agriculture

MDPI

Article

DCFA-YOLO: A Dual-Channel Cross-Feature-Fusion Attention YOLO Network for Cherry Tomato Bunch Detection

Shanglei Chai [1], Ming Wen [1], Pengyu Li [1], Zhi Zeng [2] and Yibin Tian [1,*]

[1] College of Mechatronics and Control Engineering, Shenzhen University, Shenzhen 518060, China; 2450096004@mails.szu.edu.cn (S.C.); wenming@szu.edu.cn (M.W.); 2310294020@email.szu.edu.cn (P.L.)
[2] School of Computer and Information Science, Chongqing Normal University, Chongqing 401331, China; zzh406@cqnu.edu.cn
* Correspondence: ybtian@szu.edu.cn

Abstract: To better utilize multimodal information for agriculture applications, this paper proposes a cherry tomato bunch detection network using dual-channel cross-feature fusion. It aims to improve detection performance by employing the complementary information of color and depth images. Using the existing YOLOv8_n as the baseline framework, it incorporates a dual-channel cross-fusion attention mechanism for multimodal feature extraction and fusion. In the backbone network, a ShuffleNetV2 unit is adopted to optimize the efficiency of initial feature extraction. During the feature fusion stage, two modules are introduced by using re-parameterization, dynamic weighting, and efficient concatenation to strengthen the representation of multimodal information. Meanwhile, the CBAM mechanism is integrated at different feature extraction stages, combined with the improved SPPF_CBAM module, to effectively enhance the focus and representation of critical features. Experimental results using a dataset obtained from a commercial greenhouse demonstrate that DCFA-YOLO excels in cherry tomato bunch detection, achieving an mAP50 of 96.5%, a significant improvement over the baseline model, while drastically reducing computational complexity. Furthermore, comparisons with other state-of-the-art YOLO and other object detection models validate its detection performance. This provides an efficient solution for multimodal fusion for real-time fruit detection in the context of robotic harvesting, running at 52fps on a regular computer.

Keywords: cherry tomato bunch detection; robotic harvesting; multimodal image; feature extraction; feature fusion; YOLO network

Academic Editor: Dimitre Dimitrov

Received: 5 January 2025
Revised: 22 January 2025
Accepted: 24 January 2025
Published: 26 January 2025

Citation: Chai, S.; Wen, M.; Li, P.; Zeng, Z.; Tian, Y. DCFA-YOLO: A Dual-Channel Cross-Feature-Fusion Attention YOLO Network for Cherry Tomato Bunch Detection. *Agriculture* **2025**, *15*, 271. https://doi.org/10.3390/agriculture15030271

1. Introduction

With the aging of the global population and the increasing shortage of agricultural labor, modern agriculture is facing unprecedented challenges [1]. Against this backdrop, the rapid development of smart agricultural technology becomes an important driving force for promoting agricultural transformation. As a popular fruit, cherry tomatoes play an important role in global agricultural production due to their appeasing taste and richness in lycopene [2]. However, due to the small size and uneven distribution, their harvesting faces significant challenges. Existing mechanical picking equipment often misses fruits or mistakenly picks leaves because it is difficult to accurately identify the location of the fruit, making it difficult to effectively improve robotic harvesting efficiency and

quality. Therefore, developing efficient and intelligent identification and position detection for cherry tomatoes is the key to improving production efficiency and fruit quality.

Fruit detection is one of the core components of fruit-harvesting robots [3]. In recent years, Deep Neural Networks (DNNs) have made significant progress in fruit detection. These algorithms no longer rely on manually designed features and can automatically learn complex patterns in images, effectively overcoming the limitations of traditional fruit detection methods when facing complex lighting and background interference. Object detection methods based on deep learning are mainly divided into single-stage detection and two-stage detection. The single-stage detection directly regresses and predicts the bounding box of the target. Its advantages are a fast detection speed and high efficiency, and it is one of the most widely used methods. Typical single-stage detection methods include the YOLO series and the SSD network [4–6]. For example, Lyu et al. improved the backbone network and loss function of YOLOv5 in YOLOv5-CS for the detection and counting of green citrus in orchard environment [7]. Wang et al. used ShuffleNet v2 as an improved backbone and introduced the CBAM attention mechanism to improve detection accuracy and reduce the model size [8]. Gai et al. improved YOLOv4 by combining it with DenseNet for the maturity detection of small tomatoes [9]. Zhao et al. achieved an accurate detection of grapes and picking points by improving YOLOv4, where the average precision of grape detection reached 93.27% [10]. Another popular single-stage detection method is the SSD model by Yuan et al. [11], who verified its performance on cherry tomatoes using different backbone networks (VGG16, MobileNet, Inception V2). In addition, Fuentes-Peñailillo et al. proposed a seedling counting model that combines traditional image processing with MobileNet-SSD, achieving a maximal precision of 96% [12]. In contrast, two-stage detection processes candidate region extraction and classification in two steps. Although the detection speed is slower, it performs better in precision. Typical two-stage detection methods include the RCNN series [13–15] and SPPNet [16]. For example, Hu et al. used Faster R-CNN combined with color space conversion and fuzzy set method to realize bounding box detection and the segmentation of tomatoes [17]. It performed well in cases where the fruit edges were blurred or overlapped. Song et al. built a Faster R-CNN model based on VGG16 and achieved kiwifruit detection under different lighting conditions [18]. The average detection accuracy reached 87.61%. Gao et al. used Faster R-CNN to detect and classify apples under four different growth conditions, with an average precision of 87.9% [19].

Although single-modal visual data processing based on deep learning has made significant progress in fruit detection, they still face many challenges in complex environments. First, single-modal methods (e.g., RGB only detection) are highly susceptible to illumination changes, occlusion, and background clutter, leading to insufficient detection accuracy and robustness. For instance, in low-light conditions or when fruits are partially occluded by leaves, the performance of these methods degrades significantly. Second, most existing models rely heavily on large-scale annotated datasets, which are labor-intensive and time-consuming to acquire, especially for small and densely distributed fruits like cherry tomatoes. Third, while some studies have attempted to improve detection accuracy by increasing model complexity, this usually comes at the cost of increased computational cost, making them unsuitable for real-time applications, such as robotic harvesting. To address these issues, multimodal fusion has gradually become an important research direction to meet these challenges. Especially when picking, robots are usually equipped with multiple sensors, such as color and depth cameras. Fusing information from different modalities can not only improve the precision of fruit detection but also enhance the system's ability to adapt to environmental changes. In recent years, multimodal methods combining color (RGB) and depth images have been widely used in fruit

detection and made remarkable progress. Tu et al. developed a fruit and maturity detection model using RGB-D images [20]. Their ablation experiment showed that the introduction of depth improved the detection accuracy by 3.03%. Cui et al. proposed a single cherry tomato detection method that combines RGB-D inputs. The RGB image is converted into the LAB color space and then fused with the depth map and normal vector map obtained from the point cloud as the inputs of an improved YOLOv7 to detect cherry tomatoes [21,22]. Similarly, Rong et al. optimized YOLOv5 by fusing RGB and depth images to improve the detection performance of cherry tomato clusters/bunches [23]. In addition, Kaukab et al. used multimodal data as the input of YOLOv5 and effectively reduced the impact of depth image noise via a deep fusion method of non-targeted background removal and achieved an precision of 96.4% for apple detection [24].

The fusion methods of multimodal images can be divided into three types, early fusion, mid-term fusion, and late fusion, according to its introduction at different processing stages. In early-fusion methods, images from different modalities are directly integrated at the image level before being fed into the network. For example, Liu et al. significantly improved the average precision of kiwifruit detection to 90.7% by superimposing RGB and near-infrared images to form a four-dimensional tensor as the input of the VGG16 [25]. However, this method is prone to pixel misalignment due to offset between images of different modalities. In contrast, late fusion linearly combines independent predictions at the decision stage. Sa et al. combined the detection results of RGB and NIR images in the last step of the shared feature extraction network Faster R-CNN and improved the detection accuracy from 81.6% to 83.8% [26], especially effective for fruits with similar colors, such as green peppers and melons. However, in late fusion, different modality features cannot learn from each other, resulting in poor interactions between different network branches. As a feature-level fusion method, mid-term fusion combines multimodal feature maps and can effectively balance the depth and efficiency of feature integration. For example, Wei et al. designed a multi-branch backbone network that includes color, infrared, and polarization image inputs [27]. By using operations such as feature connection, dimensionality reduction, and activation, they achieved higher target-recognition precision in complex environments. However, it also significantly reduces the inference speed.

Despite the progress in multimodal fusion, several challenges remain unresolved. First, early-fusion methods often suffer from pixel misalignment due to the spatial offset between RGB and depth images, leading to inaccurate feature extraction. Second, late-fusion methods lack effective interaction between modalities, as they process each modality independently until the final decision stage, which limits their ability to leverage complementary information. Third, mid-term-fusion methods, while achieving better feature integration, often introduce high computational complexity, making them unsuitable for real-time applications such as robotic harvesting. Additionally, most existing multimodal fusion methods do not fully exploit the potential of attention mechanisms to enhance the saliency of critical features and suppress irrelevant information, which is crucial for improving detection accuracy in complex environments. These limitations underscore the need for a more efficient and robust multimodal fusion approach that can address these challenges while maintaining real-time performance.

