)$$

Here, P and G represent the counts of D and D', L_{count} is used to constrain the consistency between the predicted total count and the actual total count, directly reflecting the model's error at the overall count level; L_{OT} is used to characterize the matching relationship between the predicted density distribution and the actual density distribution, thereby enhancing the model's ability to learn the spatial distribution structure of apples; L_{TV} suppresses local outliers through smoothing constraints, making the predicted density map more continuous and stable.λ_1、λ_2 are loss coefficients, set to 0.01 and 0.1, respectively, in our experiments. This joint loss design balances the requirements of both counting accuracy and reasonable density distribution, effectively improving the training stability and prediction quality of the apple counting model in complex scenarios.

This study implements the proposed apple-counting method, which utilizes a semantically guided and detail-aware Transformer, based on the PyTorch deep learning framework, and performs model training and testing on a high-performance GPU server. The specific configuration is shown in Table 1.

Table 1. Specific Configuration of the Experimental Environment

Name	Configuration Information	Name	Configuration Information
Operating System	Ubuntu20.04	Learning Framework	Pytorch1.12
CPU	Intel Xeon 6330*2	CUDA	11.1
RAM	32GB DDR4*16	CuDNN	8.3.2
GPU	NVIDIA A100 40GB*4	Torch	1.10
Python	3.8	Torchvision	0.11

4.3. Comparative Experiment

To comprehensively evaluate the performance of the proposed method in counting apples in complex orchard scenarios, this paper compares SDAFormer with several representative counting models, covering classical convolutional density map methods, point-supervised density map methods, and Transformer-based counting methods. The compared models include MCNN[33], CSRNet[34], Bayesian Loss[35], PCCNet [19], DM-Count[36], CCTrans [38], and CHS-Net[39]. All comparison methods were trained and tested using the same dataset split and evaluation metrics to ensure the fairness and comparability of the experimental results.

Table 2. Comparison of SDAFormer with Different Methods

Number	Method	MAE	RMSE
1	MCNN	9.52	11.83
2	CSRNet	5.85	7.52
3	PCCNet	5.98	8.01
4	BayesianLoss	4.29	5.94
5	DM-Count	3.76	5.01
6	CCTrans	3.72	4.98
7	CHS-Net	3.70	4.89
8	Ours	3.61	4.76

As shown in Table 2, the SDAFormer proposed in this paper achieves the best results on both MAE and RMSE metrics, with values of 3.61 and 4.76, respectively. Compared with classical convolutional density map methods such as MCNN, CSRNet, and PCCNet, our method demonstrates a clear advantage, indicating that relying solely on convolutional stacking makes it difficult to simultaneously capture the local visible structural features and adapt to scale variations in complex orchard images.

Compared to Bayesian loss-based point-supervised modeling methods, our approach also demonstrates strong competitiveness. While such methods indeed exhibit good training stability in point-supervised scenarios, they are still prone to issues such as false activation of local background and response sticking between adjacent apples in scenarios where apple and background textures are similar, local occlusions are prominent, and scale variations are significant—especially in the absence of stronger semantic guidance and explicit position modeling mechanisms. Compared to Transformer-based methods, SDAFormer also demonstrates advantages over the CCTrans model. This indicates that for apple counting tasks in complex scenarios, relying solely on global self-attention is insufficient to fully address issues such as severe occlusion and multi-scale variations. Compared to strong baselines such as DM-Count and CHS-Net, SDAFormer still achieves the best results.

As evidenced by the method design outlined in this chapter, SADRM utilizes high-level semantic information to guide low-level details during the feature representation stage, enabling more complete object representation in locally visible regions obscured by branches and leaves or by overlapping fruits; CAMM introduces coordinate-aware and multi-scale contextual modeling during the density map regression stage, granting the model stronger adaptability to apple distributions characterized by larger nearby objects and smaller distant ones, as well as the coexistence of sparse and dense regions. Consequently, SDAFormer's advantages are primarily evident in samples with significant occlusion and scale variations, which aligns with the subsequent visualization results showing more concentrated density responses and weaker adhesion between adjacent objects.

