agriculture

MDPI

Article

Research on Strawberry Visual Recognition and 3D Localization Based on Lightweight RAFS-YOLO and RGB-D Camera

Kaixuan Li, Xinyuan Wei, Qiang Wang and Wuping Zhang *

College of Software, Shanxi Agricultural University, Jinzhong 030801, China; 20232092@stu.sxau.edu.cn (K.L.); 20232095@stu.sxau.edu.cn (X.W.); 20233787@stu.sxau.edu.cn (Q.W.)
* Correspondence: zhangwuping@sxau.edu.cn

Abstract

Improving the accuracy and real-time performance of strawberry recognition and localization algorithms remains a major challenge in intelligent harvesting. To address this, this study presents an integrated approach for strawberry maturity detection and 3D localization that combines a lightweight deep learning model with an RGB-D camera. Built upon the YOLOv11 framework, an enhanced RAFS-YOLO model is developed, incorporating three core modules to strengthen multi-scale feature fusion and spatial modeling capabilities. Specifically, the CRA module enhances spatial relationship perception through cross-layer attention, the HSFPN module performs hierarchical semantic filtering to suppress redundant features, and the DySample module dynamically optimizes the upsampling process to improve computational efficiency. By integrating the trained model with RGB-D depth data, the method achieves precise 3D localization of strawberries through coordinate mapping based on detection box centers. Experimental results indicate that RAFS-YOLO surpasses YOLOv11n, improving precision, recall, and mAP@50 by 4.2%, 3.8%, and 2.0%, respectively, while reducing parameters by 36.8% and computational cost by 23.8%. The 3D localization attains millimeter-level precision, with average RMSE values ranging from 0.21 to 0.31 cm across all axes. Overall, the proposed approach achieves a balance between detection accuracy, model efficiency, and localization precision, providing a reliable perception framework for intelligent strawberry-picking robots.

Keywords: strawberry; maturity detection; 3D localization; RAFS-YOLO; deep learning

Academic Editor: Maciej Zaborowicz

Received: 22 September 2025
Revised: 17 October 2025
Accepted: 20 October 2025
Published: 24 October 2025

Citation: Li, K.; Wei, X.; Wang, Q.; Zhang, W. Research on Strawberry Visual Recognition and 3D Localization Based on Lightweight RAFS-YOLO and RGB-D Camera. *Agriculture* **2025**, *15*, 2212. https://doi.org/10.3390/agriculture15212212

1. Introduction

Against the backdrop of the continuous advancement of smart agriculture, intelligent picking robots, which aim to improve efficiency and alleviate labor shortages, have become a research hotspot in modern agriculture [1]. Among them, as a typical representative of soft and perishable fruits, strawberries have always been the core object of intelligent picking research because they have delicate fruits that are vulnerable to damage and require precise judgment of maturity [2]. Visual recognition, as one of the core links of picking robots, has an important impact on picking efficiency and success rate [3–6]. However, in practical applications, visual recognition faces multiple challenges: first, the plant leaves are lush, and the fruits are often partially or severely occluded, leading to high detection difficulty and a high missed detection rate [7]; second, there are a wide variety of strawberry varieties, and the fruits have significant differences in size, shape, and color, making it difficult for a single model to have good generalization

ability [8]; third, the color of strawberries changes slightly during ripening, and methods relying on color thresholds are difficult to accurately distinguish maturity levels, thereby affecting the judgment of picking timing [9]; fourth, the lighting conditions change drastically, and shadows or strong reflections generated under different times, angles, and intensities will weaken the stability and robustness of target recognition [10]. The above problems make it difficult for traditional image processing methods or shallow machine learning technologies to meet the needs of complex agricultural environments, thus seriously restricting the application and popularization of strawberry picking robots in practical scenarios.

In recent years, the rapid development of deep learning has provided new opportunities for agricultural visual inspection. Deep learning methods represented by convolutional neural networks have been widely applied to tasks such as fruit recognition [11], disease detection, and yield estimation, and have achieved remarkable results in crops like apples [12], tomatoes [13], citrus [14], and pear [15]. According to differences in model structures, deep learning object detection algorithms can generally be divided into two categories. One is the two-stage method represented by FasterR-CNN and CascadeR-CNN. This type of method first generates candidate regions, and then performs classification and bounding box regression on these regions [16]. For example, Xiong et al. [17] proposed a visual detection method for green citrus based on FasterR-CNN, which can accurately identify green citrus under different lighting conditions; Wang et al. [18] proposed a detection method for tomato young fruits in similar-color backgrounds based on the improved FasterR-CNN with an attention mechanism, which can efficiently identify tomato young fruits in complex environments. Kong et al. [19] proposed the Faster-RFormer model based on Faster-RCNN, which enables accurate apple detection in complex orchard environments. The other type is the one-stage object detection method represented by models such as SSD [20] and the YOLO series [21]. This type of method does not need to generate candidate regions and directly predicts the category and bounding box of the object [22]. For example, Liang et al. [23] proposed a detection framework based on the SSD network for real-time detection of mango fruits; Gai et al. [24] proposed a YOLOv4 deep learning algorithm for cherry fruit detection, which is suitable for small-sized cherries; Sun et al. [25] proposed the G-YOLO-NK model based on YOLOv5s, which enables real-time detection of passion fruit in complex orchard scenarios; Wu et al. [26] proposed the DNE-YOLO model based on YOLOv8 to achieve accurate, fast, and effective apple detection with limited computing resources and in various complex natural environments; Wang et al. [27] proposed the Jujube-YOLO model based on YOLOv11n, which can complete the fresh jujube fruit recognition task in unstructured environments; Nan et al. [28] proposed the WGB-YOLO network model, which exhibits excellent performance in detecting dragon fruit in densely planted orchards. Wang et al. [29] proposed a precise, lightweight small-model method for detecting small apple fruits based on the channel-pruned YOLOv5s deep learning algorithm, enabling fast and accurate detection of small apple fruits. Compared with two-stage methods, one-stage methods, featuring faster inference efficiency and lower computational overhead, have greater deployment advantages on resource-constrained embedded platforms, which aligns with the requirements for real-time performance and edge computing in smart agriculture. Therefore, in terms of strawberry detection, extensive research has also been conducted on one-stage algorithms. For example, Shen et al. [30] proposed the RTF-YOLO network model based on YOLOv5s, which is used for strawberry detection under fluctuating illumination and fruit occlusion scenarios; Du et al. [31] proposed a DSW-YOLO network model based on the YOLOv7 network model, which can quickly and accurately detect mature strawberry fruits with

different occlusion levels in complex field environments; Wang et al. [32] proposed the DSE-YOLO network model, which can accurately detect strawberries at each developmental stage in natural scenes; Wang et al. [33] realized strawberry detection and maturity classification based on the YOLOv8 model and image processing methods; Liu et al. [34] proposed the YOLOv11-HRS network model, aiming to address challenges such as complex environmental interference, multi-scale target variations, and small target recognition in strawberry ripeness detection.

Overall, although existing research has made significant progress in strawberry fruit object detection, there are still shortcomings: on the one hand, much work mainly focuses on the detection of mature fruits, with limited attention paid to the identification of fruits at different ripening stages; on the other hand, although some studies have attempted to conduct maturity classification, they often neglect detection efficiency, making it difficult to meet the real-time requirements of practical applications. In addition, there are also methods that combine object detection with traditional image processing to complete maturity classification, but this leads to a complex overall process and poor portability. Therefore, the organic combination of object detection and maturity recognition, while ensuring efficiency and improving detection accuracy, still holds significant research value [35]. To achieve this goal, introducing more advanced object detection models has become a necessary choice. As a new-generation object detection model, YOLOv11 has demonstrated excellent performance on multiple public datasets. It possesses both high detection accuracy and inference efficiency, exhibiting strong versatility [36]. However, when directly applied to strawberry fruit detection tasks, it still suffers from insufficient adaptability: first, the overall scale of the model is relatively large, leading to high inference overhead on resource-constrained agricultural machinery platforms, which makes it difficult to ensure real-time performance; second, in complex agricultural scenarios, affected by factors such as fruit occlusion, differences in maturity, and indistinct features of small targets, there is still room for improvement in terms of detection accuracy and robustness. This indicates that targeted improvements to YOLOv11 to better meet the practical needs of strawberry detection tasks are of great significance.

In addition to visual recognition, fruit localization is also a key link in intelligent picking technology [37]. In practice, picking robots need to identify fruits. This includes detecting their presence and judging their maturity. Additionally, they must obtain precise 3D positions of the fruits. All these steps help the robotic arm with path planning and grasping. Current common fruit localization methods mainly rely on depth perception technologies, including depth acquisition methods based on binocular stereo vision [38], structured light [39], and Time of Flight (ToF) [40], as well as lidar based on active ranging [41]. Among them, RGB-D cameras, as typical representatives, are usually implemented based on one of the aforementioned principles. They can output color images and depth information simultaneously while ensuring precise alignment of the two, thus being widely used in greenhouses and short-range scenarios and effectively improving 3D perception accuracy [42]. However, if we only rely on depth maps or point clouds for fruit detection and localization, it is often difficult to obtain stable results in complex environments, and they are easily disturbed by leaf occlusion and depth noise [43]. In contrast, a more reliable strategy is to combine object detection models with RGB-D cameras: first, achieve accurate fruit detection through RGB images, then calculate the corresponding spatial coordinates by integrating depth information. This approach not only ensures detection accuracy and localization reliability but also offers advantages in terms of real-time performance.