In order to further optimize the performance of multimodal fusion, in recent years, researchers began to introduce attention mechanisms to highlight critical information and reduce the interference of irrelevant features. Woo et al. introduced self-learning weight parameters in the CBAM module, which effectively improves the weight of the region of interest and suppresses invalid features [28]. Li et al. proposed a solution using DenseBlock combined with the attention mechanism to achieve image denoising when fusing RGB and NIR images for multimodal segmentation tasks [29]. Inspired by their work, this

paper proposes a dual-channel cross-fusion attention YOLO (DCFA-YOLO) network to meet the needs of accurate and fast multimodal cherry tomato bunch detection. Specifically, we introduce a parallel attention mechanism in the mid-term-fusion stage. By enhancing the learning ability and saliency of the region of interest, it overcomes the interference introduced by pixel offset in early fusion and the high independence of each modality feature in late fusion. DCFA-YOLO integrates multimodal data of color and depth images and uses dual-channel cross-fusion and attention mechanisms to effectively extract and fuse features of different modalities. At the same time, the model is designed with the goal of being lightweight for real-time applications. Even when multimodal fusion is used, the size of its model parameters remains similar to that of the single-modal method, ensuring its high efficiency and ease of deployment. The design of this model not only enhances the ability to understand and process multimodal data but also takes into account the conservation of computing resources. It provides a reliable and efficient solution for precision agriculture and intelligent fruit harvesting. The contributions of this paper can be summarized as follows:

- A new feature fusion method is developed to combine different modal features effectively. It integrates a dual-channel cross-fusion attention mechanism into YOLO to enhance the fusion of color and depth images in a balanced way, significantly improving fruit detection accuracy and robustness.
- An efficient lightweight design is proposed by replacing the C2f module with the ShuffleNetV2 unit, optimizing the backbone network for faster and more efficient early feature extraction, while reducing computational complexity.
- The attention mechanism is enhanced by introducing a SPPF_CBAM unit to the early feature extraction stage, improving the model's ability to focus on key features through dynamic channel and spatial attention.
- The proposed DCFA-YOLO has been evaluated for cherry tomato bunch detection using a dataset obtained from a commercial greenhouse, achieving an mAP50 of 96.5%, outperforming multiple YOLO models while being relatively lightweight.

The paper is organized as follows: Section 1 provides an overview of related work and existing challenges in fruit detection as well as multimodal object detection. Section 2 describes the proposed DCFA-YOLO method in detail. Section 3 experimentally verifies and analyzes the effectiveness of DCFA-YOLO. Section 4 discusses the results and implications and potential directions for future research, while Section 5 concludes the work.

2. Proposed Methods

The proposed DCFA-YOLO is based on the YOLOv8_n model [30]. Through targeted improvements in the backbone network, feature fusion mechanism, and attention mechanism, the feature extraction and fusion capabilities of multimodal images were significantly improved. At the same time, the computational complexity of the model was effectively reduced. In view of the long-tail characteristics of the target distribution of fruits in complex harvesting environments, the Distribution Focus Loss (DFL) was employed to optimize the model's ability to focus on targets of different sizes. This provided an efficient solution for the multimodal fruit detection task. The overall structure of DCFA-YOLO is shown in Figure 1.

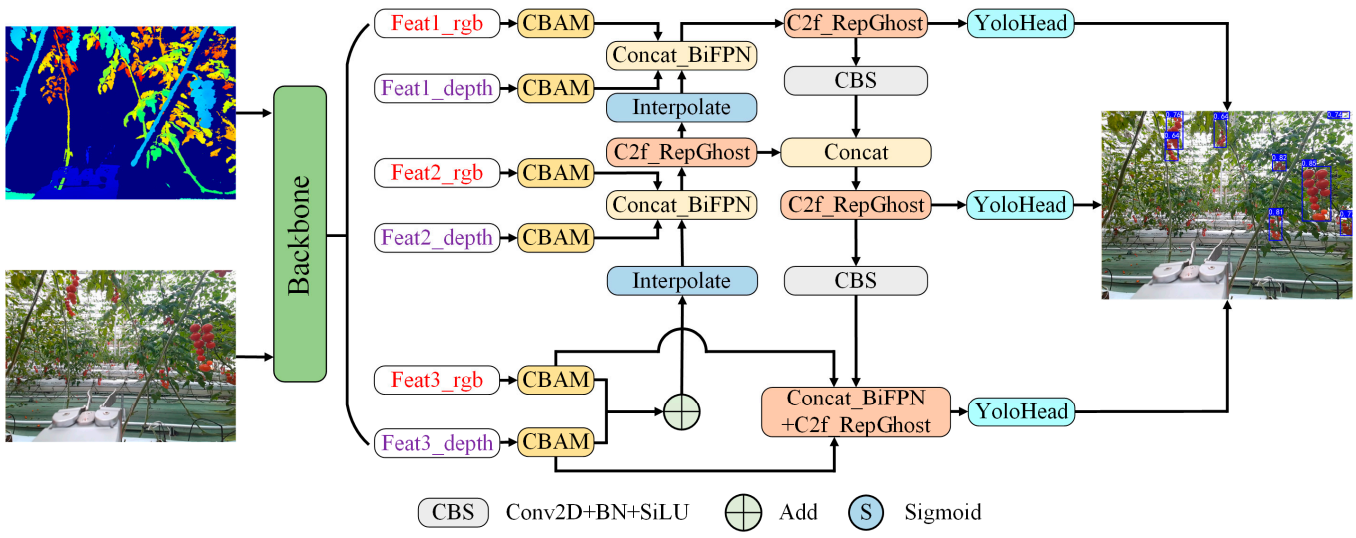

Figure 1. The overall architecture of the proposed DCFA-YOLO.

2.1. Cross-Fusion Mechanism

The input to the cross-fusion mechanism consisted of three groups of features initially extracted by the backbone network at different depths (Figure 2): two initial modality features (RGB and depth) and one additional set of features obtained by adding Feat3_rgb and Feat3_depth, followed by upsampling. These three groups served as the basic feature inputs for the cross-fusion process. To enhance the semantic information of these features, the two Feat3 basic feature layers were first added, which were upsampled and cross fused with other two sets of feature layers sequentially. Since the relatively shallow Feat1 and Feat2 contained more detailed information, we directly fused the independent upsampled RGB and depth feature layers using the Concat_BiFPN module.

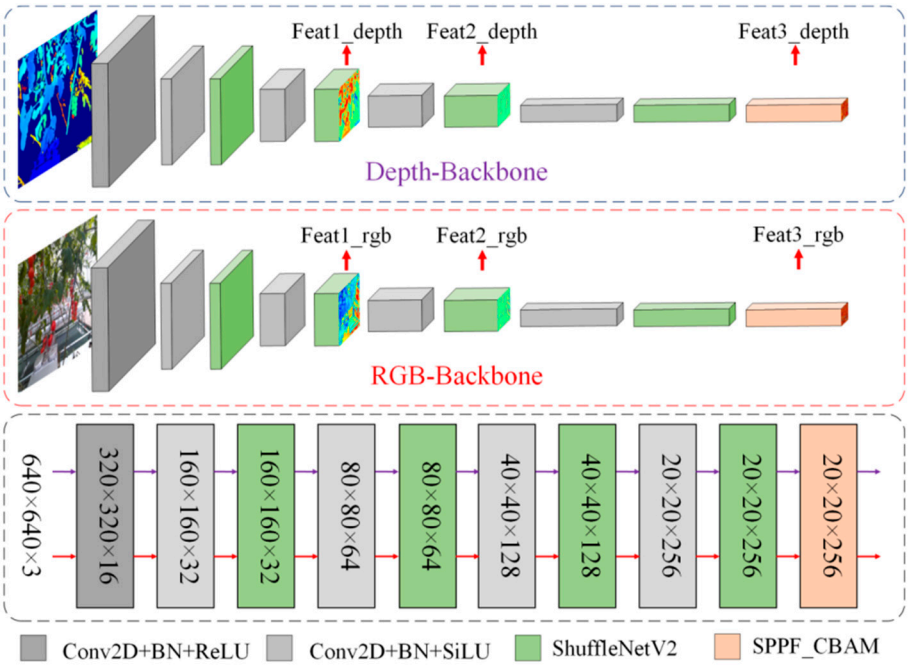

Figure 2. The structure of the backbone network.

The Concat_BiFPN module has many advantages over the traditional BiFPN_Add and simple Concat concatenation [31]. First, unlike BiFPN_Add, Concat_BiFPN not only implements dynamic weighting of features through learnable weight parameters but also

retains the independence of each input feature. The BiFPN_Add directly sums the input features after weighting, which may lead to the loss of modal feature details despite the tight fusion. Concat_BiFPN, on the other hand, combines features along the specified dimension by weighting, which can not only integrate multimodal information but also retain the differences of the original features. This provides a richer and more complete feature expression for subsequent processing. Second, compared with the single Concat operation, Concat_BiFPN introduces a dynamic weighting mechanism, which enables the contribution of features from each modality to be adaptively adjusted according to task requirements. This overcomes the feature imbalance problem in simple concatenation. Third, Concat_BiFPN supports a flexible adjustment of the dimension and number of input features, making it more adaptable.