4.4. Ablation Studies

To further analyze the contribution of each component module to the overall performance, this paper first conducts ablation experiments on SADRM and CAMM. While keeping the backbone network, training strategy, and loss function consistent, we examine the impact on the SDAFormer

model's performance when introducing SADRM alone, CAMM alone, or both simultaneously. The results are shown in Table 3.

Table 3. Comparison of Experiments Across Different Modules

Number	SADRM	CAMM	MAE	RMSE
1	√	×	3.95	5.33
2	×	√	4.18	5.61
3	√	√	3.61	4.76

As shown by the overall module ablation results, when only SADRM is introduced, the model's MAE and RMSE are 3.95 and 5.33, respectively; when only CAMM is introduced, they are 4.18 and 5.61; and when both are introduced together, they further decrease to 3.61 and 4.76. These results indicate that, in the occlusion and scale variation scenarios addressed by the model, improving the quality of intermediate features for occluded apples is more critical. Therefore, after incorporating SADRM, the model's counting error decreases more significantly compared to the CAMM module. When CAMM is incorporated, its role is primarily to perform regression correction on scale variations and spatial distribution differences based on existing, relatively stable feature representations. When both modules are introduced simultaneously, the model achieves the best results, indicating that SADRM and CAMM are functionally complementary.

To analyze the roles of the coordinate encoding and multiscale convolution branches within CAMM, this paper conducted internal structure ablation experiments, ensuring that only the internal components of the CAMM module were compared while keeping the SADRM module unchanged. The specific results are shown in Tables 4.

Table 4. CAMM Internal Structure Ablation Experiment

Number	Coordinate encoding	Multiscale branching	MAE	RMSE
1	×	×	3.95	5.33
2	√	×	3.89	5.21
3	×	√	3.81	5.08
4	√	√	3.61	4.76

The results show that the multi-scale branching yields greater improvements than coordinate encoding alone; in scenes where the scale of apples varies significantly, modeling the context range has a more direct impact on density map regression. Coordinate encoding, on the other hand, incorporates positional prior knowledge, further enhancing the model's adaptability to the non-uniform spatial distribution in orchard images. The best results are achieved when the two are combined, indicating that CAMM is not a single-scale module but rather enhances structural stability in the regression stage through the synergy of position-aware and multi-scale context modeling.

In summary, SADRM primarily addresses the representation challenges caused by insufficient local visible information under occlusion conditions, while CAMM primarily addresses the regression challenges caused by scale variations and spatial distribution differences. The former operates at the feature representation stage, and the latter at the density regression stage; these two mechanisms are closely integrated sequentially within the network, jointly supporting the performance improvement of the SDAFormer model.

4.5. Analysis of Visualization Results

The SDAFormer was designed primarily to address two significant challenges in apple counting within real-world orchard environments: multi-scale variations and occlusion. Since the model is based on density map estimation, we compare the true density map with the predicted density maps from different models.

(1) Analysis of Counting Results for Various Models in Multi-Scale Variation Scenarios

As shown in Figure 4, the first row displays the original image; the second row shows the ground truth (GT) output of the real-world density map; the third and fourth rows present the results from two selected models; and the last row displays the predicted density map from our SDAFormer model.