In conclusion, this paper proposes a method for strawberry maturity identification and 3D positioning that combines a target detection model with an RGB-D camera. Based

on YOLOv11, a lightweight RAFS-YOLO model is designed. By drawing on the ideas of existing structures, three functional modules—CRA, HSFPN, and DySample—are introduced and improved to achieve high-precision and efficient detection of strawberry fruits in complex environments. Among them, the CRA module refers to the structural idea of Retblock and incorporates multi-branch residuals and a lightweight attention mechanism to enhance spatial modeling capabilities while reducing computational overhead; the HSFPN module draws on the FPN and BiFPN, and effectively improves detection accuracy in complex scenarios and small target recognition performance by introducing inter-layer selective connection and feature selection strategies; the DySample module is lightweight adjusted based on the dynamic sampling mechanism and adopts an adaptive feature sampling strategy, which reduces the amount of computation while improving upsampling accuracy and feature reconstruction quality. Relying on the optimized design of the above modules and combining the depth information provided by the RGB-D camera, this paper realizes a joint integration method of detection and positioning, which can achieve high-precision identification and real-time 3D positioning of strawberry fruits. This paper aims to construct a visual perception module with high-precision detection and real-time spatial positioning capabilities for intelligent picking systems, providing technical support for fruit identification, maturity judgment, and grasping decision-making.

1. A lightweight RAFS-YOLO model is constructed based on YOLOv11, which effectively reduces the number of parameters and computational overhead while ensuring detection accuracy;
2. Three modules—CRA, HSFPN, and DySample—are introduced and improved in the network structure. These modules optimize spatial location modeling capabilities, multi-scale feature fusion, and upsampling efficiency, thereby enhancing the model's detection robustness and inference efficiency;
3. A visual positioning system that combines a detection model with the depth information of an RGB-D camera is constructed, which enables real-time fruit detection and high-precision 3D coordinate output, providing a reliable visual foundation for the automated decision-making of intelligent picking systems.

2. Materials and Methods

This section mainly introduces the collection and annotation process of the dataset, as well as the overall framework and key technical links of the proposed visual recognition and localization method for strawberry ripeness. The overall process consists of four parts:

1. Dataset collection and annotation, which is used to construct the basic data resources required for model training and performance evaluation;
2. The improved object detection model RAFS-YOLO, which is used to realize the automatic recognition of strawberry fruits and ripeness discrimination in RGB images;
3. Camera calibration, which involves accurate calibration of the RGB camera part in the RGB-D camera to ensure the accuracy of 3D spatial localization;
4. Spatial coordinate calculation, which maps image pixel coordinates to 3D spatial coordinates based on the depth information provided by the RGB-D camera, thereby achieving the spatial localization of strawberries.

The overall method flow is shown in Figure 1.

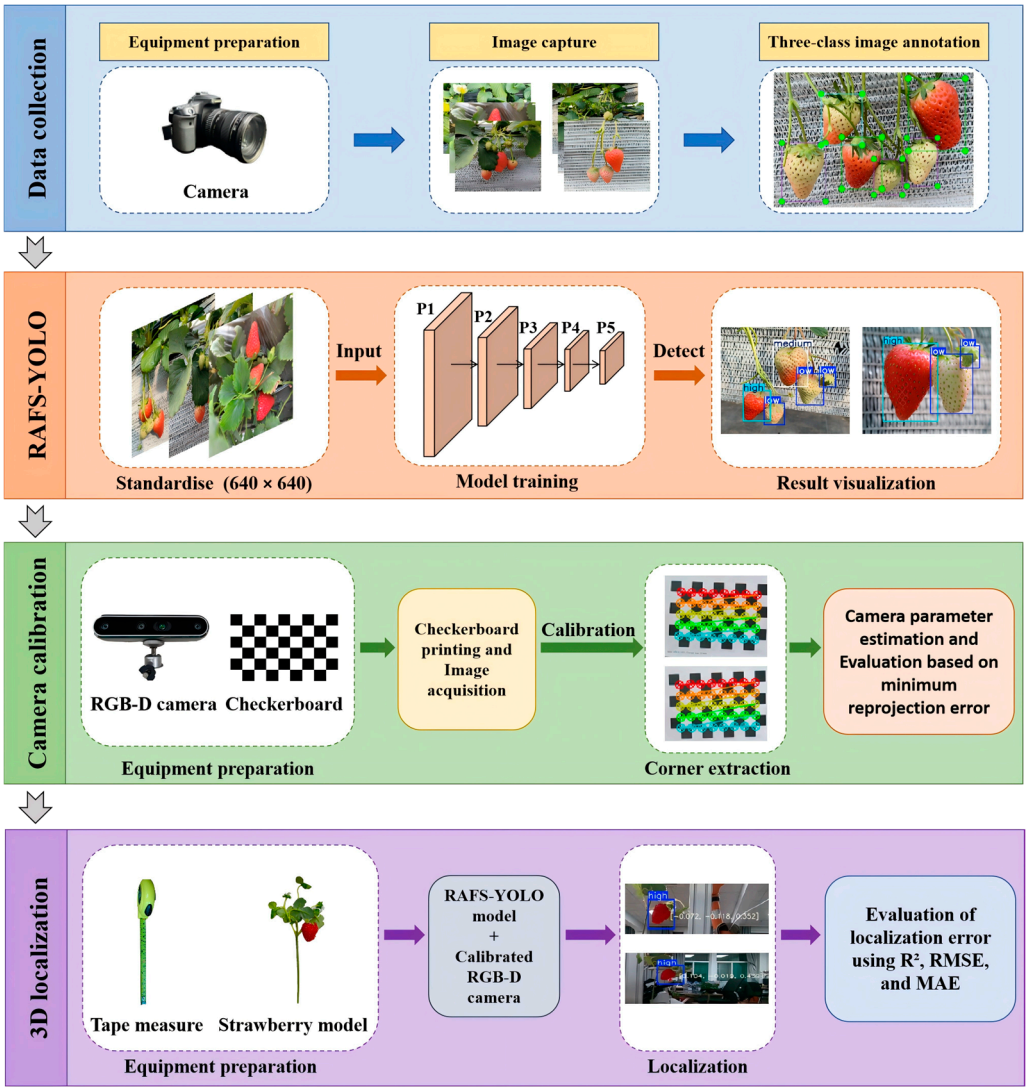

Figure 1. Overall flow chart (drawn by the authors).

2.1. Dataset and Annotation

To cover complex scenarios in strawberry cultivation, we independently collected and annotated a strawberry detection dataset. The dataset was sourced from the Strawberry Greenhouse of the Fruit Tree Institute, Shanxi Agricultural University, containing images of strawberry plants of different varieties. The main strawberry varieties grown in the greenhouse include three types: Benihoppe, Akihime, and Sweet Charlie. Approximately 70, 60, and 50 plant samples were collected, respectively, totaling about 180 strawberry plants. This distribution balances the proportion of major cultivated varieties, helping to improve the model's generalization ability across different varieties. During collection, a Canon DSLR camera was used to capture RGB images. To ensure data diversity and reproducibility, a unified shooting standard was followed for each strawberry plant: the horizontal distance between the camera and fruits was maintained at approximately 40–60 cm; the shooting height was roughly level with the fruit center (within ±5 cm); multi-angle images were taken for each plant from the front, left, right, and top—about 5–7 images per plant—to simulate the mobile observation perspective of a robot during picking. The camera was kept horizontal or slightly tilted downward (approximately 10–15°), and shooting was done under natural light. Exposure was adjusted appropriately to obtain images with balanced illumination.

The final dataset contains 1158 RGB images, each with a resolution of 1280 × 720, including a total of approximately 8627 strawberry targets—with 1 to 10 fruits per image. The number of all fruit targets was automatically counted via annotation files to ensure the accuracy and consistency of data statistics. Based on the coverage of red areas on the fruit surface, samples were divided into three maturity levels: low, medium, and high, with quantities of 4776, 2452, and 1399, respectively, accounting for approximately 55.4%, 28.4%, and 16.2%. Scene complexity varies: some backgrounds have neat elevated planting structures, while others contain interference such as leaves and drip irrigation tubes; lighting conditions range from soft morning light to strong midday light; and the degree of fruit occlusion also differs significantly, as shown in Figure 2. These diverse collection conditions effectively enhance the model's robustness and generalization ability under different lighting and complex backgrounds.