The formula for the Concat_BiFPN module is

$$\hat{x} = \text{concat}\left(\overline{w_1} \cdot x_1, \overline{w_2} \cdot x_2, \overline{w_3} \cdot x_3, \text{dim}\right) \quad (1)$$

where x_1, x_2, and x_3 are input feature maps from different modalities; $\overline{w_1}$, $\overline{w_2}$, and $\overline{w_3}$ are learned weights used to adjust the contribution of different modal features; and dim is the concatenated dimension, which can usually be selected as the channel dimension of the feature map.

$$[\overline{w_1}, \overline{w_2}, \overline{w_3}] = [w_1, w_2, w_3]/(w_1 + w_2 + w_3 + \epsilon) \quad (2)$$

where $w = [w_1, w_2, w_3]$ is the weight vector, ϵ is a small constant used to avoid division by zero, and the value 3 refers to the three input modalities (RGB, depth, and the fused feature) that are being combined in the Concat_BiFPN module. The normalization ensures their contributions are adjusted appropriately during fusion.

It is worth noting that we used bilinear interpolation in the upsampling of feature cross fusion. This allowed for smoother feature map enlargement with more information preservation.

2.2. Model Simplifications

Compared with the networks using a single-modal input, DCFA-YOLO uses multimodal cross fusion and dual-channel feature extraction, leading to a significant increase in the amount of computation. To effectively reduce the computational complexity and parameter size while maintaining high-precision detection performance, we propose targeted optimizations. Specifically, we replaced the C2f module in the backbone with the ShuffleNetV2 unit, the C2f module in cross-feature fusion was replaced by the C2f_RepGhost. These modifications were designed to balance computational efficiency with high-quality feature extraction by considering the distinct roles of the backbone and feature fusion stages.

As shown in Figure 2, the backbone w responsible for preliminary feature extraction. Its goal is to extract global features from the input multimodal data. In order to reduce the computational complexity at this stage, C2f was replaced by the ShuffleNetV2 unit, as shown in Figure 3a. ShuffleNetV2 is a lightweight convolutional neural network that significantly reduces the amount of computation and number of parameters through sophisticated techniques such as Channel Split, grouped convolutions, and multi-scale feature fusion. In addition, Channel Shuffle helped to enhance feature representation by breaking information isolation in group convolutions, resulting in more effective feature extraction with reduced computational overhead. These optimizations allowed for ShuffleNetV2 to significantly reduce GFLOPs and parameter size while maintaining relatively efficient feature extraction.

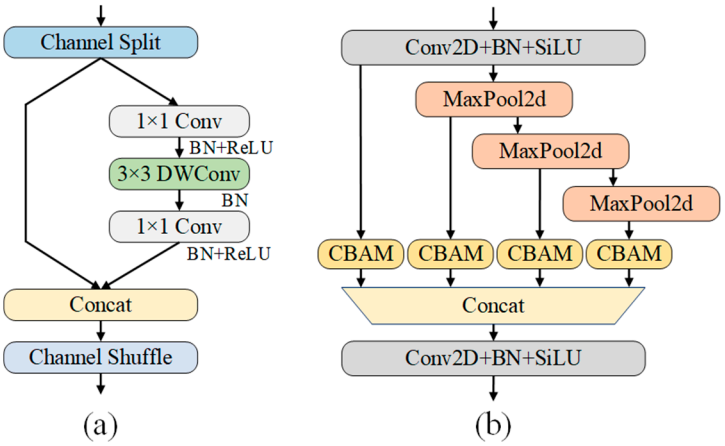

Figure 3. (**a**) The structure of the ShuffleNetV2 unit; (**b**) the structure of the SPPF_CBAM unit.

The feature-cross-fusion stage served as the neck of the network, a crucial component for enhancing feature extraction and improving model performance. As shown in Figure 4, the C2f module was replaced by the C2f_RepGhost module in the feature-cross-fusion stage to achieve efficient feature extraction. It combined depth-wise convolution (DWConv) for spatial feature extraction and pointwise convolution (PWConv) for channel interaction, improving feature expression capabilities while reducing computation. In addition, the module introduced shortcut connections to achieve residual learning, mitigating gradient vanishing during training and promoting feature reuse. Moreover, re-parameterization techniques during the inference phase merged the module's complex structures into efficient inference configurations, further reducing the computational complexity for inference. Through these designs, the C2f_RepGhost module enhanced the deep-level feature expression capability while ensuring computational efficiency.

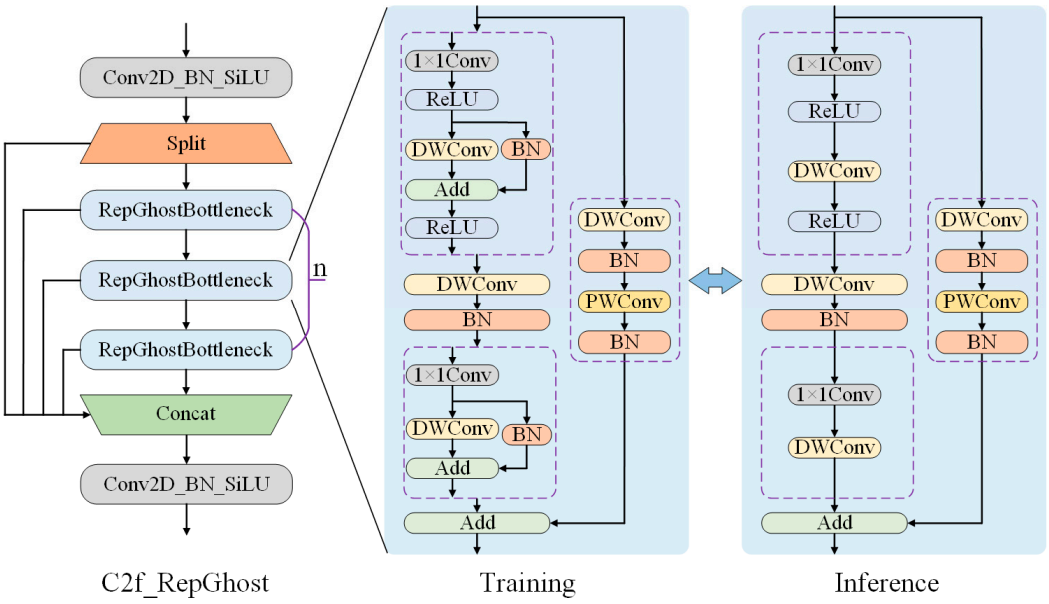

Figure 4. The structure of the C2f_RepGhost module. The module adds skip links during the training phase compared to the inference phase.

These targeted lightweight improvement strategies consider the characteristics of each stage to maximize the respective functions of the backbone and feature fusion. It not only meets the lightweight requirements of preliminary feature extraction but also meets the high performance requirements of enhanced feature fusion.

2.3. Attention Mechanism

To further improve the model's ability to extract and fuse multimodal features, an attention mechanism was introduced into the backbone network. The improvements were reflected in the optimization of the SPPF module and the enhanced attention in the early feature extraction layer. At the last backbone layer, the original SPPF module was replaced by the SPPF_CBAM, as shown in Figure 3b. The details of the CBAM module are shown in Figure 5. Through the parallel channel and spatial attention modules, the weights of different channels and spatial positions in the feature map can be adaptively adjusted. This strengthens key features and suppresses redundant information, improving the feature expression ability of the model.

In addition, to fully utilize the advantages of multimodal inputs, the model added CBAM modules to the early-feature layers of RGB and depth maps in the backbone network. This allowed for the features of each modality to be enhanced through the attention mechanism at early stage. The channel attention captured the importance of different feature channels, while the spatial attention focused on the salient regions in the feature map. Ultimately, these enhanced multimodal feature layers were able to fully complement and interact with each other during the fusion stage. This effectively improved the robustness of feature extraction, providing more accurate features for subsequent detection tasks.

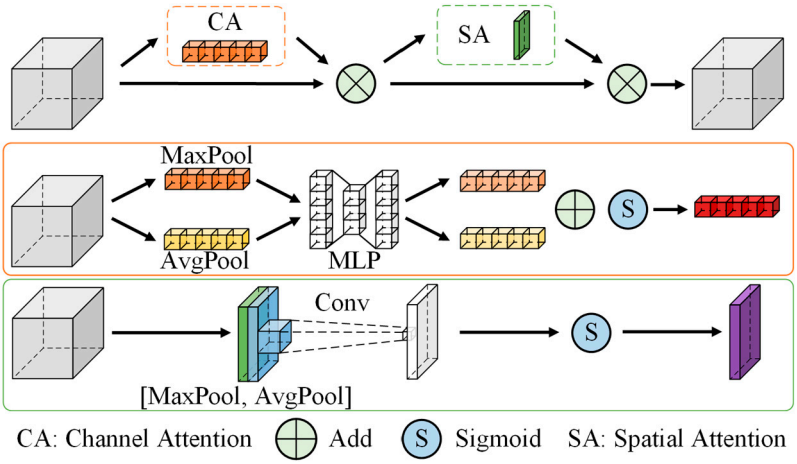

Figure 5. The structure of the CBAM module.