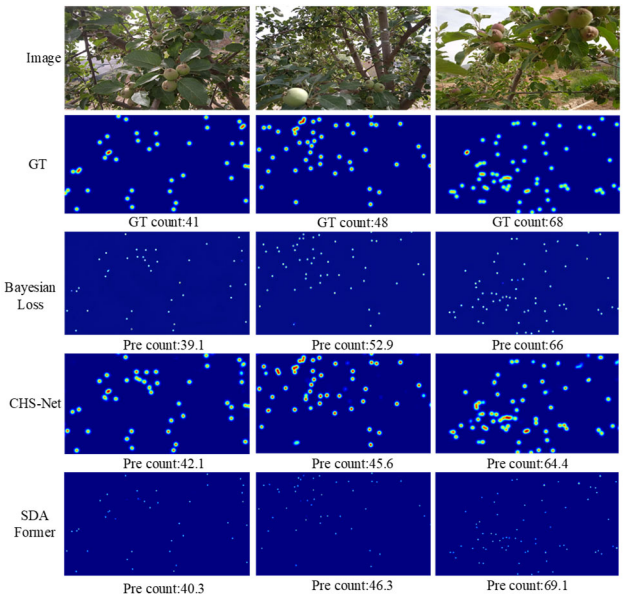

Figure 4. Comparison of Multiscale Changes

The visualization results demonstrate that different methods exhibit varying degrees of quality in density response within complex orchard scenarios. In multi-scale scenarios, when both large apples in the foreground and small apples in the background appear simultaneously, the Bayesian Loss method still struggles to accurately count small-scale targets, and response shifts occur in local regions; while CHS-Net can generate a relatively complete target distribution with count results close to the ground truth, it exhibits some response diffusion in multi-scale mixed regions, and the peak separation between adjacent fruits is not ideal; In contrast, SDAFormer produces more concentrated density peaks whose locations align more closely with ground truth annotations. Its predictions are generally closer to the true values, indicating that the model maintains good scale adaptability even when small apples in the background and large apples in the foreground coexist.

(2) Analysis of Counting Results for Various Models in Highly Occluded Scenarios

As shown in Figures 5, in highly occluded scenarios, Bayesian Loss tends to exhibit weak responses or localized omissions for severely occluded apples in complex foliage regions, which can easily lead to missed counts. Meanwhile, CHS-Net possesses some occlusion handling capabilities but is more prone to density peak diffusion and adhesion to adjacent objects. As the degree of occlusion increases, the over-smoothing phenomenon becomes more pronounced. SDAFormer maintains good peak separation between

adjacent or partially occluded apples while keeping activation levels low in the surrounding leaf regions. It preserves a clearer fruit distribution in densely occluded areas, with a spatial structure that more closely resembles the actual distribution of apples. Furthermore, the model's overall prediction error remains at a low level, demonstrating strong capabilities in handling highly occluded scenes.

Overall, the advantages of SDAFormer are not only reflected in its predictions being closer to the true values, but also in its relatively more reasonable density map structure, more concentrated local peaks, and fewer false responses in complex backgrounds. This aligns with the design objectives of SADRM to enhance semantic discrimination in occluded regions and CAMM to strengthen multi-scale and location-aware regression capabilities.

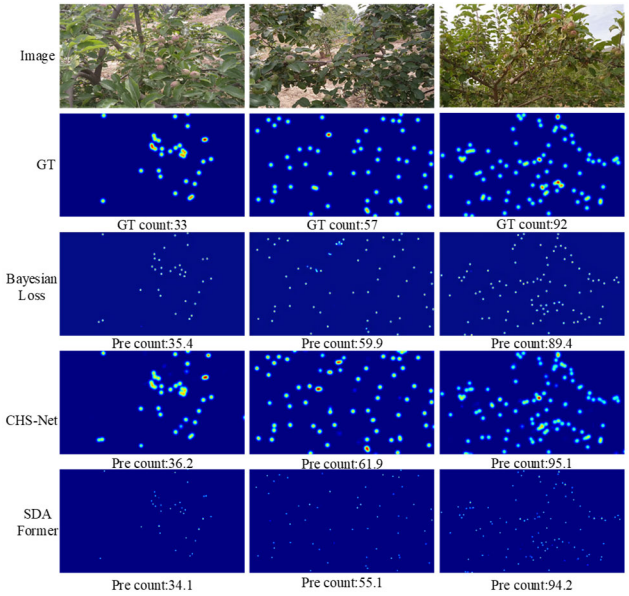

Figure 5. Comparison under occlusion

In Figures 6, we selected a local image from the top-right corner of the first original image in a multi-scale scene for analysis. It is evident that the apple in the top-right corner of the selected local image exhibits a significant difference in scale compared to the other apples. The counting performance of the various models varies to some extent when dealing with apple targets that undergo drastic scale changes. The Bayesian Loss model performs relatively poorly, failing to effectively identify the apple targets, whereas the CHS-Net and SDAFormer models demonstrate adaptability to scenarios with drastic scale changes and handle them well. Even under conditions of drastic scale changes, where the targets are slightly occluded by branches and leaves, the latter two models can still accurately identify and count them, exhibiting superior counting performance in such scenarios.