Figure 2. Examples of the self-built strawberry dataset (drawn by the authors).

It should be noted that DSLR camera shooting is only used in the offline dataset construction phase, aiming to obtain high-resolution, low-distortion, and accurately annotated sample data to improve the quality of feature learning during the model training phase. This phase mainly serves algorithm development and verification and does not involve the real-time operation of the system. In the actual system deployment phase, this study uses an RGB-D camera (Intel RealSense D457) for real-time image acquisition and detection, realizing an end-to-end online processing workflow to avoid the delay problem of the "shooting → transmission → offline processing" process. Due to differences in resolution, field of view, and color reproduction between the two cameras, an online data augmentation strategy is adopted in the model training phase to improve the model's adaptability to different imaging conditions, thereby effectively mitigating the impact of potential domain shifts.

During the data annotation process, the LabelImg tool (Version 1.8.6) was used for manual image annotation, and rectangular boxes closely fitting the edges of fruits were adopted as target position labels to minimize background interference. When fruits were partially occluded, the rectangular boxes were drawn to cover their approximate contours, ensuring the integrity of the target area and thereby improving positioning accuracy. Maturity categories were divided based on the coverage rate of the red area on the fruit surface: a red ratio of less than 30% was defined as "low", 30–80% as "medium", and more than 80% as "high", realizing three-category annotation. Maturity annotation was completed by 2 researchers with an agricultural engineering background through manual visual judgment. The two annotators independently annotated all images and cross-checked the results; in case of discrepancies, they reached a consensus through discussion to ensure the consistency and reliability of the annotation standards. This manual classification method can more accurately reflect the actual maturity status of fruits under complex lighting and partial occlusion conditions, avoiding the misjudgment problem that the automatic color threshold method may encounter in greenhouse imaging environments. The annotation results recorded information such as the center coordinates, width, and height of strawberry targets. Since YOLO model training requires annotation files in txt format, the original xml files were finally batch-converted to txt format using a Python script, thus generating a dataset that can be directly used for model training.

White strawberry samples are not included in the dataset of this study, mainly due to their limited planting scale and relatively less prominent ripening characteristics [44]. In contrast, red strawberries have greater advantages in terms of planting area, market demand, and industrial value. Meanwhile, their ripening process is accompanied by obvious color changes, which provides a more reliable basis for ripeness identification based on visual features. Therefore, this study focuses on the ripeness identification and localization of red strawberries. As can be seen from Figure 3a, the samples of strawberries with low ripeness and medium ripeness account for a relatively high proportion in the dataset. This is mainly because the differences in their appearance features such as color and texture are relatively subtle, which are easy to cause model confusion. Therefore, the number of these two types of samples was deliberately increased during the data collection process to enhance the learning ability of the model. In contrast, strawberries with high ripeness have more obvious color features and discriminative markers, so the required number of samples is relatively small. Although this collection strategy leads to a certain imbalance in category distribution, it can effectively improve the recognition robustness and generalization ability of the model when distinguishing easily confused categories. As can be seen from Figure 3b, the strawberry samples of different ripeness categories are relatively evenly distributed spatially in the images, covering the width and height ranges of the images overall. This indicates that the dataset avoided the bias of targets being overly concentrated in specific areas during the collection process, and can effectively support the model in learning the detection and localization capabilities under different spatial positions. Meanwhile, despite the differences in the number of samples, the three categories of strawberries all have a certain degree of spatial diversity, which helps to improve the generalization performance and robustness of the model in complex scenarios. In this study, the dataset was split into an 8:1:1 ratio, where the training set contains 926 images, and the validation set and test set each contain 116 images. During the splitting process, the independence of samples was strictly maintained across all subsets to ensure no data leakage occurred during the training and evaluation phases, thereby improving the credibility of the experimental results and the reliability of the model's generalization performance.

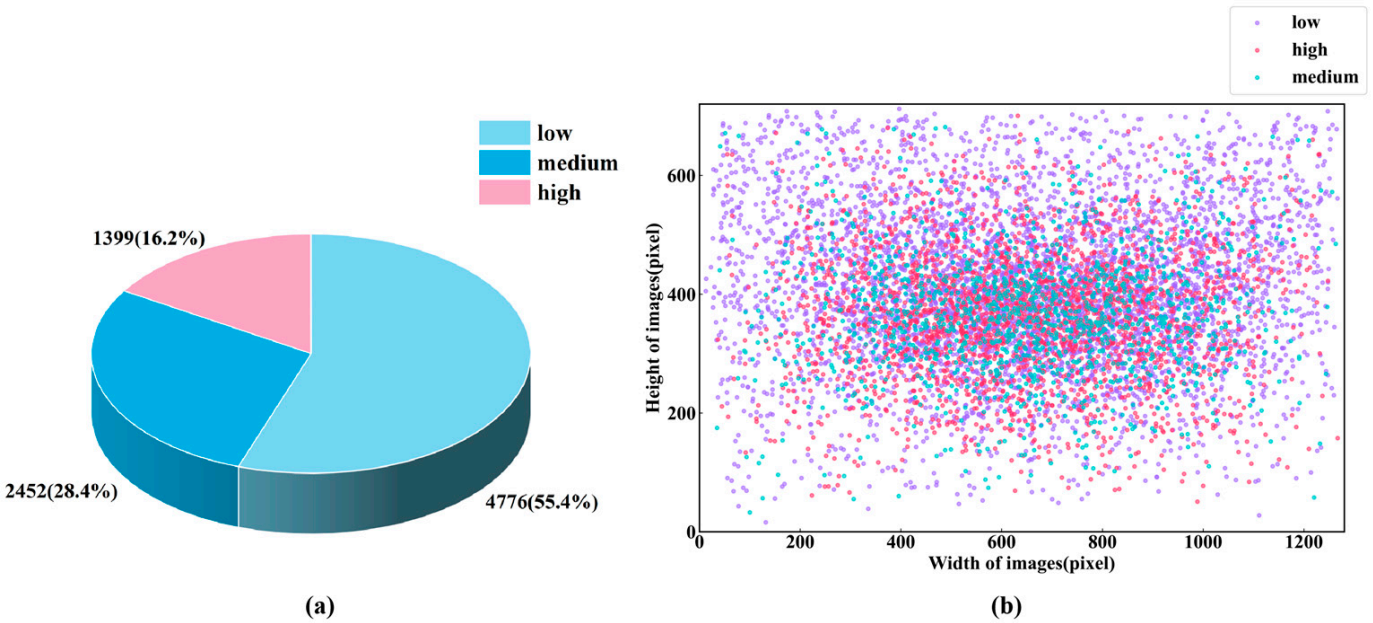

Figure 3. Distribution of training data: (**a**) number of targets in each category; (**b**) target scale distribution (drawn by the authors).

2.2. Overall Architecture of the RAFS-YOLO Model

RAFS-YOLO is a lightweight and high-precision object detection framework optimized specifically for the task of strawberry ripeness detection. The architecture consists of three main components: the Backbone for feature extraction, the Neck for feature fusion, and the Head for accurate classification and localization. The model is based on YOLOv11 [45]. While inheriting its efficient feature extraction capability, it incorporates structural optimizations and module improvements tailored to the characteristics of fruit ripeness detection. These enhancements effectively address the detection challenges in natural environments, such as diverse fruit morphologies and complex lighting conditions, while balancing computational efficiency and deployment flexibility. Figures 4 and 5 illustrate the network structures of the original YOLOv11 and RAFS-YOLO, respectively.

The backbone network plays a crucial role in the object detection system by extracting multi-level feature representations from raw RGB images. RAFS-YOLO inherits and adopts the backbone structure of YOLOv11. This structure has excellent feature extraction efficiency, and can effectively extract multi-scale features in a bottom-up manner: ranging from low-level, high-resolution detailed information to high-level, low-resolution semantic information. On the premise of keeping the overall architecture of the backbone network unchanged, to further improve the model's ability to model spatial contextual relationships and target positions, this study replaces the original Bottleneck with Retblock in C3k2, and designs the CRA module accordingly, thereby enhancing the model's feature expression ability and localization accuracy in complex scenarios.

The Neck network is the core improved component of RAFS-YOLO. In this part, this study introduces HSFPN to replace the original PAN-FPN structure of YOLOv11. HSFPN adopts a top-down multi-path feature fusion mechanism and introduces an advanced feature selection strategy during the fusion process. Specifically, in the process of upsampling high-level semantic features and fusing them with mid-level features, a dynamic upsampling module, DySample, is integrated. This module can adaptively generate sampling weights, enabling more accurate spatial information recovery and boundary detail preservation. Meanwhile, it combines the channel attention mechanism to screen

mid-level features, thereby ensuring the efficient transmission of semantic information. Subsequently, the fused features are further combined with low-level features, and the screening and optimization of low-level features are completed under the guidance of high-level semantics. Through the dynamic upsampling mechanism of DySample, the refined screening of channel attention, and the progressive fusion of multi-scale features, HSFPN significantly enhances the model's ability to recognize strawberry ripeness targets of different scales.