2.4. Loss Function

As shown in Figure 6, statistical analysis of the cherry tomato dataset (described below) revealed that the fruit sizes followed a significant long-tail distribution. Among the 22,965 annotated bounding boxes, the average object width and height were 48.2 and 79.9 pixels, while the maximal width and height reached 223 and 387 pixels, respectively, and the minima were close to zero. Looking further at the distribution of widths, 55.47% of the target widths were concentrated in [24.8, 49.6] pixels, while only 0.02% of them exceeded 198.2 pixels. Likewise, 50.44% of the target heights were within [43.9, 86.8] pixels, and only 0.02% of them were above 344.1 pixels. In terms of area distribution, 88.42% of the targets were below 8319 pixels, while only 0.01% of them were above 66,549 pixels. These statistics indicated that a large number of objects had small sizes, while a very small number of objects had significantly larger sizes, resulting in an imbalanced distribution, which poses significant challenges to traditional object detection models, especially in balancing the high detection rate of small objects and the high position precision of large ones.

To address the above problem, we used Distribution Focal Loss (DFL) in the bounding box loss function [32]. Unlike traditional methods that predict bounding boxes as dis-

crete values, DFL represents the bounding box locations as continuous probability distributions. This can more accurately describe the uncertainty of the target location and optimize the model's attention to targets of different sizes. Specifically, DFL converts the predicted bounding boxes into a probability distribution and applies a focal loss on it to address the data class imbalance. It is able to strike a balance between the small objects that dominate the dataset and the large objects that are rare but important. Specifically, the formula of DFL is

$$\mathrm{DFL}(S_i, S_{i+1}) = -\big((y_{i+1} - y)\log(S_i) + (y - y_i)\log(S_{i+1})\big) \quad (3)$$

where y_i and y_{i+1} are two adjacent discrete points of the predicted bounding box, y is the true value, and S_i and S_{i+1} are the predicted probabilities corresponding to y_i and y_{i+1}.

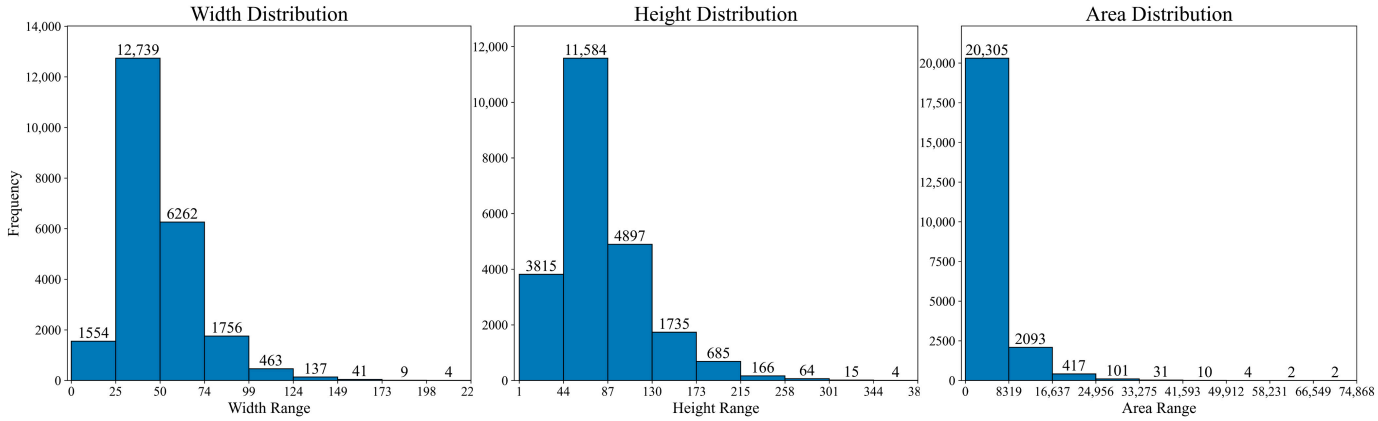

Figure 6. Statistical analyses of the cherry tomato dataset used in the current study (unit: pixel).

3. Experiments and Results

To verify the effectiveness and performance of the proposed DCFA-YOLO, a series of ablation and comparative experiments were designed and implemented. Through ablation experiments, the impact of each component on model performance is analyzed. At the same time, the full DCFA-YOLO is compared with a few state-of-the-art YOLO networks to comprehensively evaluate its performance in terms of detection accuracy, recall, computational complexity, and model parameter quantity.

3.1. Experimental Setup and Dataset

The cherry tomato images were acquired at a greenhouse in a plant science and technology park in Dongguan, China (Figure 7). A Microsoft Azure Kinect DK (Microsoft, Redmond, WA, USA) or Litemaze TOF camera (Litemaze Technology, Shenzhen, China) installed at the end of the robotic arm collected 843 RGB-D image pairs, including RGB and corresponding depth images. The purpose was to simulate the posture and distance of the robotic arm during the picking process. More detailed descriptions of the image acquisition environment and procedures have been reported in a previous study on single cherry tomato detection [22]. The original data were expanded using data augmentation methods such as cropping, flipping, and random brightness enhancement, and finally, 3372 images were obtained. Cherry tomato bunches within 1 m of the camera were labeled using LabelImg (version 1.8.1). Finally, the dataset was divided into the training set (2700 images), the validation set (336 images), and the test set (336 images).

All experimental were carried out on a desktop workstation with Intel(R) Core (TM) i7-14700K CPU (Intel, Santa Clara, CA, USA), 20 cores and 28 threads, 64G memory, and Nvidia GeForce RTX 4070 Ti SUPER GPU (Nvidia, Santa Clara, CA, USA), and the Ubuntu

22.04 operating system. All deep learning algorithms were executed in the same environment of Windows 11, Cuda 12.6, Python 3.8, Pytorch-2.4.1, and Torchvision-0.19.1. All experiments set the image size to 640 × 640, batch_size to 8, and epochs to 200.

The initial learning rate was set to 0.01, with the minimal learning rate reduced to 1% of the initial value. The optimization utilized the Stochastic Gradient Descent (SGD) optimizer, configured with a momentum of 0.937 and a weight decay factor of 0.0005 to regularize the model parameters. The learning rate scheduling followed a cosine decay strategy, dynamically adjusting the learning rate throughout the training process to enhance convergence and performance.

Figure 7. The dataset collection scene. (**a**) Early harvesting season. (**b**) Late harvesting season.

3.2. Evaluation Metrics

We used precision, recall, F1-score, and mAP as the basic detection performance evaluation metrics. The mean average precision (mAP) is an important indicator to measure the global detection performance of the model. AP is defined as the area under the precision–recall curve, which represents the comprehensive evaluation of single-category detection performance [33]. mAP50 represents the average precision value calculated when the Intersection over Union (IoU) threshold is 0.5 and is a commonly used evaluation criterion in object detection tasks. mAP75 is the average precision calculated under a higher IoU threshold (0.75), which requires the model to predict the target position more accurately and can reflect the positioning ability of the model. mAP50-95 represents the average mAP at different IoU thresholds (from 0.5 to 0.95, with an interval of 0.05), which can more comprehensively evaluate the detection performance of the model under different precision requirements. The precision (Pre), recall (Rec), and F1-score (FS) are calculated as

$$\mathrm{Pre} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}, \mathrm{Rec} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \mathrm{FS} = \frac{\mathrm{Pre} \times \mathrm{Rec}}{\mathrm{Pre} + \mathrm{Rec}} \quad (4)$$

where TP, FP, and FN represent the true positives, false positives, and false negatives. The AP and mAP are calculated as follows:

$$\mathrm{AP} = \sum_{i=1}^{\mathrm{N}} (\mathrm{R}_i - \mathrm{R}_{i-1})\mathrm{P}_i \quad (5)$$

$$\mathrm{mAP} = \frac{1}{C}\sum_{i=1}^{C} \mathrm{AP}_i \times 100\% \quad (6)$$

where N is the number of recall levels; R_i and R_{i-1} are consecutive recall levels; P_i is the the precision at recall level; R_i and C is the number of classes.

In addition, we also considered GFLOPs to measure the computational complexity of the model and parameters to reflect the size and complexity of the model. Together, these metrics provided us with a comprehensive perspective to evaluate the precision, robustness, efficiency, and resource consumption of models in object detection tasks.

3.3. Ablation Experiments

The baseline model is the YOLOv8_n implemented by a third party. Compared with the official YOLOv8_n, this version of the model is easier to integrate multimodal feature extraction branches and lightweight improvements. All ablation experiments were conducted based on this model to incrementally verify the effectiveness of each additional module.

The design of the ablation experiment is shown in Table 1. First, the single-modal RGB input was used as the baseline to build the initial detection framework (Single-RGB). Then, by adding a depth channel, a basic model of dual-modal input (Base_DM) was constructed to verify the performance improvement of multimodal input. On this basis, several improved modules were introduced, including the interpolation module (+Interpolate), the multimodal cross-fusion module (+Concat_BiFPN), the lightweight structure C2f_repghost module (+C2f_repghost), the ShuffleNetV2 lightweight module (+ShuffleNetV2), the channel and spatial attention module CBAM (+CBAM), and the fusion module combining the SPPF and CBAM (+SPPF_CBAM). The last case was the full DCFA-YOLO model.