Also in Figure 7, we selected a partially occluded image from the lower-right corner of the second raw image in the high-occlusion scenario for analysis. In the selected partially occluded image, all four apple targets are obscured to some extent. For the two apples outside the red box, all three models effectively identified them and displayed them in the density map. Within the red-boxed area, the two apples in the upper center are partially occluded by other fruits and by surrounding branches and leaves. Under these conditions of occlusion by both fruit and foliage within the red box, there are significant differences in how these two apples are identified and displayed in the density map across the three models. The Bayesian Loss model accurately identified the upper apple within the red-boxed area, but the lower apple was clearly overlooked and omitted, indicating that the Bayesian Loss model performs averagely in handling occlusion in highly occluded scenarios. Meanwhile, the CHS-Net model identified both occluded fruit targets, but the density map exhibited density peak diffusion and adhesion to adjacent targets. The SDAFormer model, although exhibiting slight diffusion of density peaks in the density map, accurately identified the fruit within the red-boxed area, demonstrating good adaptability and processing capabilities in highly occluded scenarios.

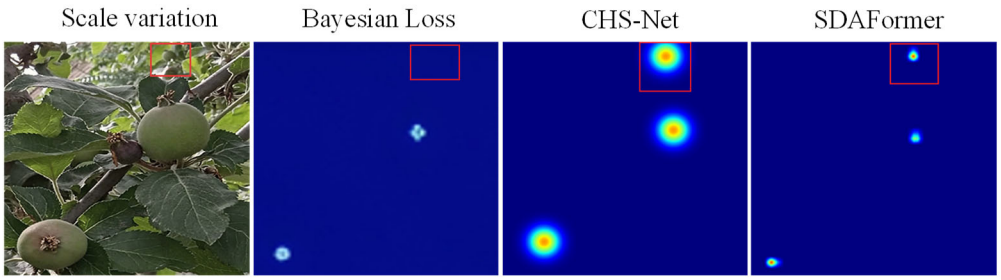

Figure 6. Analysis of Local-Scale Variations

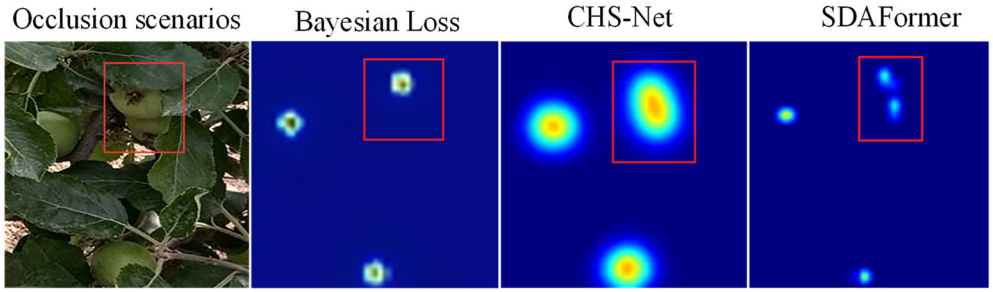

Figure 7. Partial Occlusion Analysis

5. Conclusion

This study addresses the problem of apple counting in complex scenes by proposing SDAFormer, an apple counting method based on a semantic-guided and detail-aware Transformer. To address the common issues of occlusion and

scale variations in complex orchards, this paper adopts PVT as the backbone network and constructs an integrated technical framework comprising SADRM and CAMM. SADRM enhances the network's semantic discrimination of apple targets by incorporating high-level semantic information into the shallow-level detail feature update process, thereby preserving edge, texture, and local structural information of the apples. CAMM improves the model's adaptability to spatial distribution variations and scale changes through explicit position encoding and multi-scale convolutional modeling, resulting in more stable and reasonable density map regression results.