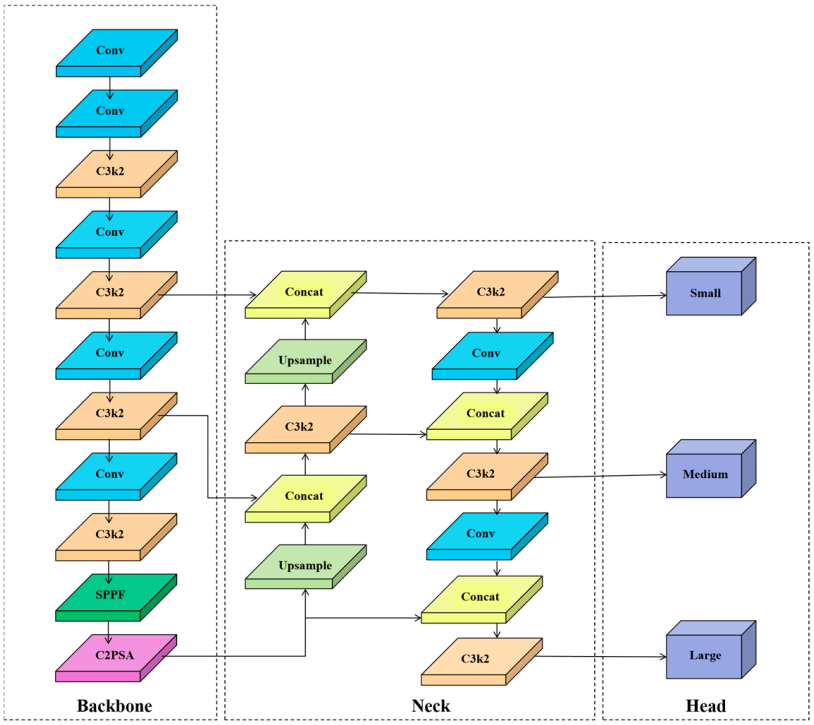

Figure 4. Network structure of the original YOLOv11 (Redrawn by the authors based on [45]).

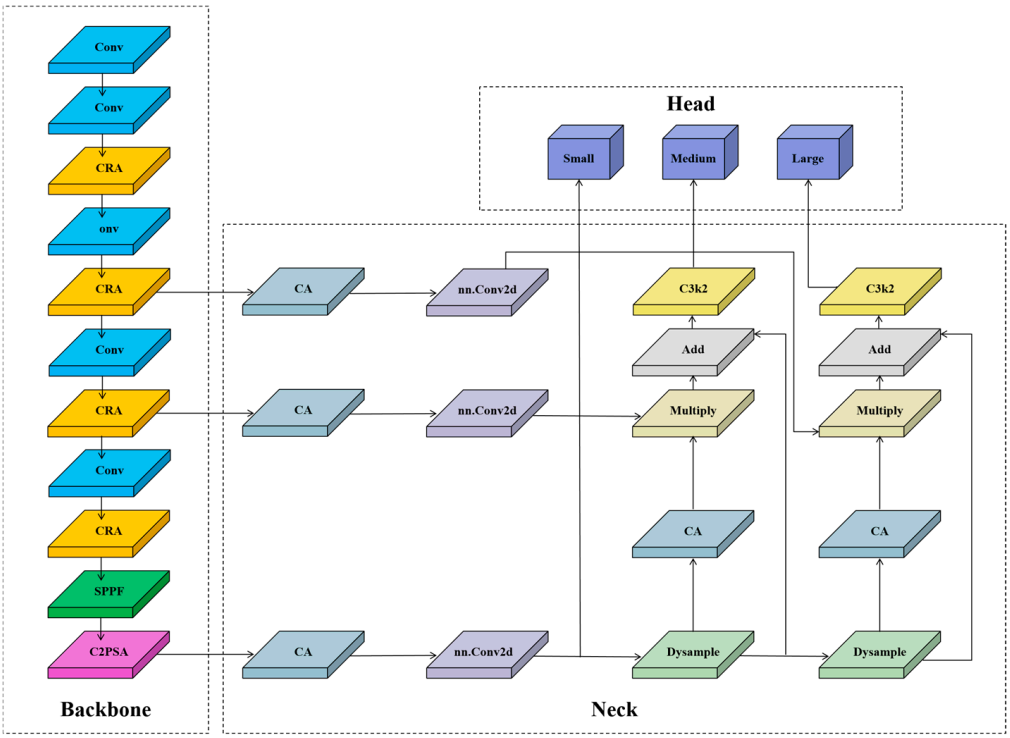

Figure 5. RAFS-YOLO network structure (drawn by the authors).

The detection head of RAFS-YOLO follows the design of YOLOv11. It outputs detection results for feature maps of different scales and adopts a three-scale detection mechanism, which corresponds to the detection of small, medium, and large-sized targets, respectively. Each scale-specific detection head outputs a prediction vector containing bounding box location offsets, object confidence scores, and class probabilities. In this task, the categories are defined as the ripeness states of strawberries, so the detection head can simultaneously predict the probability that each bounding box belongs to each category. The loss function adopts the weighted combination strategy commonly used in the YOLO series, including CIoU regression loss, binary cross-entropy loss for object confidence, and cross-entropy loss for category prediction. Through end-to-end training, the model can achieve accurate localization of strawberry targets and ripeness discrimination in complex backgrounds.

2.3. CRA (Cross-Stage Partial Convolutional Block with Retention Attention)

Although the C3k2 module has excellent local feature extraction capability, it has limitations in modeling the global spatial relationships between dense targets, handling occluded and overlapping scenarios, and its ability to represent edges and small target details is insufficient [46]. To address this, this study designs the CRA module, which significantly enhances the model's ability to model spatial structures and the robustness of feature expression in complex scenarios while keeping computational costs controllable.

The core of CRA lies in the introduction of Retblock. As shown in Figure 6, this module constructs a spatial decay matrix based on Manhattan distance to guide the attention mechanism during computation, enabling it to accurately perceive the relative spatial relationships between different regions, thereby enhancing the model's ability to understand image spatial structures. Specifically, Retblock uses a 2D spatial decay matrix to adjust the attention weights of any two image tokens based on distance, which strengthens the model's ability to focus on key targets and suppresses the interference from backgrounds and non-target regions.

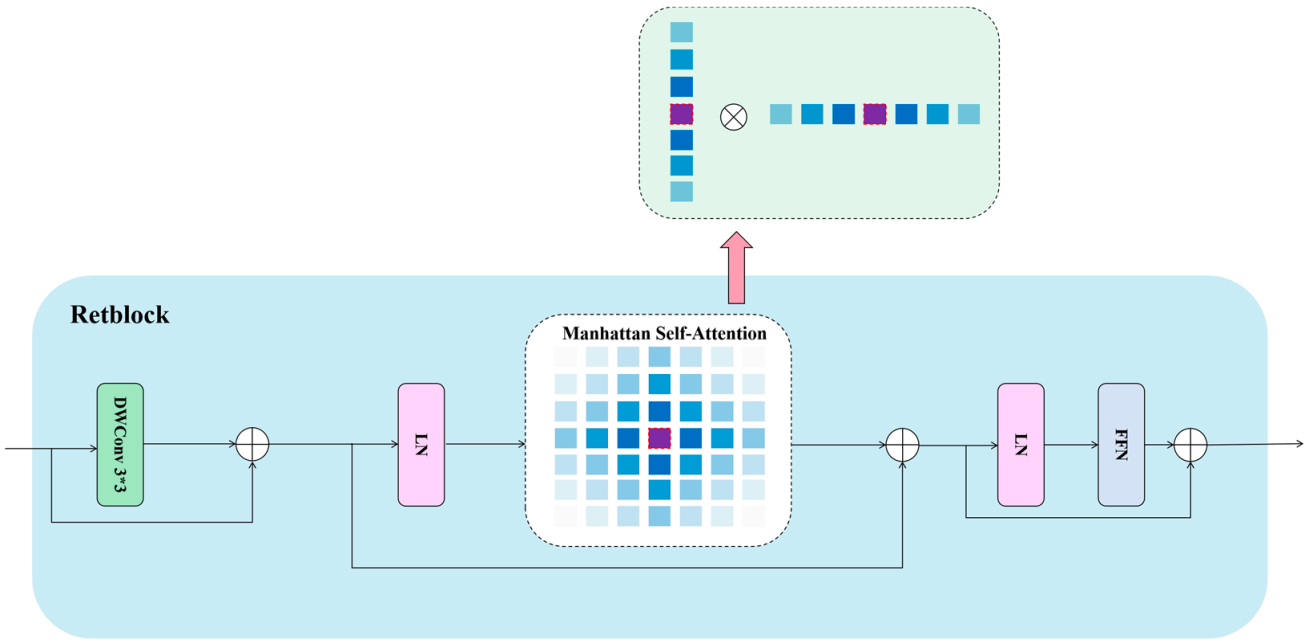

Figure 6. Structure diagram of retblock (Redrawn by the authors based on [47]).