Table 1. Comparison results of precision, recall, F1-score, mAP50, mAP50–95, mAP75, GFLOPs, and parameters (M) of the ablation experiments. The **bold value** in each column indicate the best performance in the corresponding metric, and the underlined value the second best.

Algorithm	Precision	Recall	F1-Score	mAP50	mAP50–95	mAP75	GFLOPs	Parameters (M)
Single-RGB	0.938	0.886	0.911	0.950	0.650	0.769	8.224	3.011
Base_DM	0.947	0.889	0.917	0.950	0.639	0.745	11.573	4.370
+Interpolate	0.949	0.898	0.923	0.950	0.646	0.765	11.573	4.370
+Concat_BiFPN	**0.959**	0.891	0.924	0.957	0.659	**0.781**	11.573	4.370
+C2f_repghost	0.956	0.887	0.921	0.953	0.655	0.775	10.765	3.968
+ShuffleNetV2	0.949	0.900	0.924	0.958	0.654	0.778	**7.218**	**2.078**
+CBAM	0.951	0.906	0.928	0.958	0.648	0.762	7.225	2.122
+SPPF_CBAM (Full)	0.949	**0.914**	**0.931**	**0.965**	**0.661**	**0.775**	7.589	2.679

The baseline model achieved a basic performance of 0.938 precision, 0.886 recall, and 0.911 F1-score in single mode. With multimodal input, the precision and F1-score increased to 0.947 and 0.917, confirming the positive impact of multimodal input. Adding the interpolation module (+Interpolate) further improved the precision and F1-score to 0.949 and 0.923. The addition of the multimodal cross-fusion module (+Concat_BiFPN) further improved the precision to 0.959, F1-score to 0.924, and mAP50 to 0.957. After further replacing it with the lightweight C2f_repghost module, although GFLOPs dropped to 10.765 and the model complexity was reduced, various metrics only decreased slightly. After adding the ShuffleNetV2 module, GFLOPs further dropped to 7.218, while the precision was 0.949 and F1-score remained at 0.924. The introduction of the CBAM module increased the recall to 0.906 and F1-score to 0.928. Finally, the introduction of the SPPF_CBAM module resulted in the best performance. These results confirm that the improved model achieves a good balance between improving performance and maintaining a lightweight design.

It should be noted that after adding the CBAM module to the backbone's six feature outputs, the performance improves notably with better feature extraction. However, when we further replace the final layer's SPPF module with the SPPF_CBAM module, this small adjustment leads to the best overall performance in terms of precision, recall, F1-score, and mAP50 while still maintaining a lightweight model. This improvement is primarily due to the effective combination of spatial and channel attention from CBAM with

the spatial pyramid pooling (SPPF) module, which enhances multi-scale feature extraction, ensuring better performance without a significant increase in computational cost.

Figure 8 shows the heat map comparison for the ablation experiments. The baseline Single-RGB model exhibits focused attention on most targets but struggles with partially occluded cherry tomato clusters. The multimodal Base_DM model improves attention on these targets. Subsequent modules, such as +Interpolate, +Concat_BiFPN, and +C2f_repghost, show increasingly accurate and concentrated attention across all cherry tomato bunches. The +ShuffleNetV2 and +CBAM modules enhance the attention distribution, with +SPPF_CBAM achieving the best focus across all targets, particularly improving the detection of partially occluded clusters.

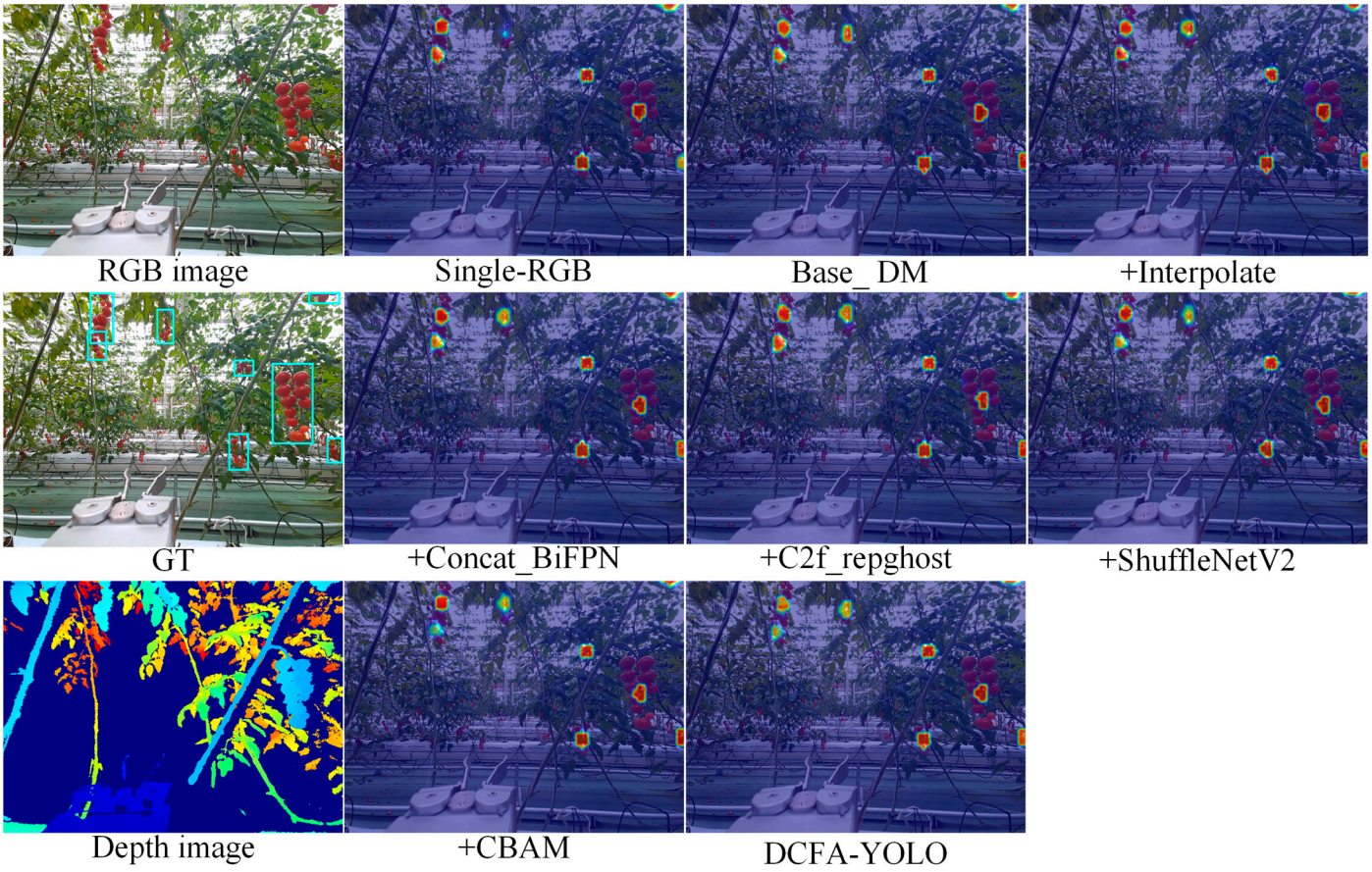

Figure 8. Comparison of heat maps for ablation experiments.

3.4. Comparison of Different Detection Algorithms

To compare the detection performance of DCFA-YOLO to other DNN models, we selected state-of-the-art YOLO models, YOLOv5_n [34], YOLOv8_n [30], YOLOv9_t [35], YOLOv10_n [36], YOLO11_n [37], EfficientDet [31], SSD [6], and CenterNet [38], for quantitative and qualitative comparison in terms of evaluation metrics and visual effects. The quantitative results are summarized in Table 2.

Table 2. Comparison results of precision, recall, F1-score, mAP50, mAP50–95, mAP75, GFLOPs, and parameters (M) of different detection models. The **bold value** in each column indicates the best performance in the metric and the underlined value the second best.

Algorithm	Precision	Recall	F1-Score	mAP50	mAP50–95	mAP75	GFLOPs	Parameters (M)
YOLOv5_n	0.865	0.838	0.851	0.905	0.565	0.626	7.100	2.503
YOLOv8_n	0.871	0.859	0.865	0.923	0.617	0.708	8.100	3.005
YOLOv9_t	0.912	0.879	0.895	0.946	0.656	0.770	7.600	**1.970**

YOLOv10_n	0.888	0.880	0.884	0.930	0.638	0.749	8.200	2.694
YOLO11_n	0.883	0.838	0.861	0.921	0.616	0.705	**6.300**	2.582
EfficientDet	0.948	0.781	0.860	0.900	0.549	0.627	7.401	3.828
SSD	0.870	0.896	0.880	0.929	0.609	0.716	58.363	11.671
CenterNet	0.952	0.797	0.870	0.919	0.572	0.654	109.710	32.665
DCFA-YOLO	**0.949**	**0.914**	**0.931**	**0.965**	**0.661**	**0.775**	7.589	2.679

DCFA-YOLO shows significant improvements in almost all key metrics. Compared with YOLOv5_n, YOLOv8_n, and YOLOv9_t, DCFA-YOLO achieves a notable increase in precision, recall, F1-score, and mAP50. Specifically, its precision improves from 0.871 (YOLOv8_n) to 0.949, and its F1-score rises from 0.865 to 0.931. Even compared with the higher-performance YOLOv9_t, DCFA-YOLO also shows obvious advantages, especially in F1-score and mAP50, with notable improvement in both. Though YOLOv10_n and YOLO11_n have lower GFLOPs and parameters, DCFA-YOLO outperforms them in key metrics like precision and mAP50. Moreover, it surpasses models such as EfficientDet and CenterNet. This further shows that DCFA-YOLO has achieved a good balance between performance and being lightweight, making it an excellent model for both accuracy and computational efficiency.