Experiments demonstrate that SDAFormer outperforms mainstream models, achieving an MAE of 3.61 and an MSE of 4.76. The results indicate that the proposed SDAFormer model improves the accuracy and stability of apple counting under scale variations and occlusion conditions. However, since the training data primarily originates from a single orchard environment, the model's generalization capability in real-world, diverse, and complex scenarios requires further validation. Future research will focus on further diversifying data across regions to enhance the model's practicality and generalization capabilities.

References

[1] Villacrés J, Viscaino M, Delpiano J, et al. Apple orchard production estimation using deep learning strategies: a comparison of tracking-by-detection algorithms[J]. Computers and Electronics in Agriculture, 2023, 204: 107513. DOI:10.1016/j.compag.2022.107513.

[2] He L, Fang W, Zhao G, et al. Fruit yield prediction and estimation in orchards: a state-of-the-art comprehensive review for both direct and indirect methods[J]. Computers and Electronics in Agriculture, 2022, 195: 106812. DOI:10.1016/j.compag.2022.106812.

[3] Schmitz C, Zimmermann L, Schiffers K, et al. ProbApple: a probabilistic model to forecast apple yield and quality[J]. Agricultural Systems, 2025, 208: 104298. DOI:10.1016/j.agsy.2025.104298.

[4] Chen S, Zhang S, Li H, et al. Optimizing irrigation and nitrogen management enhances apple yield and quality through improving soil quality on the Loess Plateau[J]. Plant and Soil, 2025, 489(1): 255-271. DOI:10.1007/s11104-025-07712-z.

[5] Ahmed D, Sapkota R, Churuvija M, et al. Machine vision-based crop-load estimation using YOLOv8[EB/OL]. arXiv preprint: arXiv:2304.13282, 2023. DOI:10.48550/arXiv.2304.13282.

[6] Bhusal S, Bhattarai U, Karkee M. Trellis wire detection for obstacle avoidance in apple orchards[J]. IFAC-PapersOnLine, 2022, 55(32): 72-77. DOI:10.1016/j.ifacol.2022.11.117.

[7] Rong J, Zhang H, Zhou F, et al. Tomato cluster detection and counting using improved YOLOv5 based on RGB-D fusion[J]. Computers and Electronics in Agriculture, 2023, 207: 107741. DOI:10.1016/j.compag.2023.107741.

[8] Yu X, Wang Y, An D, et al. Counting method for cultured fishes based on multi-modules and attention mechanism[J]. Aquacultural Engineering, 2022, 96: 102215. DOI:10.1016/j.aquaeng.2021.102215.

[9] Wu Z, Sun X, Jiang H, et al. NDMFCS: an automatic fruit counting system in modern apple orchard using abatement of abnormal fruit detection[J]. Computers and Electronics in Agriculture, 2023, 211: 108036. DOI:10.1016/j.compag.2023.108036.

[10] Yan Z, Wu Y, Zhao W, et al. Research on an apple recognition and yield estimation model based on the fusion of improved YOLOv11 and DeepSORT[J]. Agriculture, 2025, 15(7): 765. DOI:10.3390/agriculture15070765.

[11] Sapkota R, Meng Z, Churuvija M, et al. Comprehensive performance evaluation of YOLOv12, YOLO11, YOLOv10, YOLOv9 and YOLOv8 on detecting and counting fruitlet in complex orchard environments[EB/OL]. arXiv preprint: arXiv:2407.12040, 2024. DOI:10.48550/arXiv.2407.12040.

[12] Cao D, Luo W, Tang R, et al. Research on apple detection and tracking count in complex scenes based on the improved YOLOv7-Tiny-PDE[J]. Agriculture, 2025, 15(5): 483. DOI:10.3390/agriculture15050483.

[13] Häni N, Roy P, Isler V. Apple counting using convolutional neural networks[C]//Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems. 2018: 2559-2565. DOI:10.1109/IROS.2018.8594304.