The Manhattan Self-Attention (MaSA) mechanism shown in Figure 6 is developed based on the retention mechanism proposed by RetNet [47]. Unlike the one-dimensional, unidirectional temporal decay in RetNet, MaSA constructs a two-dimensional, bidirectional spatial decay matrix based on Manhattan distance, introducing explicit spatial priors into the visual backbone network, thereby significantly enhancing the ability to model spatial structures. Meanwhile, to reduce the computational complexity of global attention, MaSA adopts an attention form based on horizontal and vertical axis decomposition, synchronously decomposing the complete 2D decay matrix during the attention computation process. With the help of this decomposition strategy, Retblock not only effectively retains spatial prior information but also reduces the complexity of the self-attention mechanism from quadratic to linear, significantly improving computational efficiency. In addition, to further enhance the local feature expression capability of the MaSA module, Depthwise Separable Convolution (DWConv) with contextual awareness is also introduced as a local enhancement mechanism.

In the strawberry target detection task, integrating the CRA module into the backbone network effectively enhances spatial perception capability and local feature expression capability while maintaining computational efficiency. At the mechanism level, MaSA in Retblock accurately models the horizontal and vertical arrangement relationships of strawberries in images through a decoupled 2D attention mechanism, and introduces spatial decay priors based on Manhattan distance, thereby significantly strengthening the model's ability to distinguish densely distributed targets, occluded targets, and overlapping regions. In addition, the introduction of depthwise separable convolution further improves the model's contextual modeling capability for edge information and detailed regions of small targets.

2.4. High-Level Screening Feature Fusion Pyramid Network (HSFPN)

Although the PAN-FPN in YOLOv11 can achieve multi-scale feature fusion, it is insufficient in suppressing redundant information, which easily leads to the accumulation of background interference and has limited discriminative ability in complex scenarios. To address this issue, this paper introduces HSFPN [48]. While maintaining structural efficiency, it effectively enhances the model's multi-scale representation capability and the robustness of feature expression in complex environments through a feature screening mechanism guided by high-level semantics. The structural diagram of HSFPN is shown in Figure 7a.

In HSFPN, the Channel Attention (CA) mechanism is designed to achieve selective enhancement and suppression of features, thereby improving the effectiveness of cross-layer feature fusion. Specifically, for the high-level and low-level feature maps to be fused, global average pooling and max pooling operations are first applied to the high-level feature maps to extract their global semantic representations. Subsequently, the pooling results are input into a channel attention module, which is composed of two 1×1 convolution layers (structurally similar to the SE module). This module is designed to generate a weight vector that is consistent with the number of channels of the high-level features. The weight vector is then normalized via the Sigmoid function, confining the weight of each channel between 0 and 1, which reflects its importance in global semantics. This weight vector is then used to adjust the response intensity of the low-level feature maps: after aligning the channels of the low-level feature maps via 1×1 convolution, element-wise multiplication is performed between the weights and the low-level feature maps along the channel dimension, thereby achieving low-level feature screening under semantic guidance. This mechanism can effectively retain the detailed information related to the high-level response targets, while suppressing redundant features in irrelevant regions, and enhance

the discriminative ability and robustness of the fused features. Its structure is shown in Figure 7b.

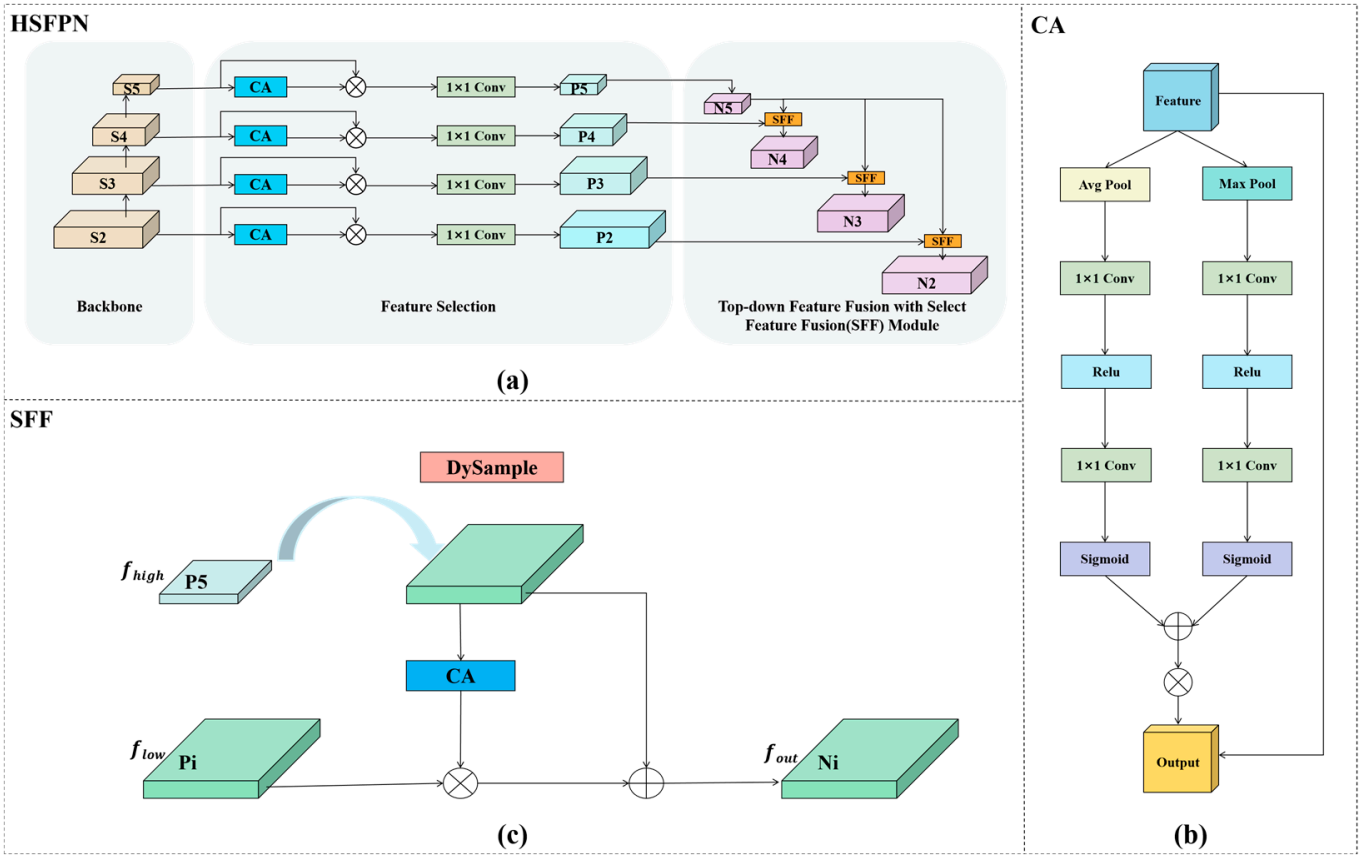

Figure 7. High-level screening feature fusion pyramid network: (**a**) structure of HSFPN; (**b**) structure of CA; (**c**) structure of CA (Redrawn by the authors based on [48]).

After feature screening is completed, the SFF module upsamples the low-level feature maps to match the spatial resolution of the high-level feature maps. Subsequently, it performs element-wise addition with the corresponding high-level features, or compresses the channel dimension via 1×1 convolution after feature concatenation. This enables the effective fusion of semantic information and detailed textures, generating feature representations that contain comprehensive semantics at the current scale. The structure of the SFF module is shown in Figure 7c. In this process, feature fusion unfolds recursively between adjacent levels: high-level features first fuse with middle-level features to generate new middle-level features, which then continue to fuse with low-level features, thereby achieving the gradual downsampling of semantic information. Through this top-down guidance mechanism, HSFPN achieves sufficient interaction and dynamic screening of cross-scale features, effectively suppressing the accumulation of redundant information and significantly enhancing the information flow within the feature pyramid and the multi-scale representation capability.

The multi-scale fused feature maps generated by HSFPN are further transmitted to the detection head for target recognition and maturity discrimination. Due to the introduction of the channel attention mechanism during the fusion process to suppress redundant information and the implementation of feature screening under semantic guidance, the finally output feature maps not only retain rich semantic information but also significantly reduce background interference, which helps improve the recognition accuracy of the detection head in complex environments. Especially in challenging scenar-

ios such as fruit occlusion, high-level features can infer the existence of targets based on locally visible color or shape cues, thereby guiding the retention of key detailed information in low-level features. For example, when strawberry fruits are partially occluded by leaves, high-level features can still perceive the target using local red areas, thereby avoiding the missed detection problem caused by low-level textures being misjudged as the background. By constructing a fusion mechanism guided by high-level semantics and supported by low-level details, HSFPN effectively makes up for the limitations of traditional FPN under challenging conditions such as occlusion and illumination changes, and enhances the discriminative ability and environmental robustness of the model.