Figure 9 compares precision–recall curves across various detection models, showing that DCFA-YOLO consistently outperforms other models in terms of both precision and recall. By closely inspecting the PR curve, it is clear that the performance improvements noted in Table 2, especially in precision, recall, and F1-score, are visually reflected, demonstrating the model's strong overall detection capabilities.

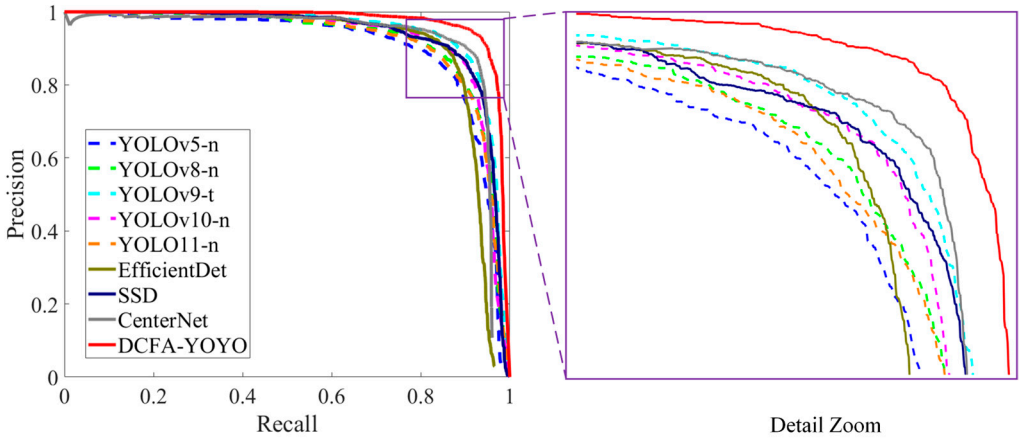

Figure 9. Precision–recall curves of different detection models.

Figures 10–12 present the visualization of detection results under normal lighting, high-light, and low-light scenarios, respectively. Under normal lighting (Figure 10), the compared models exhibit varying degrees of incomplete detection, frequently missing small, or overlapping objects and displaying suboptimal boundary precision. In contrast, DCFA-YOLO can detect cherry tomato bunches more accurately, significantly reducing missed and false detections.

Figures 11–12 illustrate detection results under challenging lighting conditions. In the high-light case (Figure 11), excessive brightness causes the image to appear washed out, leading to increased missed detections in most YOLO-based models and over-detection in CenterNet. In the low-light scenario (Figure 12), the darkness of the image exacerbates missed detections for YOLO-based models, while CenterNet again suffers from over-detection. Despite these challenges, DCFA-YOLO demonstrates superior adaptability in both scenarios, maintaining high detection accuracy with minimal false positives or missed targets.

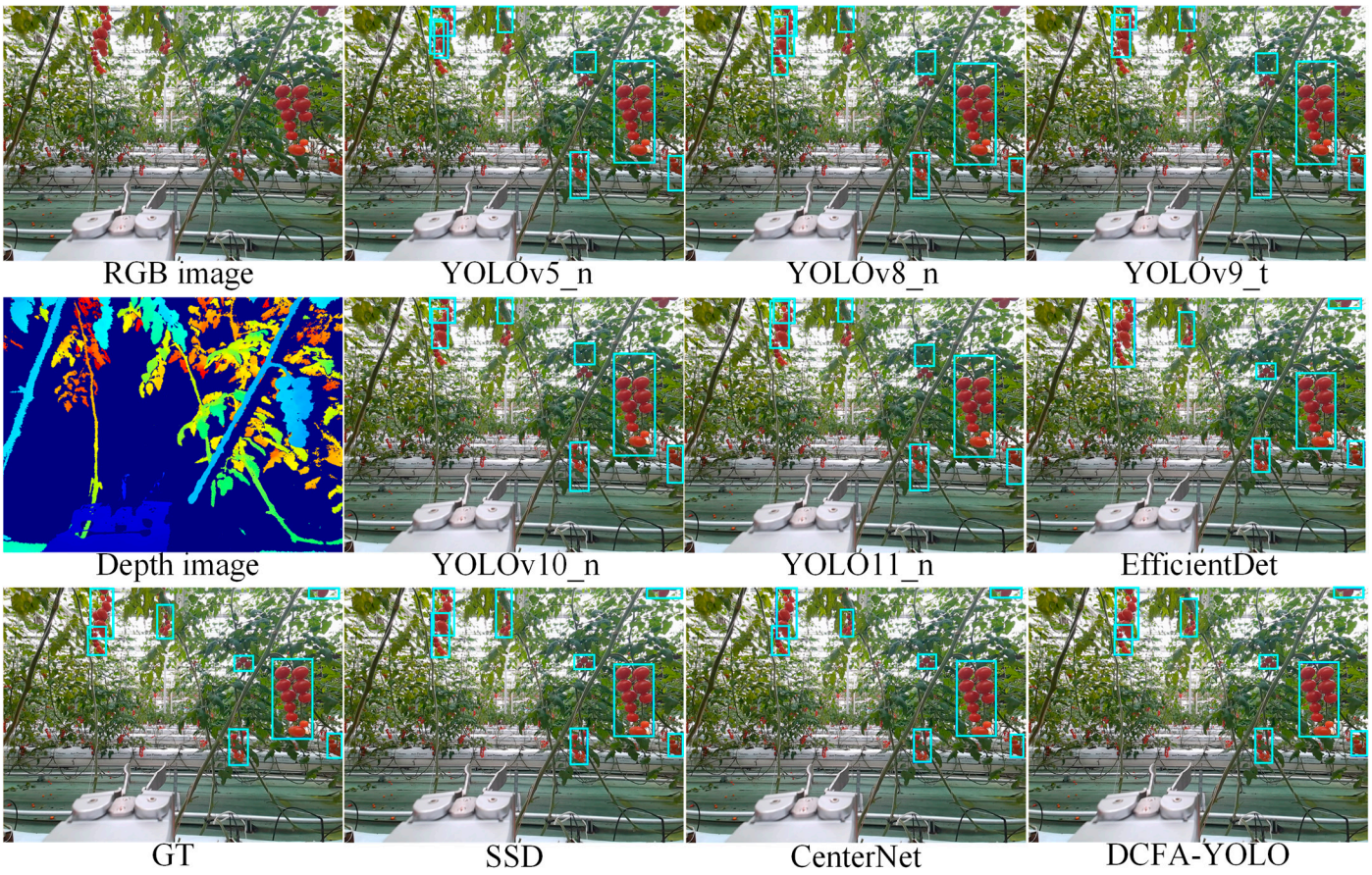

Figure 10. Visualization of detection results of different models under normal lighting conditions.