[14] Wang Q, Nuske S, Bergerman M, et al. Detection and localization of overlapped fruits: application in an apple harvesting robot[J]. Electronics, 2020, 9(6): 1023.

[15] Zhang S, Wu X, You Z, et al. A method of apple image segmentation based on color-texture fusion feature and machine learning[J]. Agronomy, 2020, 10(7): 972.

[16] Fan P, Lang G, Yan B, et al. A method of segmenting apples based on gray-centered RGB color space[J]. Remote Sensing, 2021, 13(6): 1211.

[17] LeCun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11): 2278-2324.

[18] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition[C]//International Conference on Learning Representations. 2015.

[19] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016: 770-778.

[20] Ren S, He K, Girshick R, et al. Faster R-CNN: Towards real-time object detection with region proposal networks[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(6): 1137-1149.

[21] Redmon J, Divvala S, Girshick R, et al. You only look once: Unified, real-time object detection[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016: 779-788.

[22] Gao F, Wu Z, Suo R, et al. Apple detection and counting using real-time video based on deep learning and object tracking[J]. Transactions of the Chinese Society of Agricultural Engineering, 2021, 37(21): 217-224.

[23] Zhao J, et al. Research on apple recognition algorithm in complex orchard environment based on deep learning[J]. Sensors, 2023, 23(12): 5425.

[24] Hu Y, et al. Fruit detection and counting in apple orchards based on improved Yolov7 and multi-object tracking methods[J]. Sensors, 2023, 23(13): 5903.

[25] Abeyrathna R M R D, Nakaguchi V M, Minn A, et al. Recognition and counting of apples in a dynamic state using a 3D camera and deep learning algorithms for robotic harvesting systems[J]. Sensors, 2023, 23(8): 3810.

[26] Yang X, et al. Automatic apple detection and counting with AD-YOLO and MR-SORT[J]. Sensors, 2024, 24(21): 7012.

[27] Matos R, de Belen R A J, Perez T, et al. Tracking and counting apples in orchards under intermittent occlusions and low frame rates[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. 2024: 4681-4690.

[28] Jin T, Han X, Wang P, et al. Enhanced deep learning model for apple detection, localization, and counting in complex orchards for robotic arm-based harvesting[J]. Smart Agricultural Technology, 2025, 10: 100784.

[29] Cao D, Luo W, Tang R, et al. Research on apple detection and tracking count in complex scenes based on the improved YOLOv7-Tiny-PDE[J]. Agriculture, 2025, 15(5): 483.

[30] Wang X, Tang J, Whitty M A. DeepPhenology: Estimation of apple flower phenology distributions based on deep learning[J]. Computers and Electronics in Agriculture, 2021, 184: 106123.

[31] Bhattarai U, Karkee M. A weakly-supervised approach for flower/fruit counting in apple orchards[J]. Computers in Industry, 2022, 138: 103635.

[32] Lempitsky V, Zisserman A. Learning to count objects in images[C]//Advances in Neural Information Processing Systems 23. 2010: 1324-1332.

[33] Zhang Y, Zhou D, Chen S, et al. Single-image crowd counting via multi-column convolutional neural network[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016: 589-597.

[34] Li Y, Zhang X, Chen D. CSRNet: Dilated convolutional neural networks for understanding the highly congested scenes[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 1091-1100.

[35] Ma Z, Wei X, Hong X, et al. Bayesian loss for crowd count estimation with point supervision[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019: 6142-6151.

[36] Wang B, Liu H, Samaras D, et al. Distribution matching for crowd counting[C]//Advances in Neural Information Processing Systems 33. 2020.

[37] Gao J, Wang Q, Li X. PCC Net: perspective crowd counting via spatial convolutional network[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2020, 30(10): 3486-3498.

[38] Tian Y, Chu X, Wang H. CCTrans: simplifying and improving crowd counting with transformer[EB/OL]. arXiv:2109.14483, 2021.

[39] Dai M, Huang Z, Gao J, et al. Cross-head supervision for crowd counting with noisy annotations[C]//ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing. Piscataway, NJ: IEEE, 2023: 1-5.