2.5. Lightweight Dynamic Upsampling Module (DySample)

During the feature fusion process of HSFPN, feature maps need to be upsampled multiple times to match different scales. Conventional methods use simple interpolation, where the upsampling rate is fixed and treated uniformly across all positions. This may introduce unnecessary details in flat image regions or background regions, while undersampling may occur in regions with rich edge details, failing to reconstruct key information. The DySample module achieves adaptive non-uniform upsampling at different positions by dynamically generating sampling points and weights, thus balancing accuracy and efficiency [49]. Its structure is shown in Figure 8a.

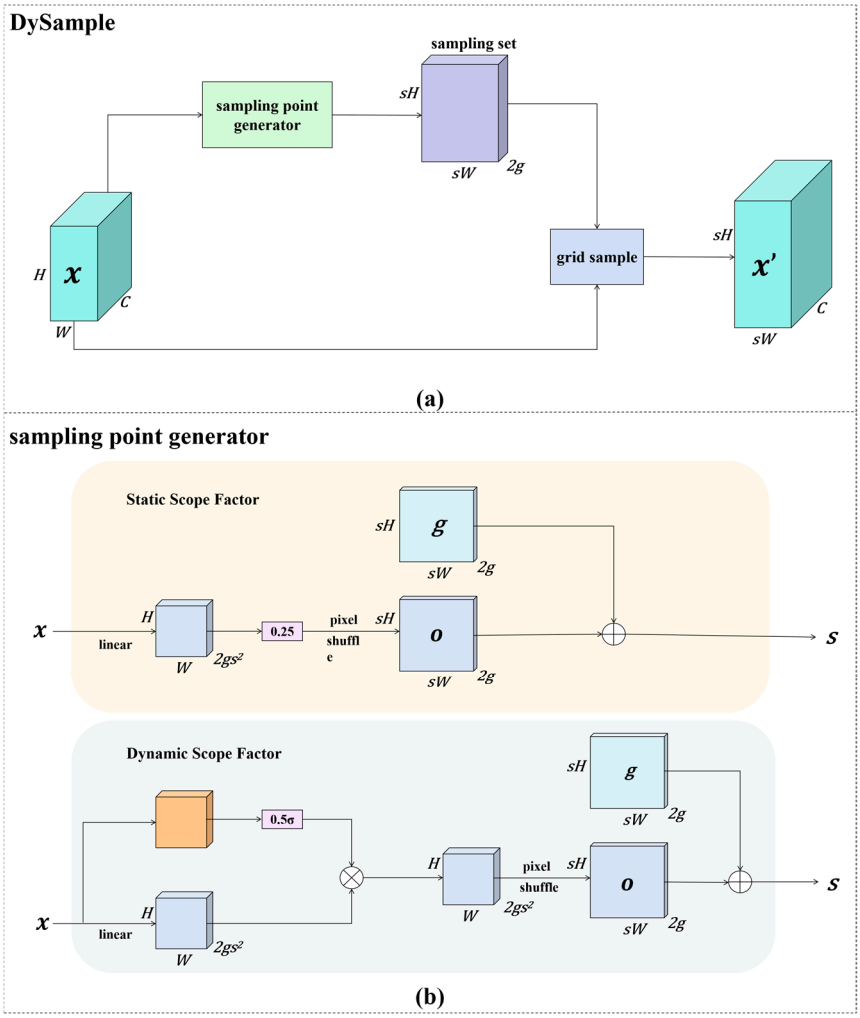

Figure 8. Dynamic upsampling section: (**a**) structure of DySample; (**b**) sampling point generator in DySample. (Redrawn by the authors based on [49]).

In the sampling point generation stage, instead of adopting the traditional method of directly taking values from the neighborhood, DySample adaptively generates candidate sampling points for each position to be interpolated. The initial positions of the sampling points are determined with reference to the uniform distribution strategy of bilinear interpolation, and then adjusted by static or dynamic range factors to improve spatial diversity and expressive ability, as shown in Figure 8b. Among them, the static range factor sets an offset limit according to the upsampling rate to avoid excessive local aggregation of sampling points; the dynamic range factor is adaptively generated by a lightweight sub-network based on local texture features. It shrinks the range in low-frequency regions to reduce redundancy and expands the range in high-frequency regions to enhance context modeling, thereby effectively improving interpolation quality and structural fidelity.

Once the sampling points and their offset positions are determined, DySample extracts the corresponding feature values from the original feature maps and aggregates them through a content-aware weighting mechanism to generate the output features at the target positions. To balance efficiency and lightweight design, it typically adopts a fixed distance-based weighting method instead of an additional attention module. The finally generated feature maps achieve higher reconstruction accuracy in key regions and avoid over-interpolation in regions with simple textures. DySample is fully implemented based on PyTorch tensor operations, requiring no additional parameters and incurring minimal computational overhead. In this paper, DySample is integrated into each upsampling stage of HSFPN to replace the Upsample module of YOLOv11, enabling the upsampled feature maps to better preserve details and significantly enhancing the detection head's ability to model small targets and complex backgrounds, thereby improving the recognition accuracy and confidence in strawberry detection tasks.

2.6. Three-Dimensional Positioning

The visual positioning system calculates the three-dimensional (3D) spatial coordinates of the detected strawberries using the depth information collected by the RGB-D camera. The RGB-D camera can output a depth map that corresponds pixel-wise to the color image, where the depth value of each pixel represents the distance from the target point in that direction to the origin of the camera's optical center, in millimeters. Thus, the 3D spatial position of the strawberries can be accurately reconstructed in the camera coordinate system.

Before using the RGB-D camera, to obtain more accurate three-dimensional coordinate information, this study adopted the chessboard calibration method to calibrate the RGB camera component of the device, so as to acquire its internal parameter matrix and distortion parameters. The reason for choosing to calibrate the RGB camera component is that the RGB image needs to be spatially aligned with the depth map, and its imaging distortion will directly affect the mapping accuracy between pixel coordinates and depth information. In contrast, the depth sensing module is usually calibrated at the factory, and its ranging accuracy is mainly affected by external environmental factors, so there is little need for additional calibration. This method captures multiple images of a planar chessboard at different poses, extracts corner coordinates by combining image processing techniques, thereby establishing the mapping relationship between the world coordinate system and the image coordinate system, and further estimating the internal and external parameters of the camera [50]. The internal parameter matrix A of the camera is shown in Equation (1):

$$A = \begin{bmatrix} f_x & \gamma & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{1}$$

wherein, f_x and f_y represent the focal lengths of the camera in the horizontal and vertical directions, respectively, with the unit of pixels; c_x and c_y denote the pixel coordinates of the image principal point (usually the center of the image); γ represents the skew factor between the coordinate axes, which is zero in an ideal case.

In addition, camera distortion mainly includes radial distortion and tangential distortion, and their calculation formulas are shown in Equations (2) and (3), respectively:

$$x_{distorted} = x\left(1 + k_1 r^2 + k_2 r^4 + k_3 r^6\right) + 2p_1 xy + p_2\left(r^2 + 2x^2\right) \tag{2}$$

$$y_{distorted} = y\left(1 + k_1 r^2 + k_2 r^4 + k_3 r^6\right) + p_1\left(r^2 + 2y^2\right) + 2p_2 xy \tag{3}$$

wherein, x and y are the normalized image coordinates, respectively; $r^2 = x^2 + y^2$ is the squared distance from the point to the optical axis; k_1, k_2 and k_3 are the radial distortion coefficients; p_1 and p_2 are the tangential distortion coefficients.

In this study, for each strawberry target detected by the model, the pixel coordinate (u, v) of the center point of its bounding box is used as the approximate representation of the fruit's position. Considering real-time performance and computational efficiency, a unified center point strategy is adopted in this study. In terms of depth acquisition, the depth value Z corresponding to the position of (u, v) is read from the depth map. To enhance the robustness of depth data, when there are holes or noise in the depth of the center point, we will select a small-scale local window within the target bounding box. After filtering out invalid values, we average the valid depth values to obtain a more stable and reliable depth estimation. During the experiment, a simulated strawberry model was used as the test object. The real-time detection function of the RGB-D camera was enabled, and the color frames and depth frames were aligned simultaneously, thereby realizing pixel-wise depth information reading and 3D coordinate calculation.

In terms of coordinate calculation, based on the pinhole camera imaging model [51], the pixel coordinate (u, v) and its corresponding depth value Z can be converted into the 3D coordinate (X, Y, Z) in the RGB camera coordinate system. The origin of the coordinate system is defined at the camera's optical center, and the calculation formula is shown in Equation (4):

$$X = \frac{(u - c_x) \cdot Z}{f_x}, Y = \frac{(v - c_y) \cdot Z}{f_y}, Z = Z \tag{4}$$

wherein, Z is the depth value of the corresponding pixel in the depth map, that is, the forward distance in the camera coordinate system.