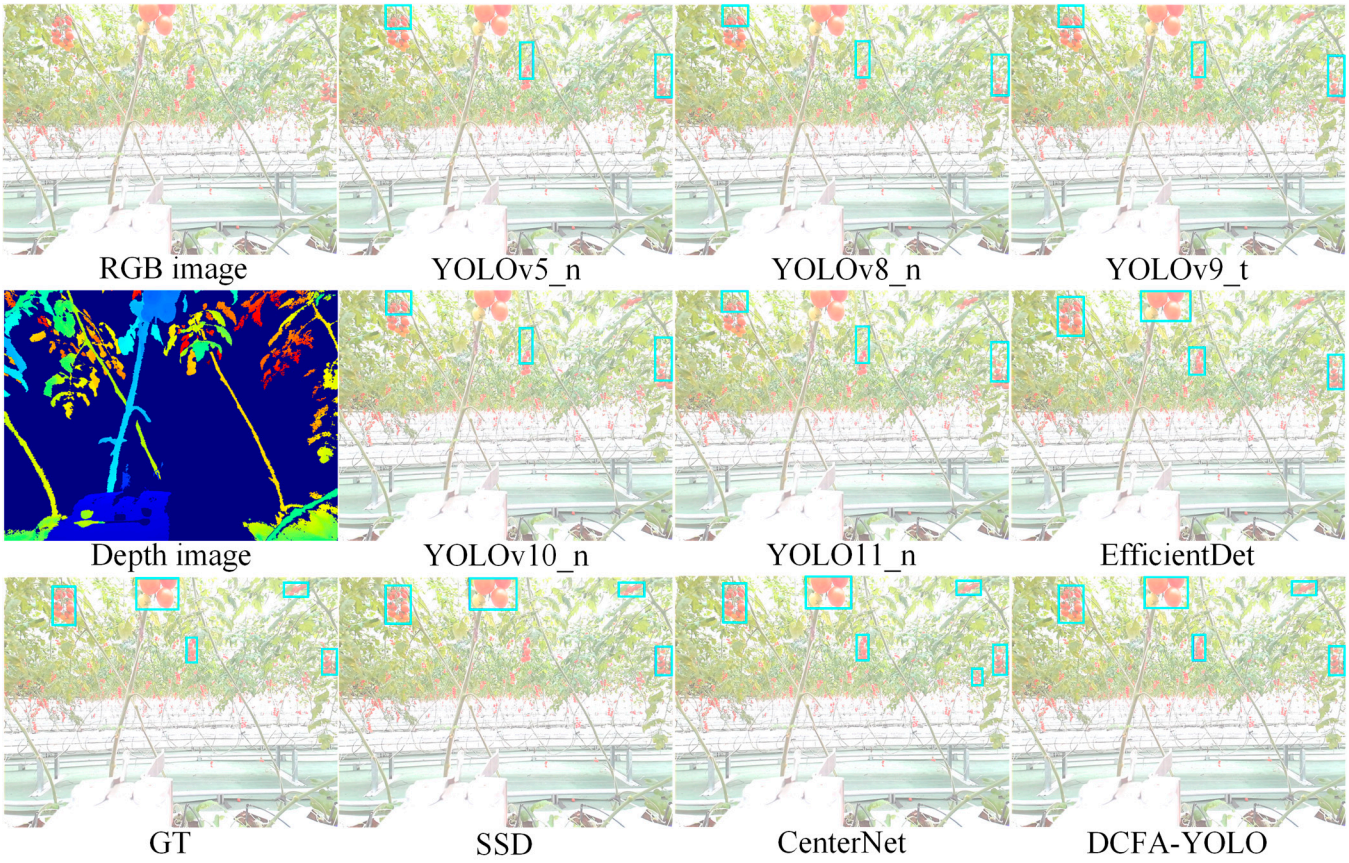

Figure 11. Visualization of detection results of different models under high-light scenarios.

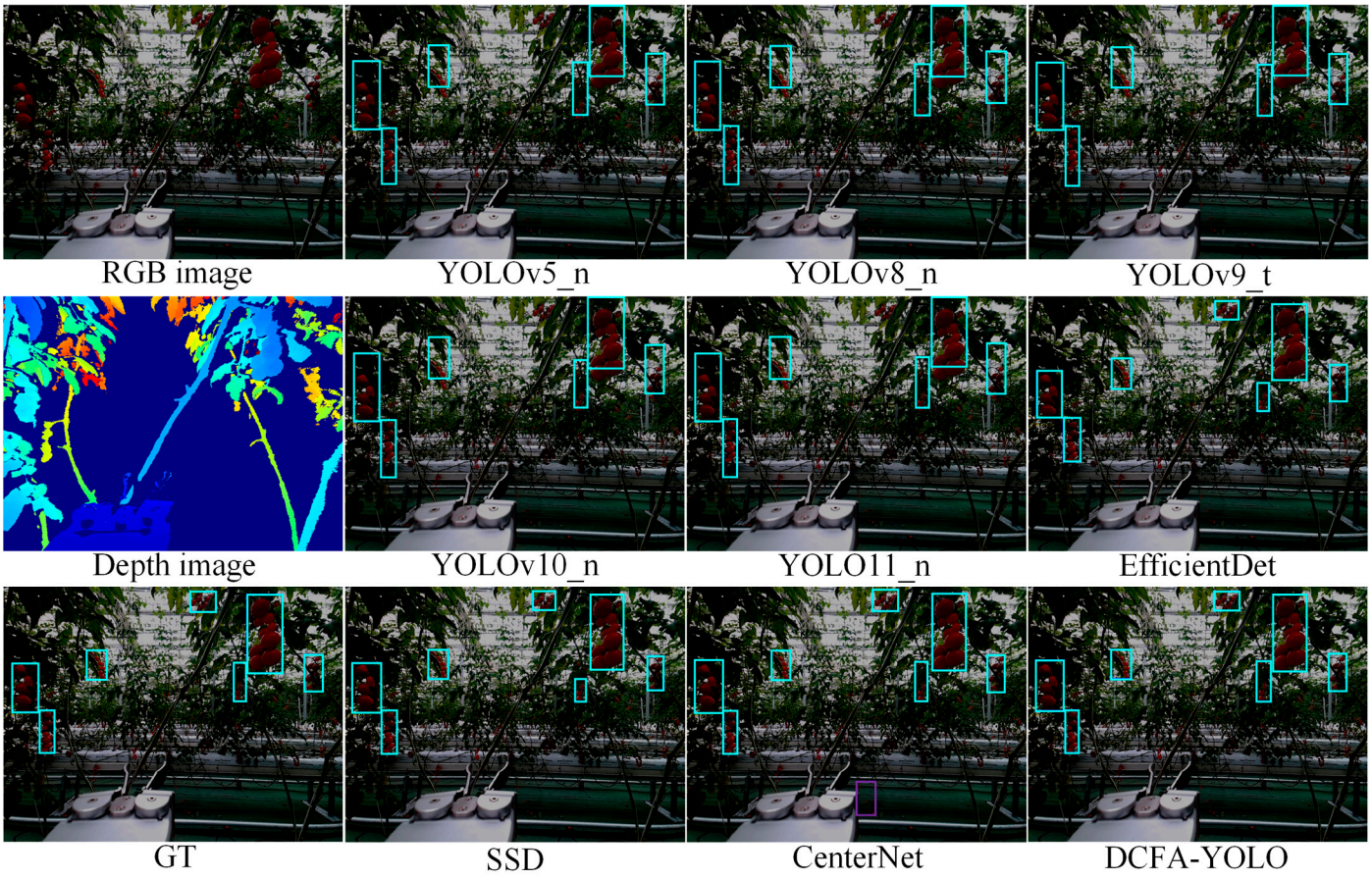

Figure 12. Visualization of detection results of different models under low-light scenarios.

4. Discussion

The proposed DCFA-YOLO method for multimodal cherry tomato bunch detection achieves the best results in six evaluation metrics (precision, recall, F1-score, mAP50, mAP75, and mAP50–95) among nine compared DNNs. It ranks the 5th among the nine models in terms of parameter size, and its GFLOPs rank the 4th lowest. For robotic fruit harvesting, the detection algorithm typically runs on edge computing devices, so the accuracy and computational cost of detection are both important, and we believe DCFA-YOLO strikes an excellent balance between the two. The computing platform in the current study is essentially a regular PC, and DCFA-YOLO runs cherry tomato bunch detection at a frame rate of 52.93 fps, which is sufficient for practical real-time robotic harvesting. By achieving high accuracy with lower computational costs, DCFA-YOLO reduces the burden on hardware, which could lead to lower energy consumption and operational costs in real-world applications. It can be integrated into agricultural systems like robotic harvesters or drones, enhancing their perception for precise and efficient fruit picking and measurement.

In a previous cherry tomato bunch detection study, precision, recall, and an F1-score of 98.9, 92.1, and 95.4 have been reported [39], higher than the corresponding values of 0.949, 0.914, and 0.931 in the current study. However, the two studies were conducted with two very different datasets that were collected by different cameras (Intel RealSense structured light stereo vs. Microsoft Kinect TOF) at different plantations. The code and dataset are not available for apple-to-apple comparisons. To partly address this, we make the code publicly available on Github (see Data Availability Statement). Unfortunately, the dataset cannot be made fully public due to a non-disclosure agreement.

In this study, we only tested cherry tomato bunch detection. For clustered fruits like cherry tomatoes, whether harvesting by single fruits or by bunches depends on the production needs in a commercial setting, and both have been performed in previous studies [22,39,40]. We believe DCFA-YOLO can be applicable to single fruit detection as well, which will be investigated in future work. However, it is important to note that the current model may face challenges in more complex scenarios, such as severe occlusions or highly dense clusters, where the accuracy of detection could be compromised. Future improvements could focus on enhancing the model's robustness to these conditions, potentially through advanced attention mechanisms or multi-scale feature fusion techniques.

We would also like to explore applying DCFA-YOLO to other fruits; for example, it may be partially transferable to detecting other fruits in clusters/bunches, such as grape, longan, or lychee. It should be noted that for these fruits, even at maturity, the color features are less salient compared to cherry tomatoes. As such, we expect that the impact of adding depth information for these fruits is likely to have a more significant impact than for cherry tomatoes.

Moreover, when integrated with harvesting robots, DCFA-YOLO can contribute to agricultural sustainability by improving harvesting efficiency and reducing fruit waste. Its high detection accuracy minimizes missed or damaged fruits, while its real-time capabilities enable faster and more precise harvesting, ultimately supporting more sustainable and resource-efficient agricultural practices.

5. Conclusions

This study introduces DCFA-YOLO, a lightweight fruit detection model with multimodal cross fusion. By integrating the dual-channel cross-fusion mechanism, a dynamically weighted feature-fusion module, and optimized lightweight design alongside attention mechanisms, the model demonstrates notable improvements in efficiency and robust-

ness for multimodal detection tasks. Specifically, the model leverages complementary information from color and depth images through targeted enhancements to the backbone network and feature-fusion stages. These improvements effectively reduce computational complexity and model parameters while maintaining high detection accuracy, addressing the challenges of small fruit detection in complex scenarios. The experimental results on cherry tomato bunch detection further validate the model's superiority, showcasing its potential for practical application in precision agriculture as a core component of robotic cherry tomato harvesting.