After completing the above coordinate transformation, each strawberry target can obtain its 3D spatial position information in the camera coordinate system, which provides key geometric data support for subsequent high-precision target positioning and picking path planning of mechanical devices. To quantitatively evaluate the accuracy of visual positioning, manual measurement was performed on the detected strawberry targets in the experiment. The specific method is to use a tape measure to obtain their real 3D coordinates, and repeat the measurement twice for the same target to take the average value. Subsequently, this average value is compared with the 3D coordinates calculated by the visual system to verify the accuracy of the positioning results.

2.7. Evaluation Metrics

To comprehensively verify the scientificity and effectiveness of the strawberry recognition and localization model, this paper evaluates the detection performance and localization accuracy of the model from multiple dimensions. Performance metrics include Precision (P), Recall (R), F1-score (F_1), Mean Average Precision (mAP), Coefficient of Determination (R^2), Root Mean Square Error ($RMSE$), and Mean Absolute Error (MAE). In addition, to evaluate the lightweight property and operational efficiency of the model, metrics such as the number of parameters (*Params*), floating-point operations ($FLOPs$), and memory size (*Size*) are introduced to comprehensively measure the model's practicality and deployment potential from multiple perspectives.

P is a metric that measures the proportion of samples actually belonging to the positive class among all samples predicted as the positive class, and its calculation formula is shown in Equation (5):

$$P = \frac{TP}{TP + FP} \tag{5}$$

R refers to the proportion of the number of positive samples correctly predicted by the model to the total number of actual positive samples, and its calculation formula is shown in Equation (6):

$$R = \frac{TP}{TP + FN} \tag{6}$$

F_1 takes P and R as the harmonic mean of these two metrics, providing a balanced measure of model precision and recall; its formula is shown in Equation (7).

$$F_1 = 2 \cdot \frac{P \cdot R}{P + R} \tag{7}$$

mAP is the average of AP across all categories, measuring the comprehensive performance in multi-category detection tasks. AP represents the overall detection performance for a single category, and its calculation formulas are shown in Equations (8) and (9):

$$mAP = \frac{\sum_{i=1}^{k} AP_i}{k} \tag{8}$$

$$AP = \int_0^1 p(r)dr \tag{9}$$

wherein, mAP@50 represents the mean average precision across all categories when the threshold is 0.5.

R^2 is used to measure the model's ability to explain data variation and reflects the degree of fit between predicted values and actual values. Its calculation formula is shown in Equation (10):

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \overline{y})^2} \tag{10}$$

$RMSE$ measures the standard deviation of the differences between measured values and true values. A smaller value indicates more accurate measurements, and its calculation formula is shown in Equation (11):

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2} \tag{11}$$

MAE measures the average of the absolute differences between measured values and actual values, reflecting the average level of prediction errors. Its calculation formula is shown in Equation (12):

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \quad (12)$$

wherein, True Positive (TP) represents the number of samples that the model predicts as positive and are actually positive; False Positive (FP) represents the number of samples that the model predicts as positive but are actually negative; False Negative (FN) represents the number of samples that the model predicts as negative but are actually positive. Here, k denotes the number of sample categories, y_i represents the actual values, $\hat{y}_i$ denotes the measured values, and n represents the number of samples.

2.8. Model Training and Implementation Details

This study was conducted on the PyTorch deep learning framework. For the model training phase of the experiment, the hardware configuration included a 13th Gen Intel (R) Core (TM) i7-13700KF processor (3.40 GHz), 64 GB of RAM, and an NVIDIA GeForce RTX 4080 GPU with 16 GB of video memory, running on the Windows 11 operating system. Ablation experiments for various modules of the model were carried out in this environment. In the spatial localization phase, the hardware configuration used in the experiment consisted of an Intel (R) Core (TM) i5-8300H CPU (2.30 GHz), 8 GB of RAM, an NVIDIA GeForce GTX 1050 Ti GPU with 4 GB of video memory, and the Windows 11 operating system. The RGB-D camera employed was the Intel RealSense Depth Camera D457. The software environment mainly included Python (version 3.10), PyTorch (version 2.2.1), CUDA (version 12.1), and Pyrealsense (version 22.55.1).

All models in the experiments of this paper (including YOLOv11n, YOLOv8n, YOLOv5n, and other models, as well as the proposed RAFS-YOLO) were trained and evaluated under the same configuration and training strategy to ensure the reproducibility and fairness of the results. The input resolution of all models is uniformly set to 640 × 640, the number of training epochs is 150, and the number of data loading workers is 4. The optimizer uses SGD with momentum, where the momentum coefficient is set to 0.937. The initial learning rate is set to 0.01, and a cosine annealing learning rate scheduling strategy is used for dynamic decay with a learning rate decay factor of 0.01. The weight decay coefficient is set to 0.0005. To stabilize the training process, 3 warm-up cycles were introduced, where the warm-up momentum was set to 0.8 and the warm-up learning rate was 0.1. All models applied a consistent Online Data Augmentation strategy, including operations such as brightness, saturation, and hue perturbation, random translation, scaling, cropping, and horizontal flipping, to enhance the generalization ability and robustness of the network. During training, the random seed was fixed (seed = 42), and training and validation were completed in the same hardware and software environment. Through the above unified experimental configuration, this paper ensures that the performance differences between different models only come from the improvement of the network structure, not from differences in training conditions or hyperparameter settings, thus ensuring the scientificity and comparability of the experimental results.

In the spatial positioning stage, an RGB-D camera is used to capture both depth stream and color stream simultaneously, with the resolution of both set to 640 × 480. Intel RealSense D457 is selected because it features high-precision active stereo depth measurement capability, supports hardware-level time synchronization between RGB and depth images, can operate stably under natural light and complex lighting conditions, and is suitable for short-range spatial positioning tasks of fruits. This device can provide reliable 3D information input for the model, ensuring the accuracy and real-time performance of subsequent positioning calculations. First, spatial alignment is performed on the two streams of data to maintain a one-to-one correspondence between the aligned depth image and color image at the pixel level. To ensure the stability of 3D positioning, the model

inference runs at a fixed frequency, with detection executed every 3 s. From the detection results, the central pixel coordinates of each target bounding box are extracted; combined with the depth value corresponding to these coordinates, the coordinates are converted into 3D positions in the camera coordinate system using camera intrinsic parameters. The reason for selecting the central pixel of the detection box as the depth sampling point is that strawberry fruits have an approximately symmetric shape, and their geometric center is usually close to the fruit's centroid, which can accurately reflect the overall depth information of the target. Compared with averaging the entire detection area, central sampling can effectively avoid depth errors caused by leaf occlusion, reflection, or background interference at the fruit edges. When the depth value of the central pixel is invalid or disturbed by noise, a 5×5 pixel window is used within its neighborhood; after filtering the depth data in the window to remove holes and outliers, the average value is calculated to obtain a more stable and reliable depth estimation.

It should be noted that although the input resolution of the model during the training phase is 640×640, while the image resolution captured by the RGB-D camera in practical application is 640×480, the impact of this resolution difference on the overall performance of the system is negligible. This is mainly attributed to two reasons: first, during the inference phase, the system directly uses the original 640×480 images captured by the camera for target detection and positioning without forced stretching or scaling, thus maintaining the geometric consistency between pixel coordinates and camera intrinsic parameters and ensuring the accuracy of the spatial mapping relationship and the reliability of positioning calculations; second, during the training process, an online data augmentation strategy including random scaling, cropping, and padding is introduced, enabling the model to learn features under various input scales, thereby acquiring discriminative features that are insensitive to resolution changes and allowing it to maintain high generalization ability and robustness when facing input resolution differences. Combining the above factors, the combined effect of the geometric consistency maintenance strategy and the data augmentation method effectively reduces the potential negative impact caused by inconsistent resolution, so that the overall system is barely affected in terms of target detection performance and positioning accuracy. In addition, regarding differences in optical structure (such as lens distortion and focal length), field of view, and color reproduction among different cameras, this paper further introduces online color perturbation (including changes in brightness, saturation, and hue) and random affine transformation during the model training phase to improve the network's robustness under different imaging conditions. The feature extraction module of RAFS-YOLO mainly performs discrimination based on the geometric shape and local texture features of fruits, and is insensitive to changes in color style and optical distortion. Therefore, despite differences in optical parameters and imaging characteristics between DSLR and Intel RealSense D457, the model can still maintain stable detection performance and spatial positioning accuracy during the actual inference phase.

3. Results

3.1. Model Selection

YOLOv11 is one of the current state-of-the-art (SOTA) object detection models, offering multiple versions such as YOLOv11n, YOLOv11s, YOLOv11m, YOLOv11l, and YOLOv11x. The main difference between these versions lies in the depth and width of the network. In strawberry detection tasks, to achieve real-time detection and localization of strawberries, the model must not only ensure accuracy but also possess high inference speed, so as to meet the real-time deployment requirements of intelligent agricultural equipment.