Author Contributions: Conceptualization, Z.Z. and Y.T.; methodology, S.C., P.L. and Y.T.; software, S.C.; validation, S.C. and M.W.; formal analysis, S.C. and Y.T.; investigation, S.C. and M.W.; resources, Y.T.; data curation, Z.Z. and P.L.; writing—original draft preparation, S.C.; writing—review and editing, M.W. and Y.T.; visualization, S.C.; supervision, Y.T.; project administration, Y.T.; funding acquisition, M.W. and Y.T. All authors have read and agreed to the published version of the manuscript.

Funding: This study was funded by the Shenzhen Talent Startup Funds (No. 827-000954) and Shenzhen University (Nos. 868/03010315 and 86902/00248).

Institutional Review Board Statement: Not applicable.

Data Availability Statement: The program code is at https://github.com/heitieya/DCFA-YOLO (accessed on 24 January 2025). The dataset cannot be made fully public due to a non-disclosure agreement, but limited samples can be provided at reasonable request.

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

References

1. Wang, Z.; Xun, Y.; Wang, Y.; Yang, Q. Review of smart robots for fruit and vegetable picking in agriculture. *Int. J. Agric. Biol. Eng*. **2022**, *15*, 33–54.
2. Agarwal, S.; Rao, A.V. Tomato lycopene and its role in human health and chronic diseases. *CMAJ* **2000**, *163*, 739–744.
3. Hua, X.; Li, H.; Zeng, J.; Han, C.; Chen, T.; Tang, L.; Luo, Y. A review of target recognition technology for fruit picking robots: From digital image processing to deep learning. *Appl. Sci.* **2023**, *13*, 4160.
4. Jiang, P.; Ergu, D.; Liu, F.; Cai, Y.; Ma, B. A Review of Yolo algorithm developments. *Procedia Comput. Sci.* **2022**, *199*, 1066–1073.
5. Redmon, J. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016.
6. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. SSD: Single shot multibox detector. In Proceedings of the Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; pp. 21–37.
7. Lyu, S.; Li, R.; Zhao, Y.; Li, Z.; Fan, R.; Liu, S. Green citrus detection and counting in orchards based on YOLOv5-CS and AI edge system. *Sensors* **2022**, *22*, 576.
8. Wang, L.; Zhao, Y.; Xiong, Z.; Wang, S.; Li, Y.; Lan, Y. Fast and precise detection of litchi fruits for yield estimation based on the improved YOLOv5 model. *Front. Plant Sci.* **2022**, *13*, 965425.
9. Gai, R.; Chen, N.; Yuan, H. A detection algorithm for cherry fruits based on the improved YOLO-v4 model. *Neural Comput. Appl.* **2023**, *35*, 13895–13906.
10. Zhao, R.; Zhu, Y.; Li, Y. An end-to-end lightweight model for grape and picking point simultaneous detection. *Biosyst. Eng.* **2022**, *223*, 174–188.
11. Yuan, T.; Lv, L.; Zhang, F.; Fu, J.; Gao, J.; Zhang, J.; Li, W.; Zhang, C.; Zhang, W. Robust cherry tomatoes detection algorithm in greenhouse scene based on SSD. *Agriculture* **2020**, *10*, 160.
12. Fuentes-Peñailillo, F.; Carrasco Silva, G.; Pérez Guzmán, R.; Burgos, I.; Ewertz, F. Automating seedling counts in horticulture using computer vision and AI. *Horticulturae* **2023**, *9*, 1134.
13. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.

14. Girshick, R. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
15. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 1137–1149.
16. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1904–1916.
17. Hu, C.; Liu, X.; Pan, Z.; Li, P. Automatic detection of single ripe tomato on plant combining faster R-CNN and intuitionistic fuzzy set. *IEEE Access* **2019**, *7*, 154683–154696.
18. Song, Z.; Fu, L.; Wu, J.; Liu, Z.; Li, R.; Cui, Y. Kiwifruit detection in field images using Faster R-CNN with VGG16. *IFAC-PapersOnline* **2019**, *52*, 76–81.
19. Gao, F.; Fu, L.; Zhang, X.; Majeed, Y.; Li, R.; Karkee, M.; Zhang, Q. Multi-class fruit-on-plant detection for apple in SNAP system using Faster R-CNN. *Comput. Electron. Agric.* **2020**, *176*, 105634.
20. Tu, S.; Xue, Y.; Zheng, C.; Qi, Y.; Wan, H.; Mao, L. Detection of passion fruits and maturity classification using Red-Green-Blue Depth images. *Biosyst. Eng.* **2018**, *175*, 156–167.
21. Cui, B.; Zeng, Z.; Tian, Y. A Yolov7 cherry tomato identification method that integrates depth information. In Proceedings of the Third International Conference on Optics and Image Processing (ICOIP 2023), Hangzhou, China, 14–16 April 2023; pp. 312–320.
22. Cai, Y.; Cui, B.; Deng, H.; Zeng, Z.; Wang, Q.; Lu, D.; Cui, Y.; Tian, Y. Cherry Tomato Detection for Harvesting Using Multimodal Perception and an Improved YOLOv7-Tiny Neural Network. *Agronomy* **2024**, *14*, 2320.
23. Rong, J.; Zhou, H.; Zhang, F.; Yuan, T.; Wang, P. Tomato cluster detection and counting using improved YOLOv5 based on RGB-D fusion. *Comput. Electron. Agric.* **2023**, *207*, 107741.
24. Kaukab, S.; Ghodki, B.M.; Ray, H.; Kalnar, Y.B.; Narsaiah, K.; Brar, J.S. Improving real-time apple fruit detection: Multi-modal data and depth fusion with non-targeted background removal. *Ecol. Inform.* **2024**, *82*, 102691.
25. Liu, Z.; Wu, J.; Fu, L.; Majeed, Y.; Feng, Y.; Li, R.; Cui, Y. Improved kiwifruit detection using pre-trained VGG16 with RGB and NIR information fusion. *IEEE Access* **2019**, *8*, 2327–2336.
26. Sa, I.; Ge, Z.; Dayoub, F.; Upcroft, B.; Perez, T.; McCool, C. Deepfruits: A fruit detection system using deep neural networks. *Sensors* **2016**, *16*, 1222.
27. Wei, Z.; Guodong, Y.; Haoji, L.; Keke, G.; Wenhan, H.; Yuan, W.; Hongwei, X. Low-observable Target Detection Method for Autonomous Vehicles Based on Multi-modal Feature Fusion. *China Mech. Eng.* **2021**, *32*, 1114.
28. Woo, S.; Park, J.; Lee, J.-Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
29. Li, H.; Wu, X.-J. DenseFuse: A fusion approach to infrared and visible images. *IEEE Trans. Image Process.* **2018**, *28*, 2614–2623.
30. Sohan, M.; Sai Ram, T.; Reddy, R.; Venkata, C. A review on yolov8 and its advancements. In Proceedings of the International Conference on Data Intelligence and Cognitive Informatics, Tirunelveli, India, 27–28 June 2024; pp. 529–545.
31. Tan, M.; Pang, R.; Le, Q.V. Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10781–10790.
32. Li, X.; Wang, W.; Wu, L.; Chen, S.; Hu, X.; Li, J.; Tang, J.; Yang, J. Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 21002–21012.
33. Zheng, L.; Shen, L.; Tian, L.; Wang, S.; Wang, J.; Tian, Q. Scalable person re-identification: A benchmark. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1116–1124.
34. Jocher, G. YOLOv5 by Ultralytics, Version 7.0; 2020. This is a GitHub link: https://github.com/ultralytics/yolov5 (accessed on 24 January 2025)
35. Wang, C.-Y.; Yeh, I.-H.; Mark Liao, H.-Y. Yolov9: Learning what you want to learn using programmable gradient information. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 22–23 January, 2025; pp. 1–21.
36. Wang, A.; Chen, H.; Liu, L.; Chen, K.; Lin, Z.; Han, J.; Ding, G. Yolov10: Real-Time End-to-End Object Detection. *arXiv* **2024**, arXiv:14458.
37. Khanam, R.; Hussain, M. Yolov11: An Overview of the Key Architectural Enhancements. *arXiv* **2024**, arXiv:17725.
38. Duan, K.; Bai, S.; Xie, L.; Qi, H.; Huang, Q.; Tian, Q. Centernet: Keypoint triplets for object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 6569–6578.

39. Li, Y.; Feng, Q.; Liu, C.; Xiong, Z.; Sun, Y.; Xie, F.; Li, T.; Zhao, C. MTA-YOLACT: Multitask-aware network on fruit bunch identification for cherry tomato robotic harvesting. *Eur. J. Agron.* **2023**, *146*, 126812.
40. Chen, W.; Liu, M.; Zhao, C.; Li, X.; Wang, Y. MTD-YOLO: Multi-task deep convolutional neural network for cherry tomato fruit bunch maturity detection. *Comput. Electron. Agric.* **2024**, *216*, 108533.