To evaluate the performance and stability of models with different scales, independent training was conducted with batch sizes of 8, 16, and 32, respectively. All other training configurations except the batch size were kept consistent, and each experiment was repeated three times. The results are presented in the form of mean ± standard deviation (mean ± std). It should be noted that *Params*, *FLOPs*, and *Size* are inherent attributes of the model structure and are not affected by changes in batch size; among the metrics, the performance indicators *P*, *R*, and *mAP*@50 are used to evaluate the performance stability of the model under different training rates.

As can be seen from Table 1, the performance fluctuation of each model under different training rates is extremely small, and the standard deviations of *P*, *R*, and *mAP*@50 are all lower than 0.005, indicating that the overall stability of the models is good. Considering speed, accuracy, and model scale comprehensively, YOLOv11n is finally selected as the baseline model for strawberry target detection.

Table 1. Experimental results of different models of YOLOv11.

Model	*P* (Mean ± std)	*R* (Mean ± std)	*mAP*@50 (Mean ± std)	*Params* (M)	*FLOPs* (G)	*Size* (MB)
YOLOv11n	0.876 ± 0.002	0.864 ± 0.001	0.926 ± 0.003	2.58	6.3	5.2
YOLOv11s	0.882 ± 0.003	0.874 ± 0.004	0.931 ± 0.002	9.41	21.3	18.3
YOLOv11m	0.874 ± 0.002	0.872 ± 0.002	0.936 ± 0.002	20.03	67.7	38.6
YOLOv11l	0.881 ± 0.001	0.873 ± 0.002	0.931 ± 0.003	25.28	86.6	48.8
YOLOv11x	0.885 ± 0.002	0.878 ± 0.001	0.934 ± 0.002	56.83	194.4	109.1

3.2. Ablation Experiments

To fully verify the effectiveness of the RAFS-YOLO model and conduct an in-depth analysis of the independent contributions and synergistic effects of each sub-module, a systematic ablation study is carried out in this section. By gradually introducing the three modules (CRA, HSFPN, and DySample) and testing their different combinations, the impact of each module on model performance and their synergistic effects are evaluated quantitatively.

In the training phase, to further verify the model's stability under different training rates, three batch sizes (8, 16, and 32) are set for independent training, while all other training configurations except the batch size are kept consistent. All experiments are repeated three times, and the results are presented in the form of mean ± standard deviation (mean ± std), as shown in Table 2. The results indicate that the standard deviations of *P*, *R*, and *mAP*@50 are all less than 0.005, which proves that the model maintains stable performance under different training rates and verifies the reproducibility of the experiments and the reliability of the results. Subsequent performance analysis and discussion are all based on the experimental mean values to systematically evaluate the impact of each module and their combinations on the overall detection performance.

In all single-module experiments, each functional module was independently integrated into and evaluated on the same baseline model (YOLOv11n), ensuring the comparability and independence of the analysis regarding their performance contributions. After the CRA module was introduced alone, the retained attention mechanism it brought significantly enhanced the model's ability to perceive key features. By incorporating the attention weighting operation into the cross-stage feature fusion process, the model's capability to distinguish fine-grained features was effectively improved. This improvement was particularly evident in object detection scenarios involving targets with highly similar colors and morphologies, such as strawberry recognition in natural environments, where the model exhibited stronger discriminative power. Experimental results show that introducing the CRA module not only further reduces computational complexity—with *Params* decreasing

from 2.58 M to 2.47 M and *FLOPs* dropping from 6.3 G to 6.1 G—but also improves *P* from 0.876 to 0.889 and *mAP*@50 from 0.926 to 0.928. These results verify the dual advantages of the CRA module in enhancing detection accuracy and achieving model lightweighting.

Table 2. Results of Ablation Experiments.

yolov11n	CRA	HSFPN	Dysample	*P* (Mean ± std)	*R* (Mean ± std)	*mAP*@50 (Mean ± std)	*Params* (M)	*FLOPs* (G)	*Size* (MB)
√				0.876 ± 0.002	0.864 ± 0.003	0.926 ± 0.002	2.58	6.3	5.2
√	√			0.889 ± 0.001	0.852 ± 0.002	0.928 ± 0.002	2.47	6.1	5.0
√		√		0.871 ± 0.003	0.899 ± 0.002	0.931 ± 0.003	2.07	7.0	4.2
√			√	0.888 ± 0.002	0.877 ± 0.001	0.924 ± 0.004	2.45	5.9	4.8
√	√	√		0.882 ± 0.003	0.869 ± 0.004	0.933 ± 0.003	1.95	7.2	4.0
√	√		√	0.907 ± 0.002	0.871 ± 0.003	0.935 ± 0.001	2.49	6.2	5.0
√		√	√	0.901 ± 0.004	0.869 ± 0.003	0.937 ± 0.002	1.75	5.0	3.6
√	√	√	√	0.918 ± 0.001	0.902 ± 0.002	0.946 ± 0.002	1.63	4.8	3.4

In the experiment where the HSFPN module was introduced alone, the model's ability to perceive and represent multi-scale features was further enhanced. By incorporating efficient feature screening and fusion strategies, HSFPN effectively suppressed redundant information in feature representation and strengthened the ability to express high-level semantic features, thereby improving the detection performance for small targets and occluded targets. Experimental results show that while the model's *Params* were reduced to 2.07 M, *R* was significantly increased from 0.864 to 0.899, and *mAP*@50 was increased from 0.926 to 0.931. The above results indicate that the HSFPN module can effectively enhance target coverage capability while maintaining the lightweight structure of the model, making it particularly suitable for the detection of strawberry targets in complex natural backgrounds.

DySample is designed to improve the information restoration capability during the upsampling stage of feature maps. It achieves adaptive reconstruction of spatial features through a dynamic weight generation mechanism, thereby enhancing the expression quality of high-resolution features. When the DySample module was introduced alone, although *mAP*@50 decreased slightly to 0.924, this result does not indicate the module itself is ineffective. Instead, it suggests that its dynamic feature reconstruction capability has not been fully exerted without the support of semantic enhancement and multi-scale fusion. This performance fluctuation may be mainly related to two factors: on the one hand, dynamic sampling may introduce slight feature offsets in scenarios with relatively stable spatial structures, thereby weakening the detection head's perception of high-level semantic information; on the other hand, dynamic weight prediction may cause gradient fluctuations in the early training stage, making it difficult for feature expression to converge in a timely manner. Subsequent experiments further verified this analysis: when DySample is used in synergy with high-level feature fusion modules (such as CRA and HSFPN), its dynamic feature reconstruction capability is fully unleashed, and the overall detection performance is significantly improved. This demonstrates the synergistic effect of the DySample module in enhancing accuracy while maintaining model lightweighting.

To further explore the interactive relationships and synergistic effects among various functional modules, this study conducted additional combined ablation experiments to systematically analyze the performance of different modules under the conditions of pairwise combination and full-module integration. Experimental results show that the synergy between CRA and HSFPN can organically integrate the attention enhancement mechanism with multi-scale semantic expression capability, thereby significantly improving the model's feature representation ability and detection accuracy—with *P* increased to 0.882, *R*

reaching 0.869, and *mAP*@50 reaching 0.933. The combination of CRA and DySample forms an effective complement through spatial structure perception and feature reconstruction strategies, further improving detection performance, with *P* and *R* reaching 0.888 and 0.877, respectively. Meanwhile, the synergistic introduction of HSFPN and DySample achieves a better balance between detection accuracy and model complexity: *P* is increased to 0.901, *mAP*@50 reaches 0.937, and *Params* is reduced to 1.75 M, demonstrating dual advantages in performance improvement and lightweighting. The above results fully indicate that there is not only significant complementarity between different modules but also synergistic gains when used in combination, thereby enhancing the model's feature expression ability and object detection performance.

When the three modules (CRA, HSFPN, and DySample) are synergistically integrated, the constructed RAFS-YOLO model achieves the optimal performance across all metrics. Specifically, *P* is increased to 0.918, *R* reaches 0.902, and *mAP*@50 climbs to 0.946—representing improvements of 4.2%, 3.8%, and 2.0%, respectively, compared to the baseline model. This indicates that the proposed module combination can significantly enhance the model's detection accuracy. Meanwhile, *Params* is reduced to 1.63 M, *FLOPs* is decreased to 4.8 G, and *Size* is compressed to 3.4 MB, which are 36.82%, 23.8%, and 34.62% lower than those of the baseline model, respectively. These results fully demonstrate the lightweight advantages of the model in structural design.

Comprehensive analysis shows that although each module can independently contribute to performance improvement and complexity optimization when introduced alone, their complementarity and synergistic gains become more prominent after synergistic integration. This not only achieves a better balance between detection accuracy and computational efficiency but also significantly enhances the model's overall expression capability and application value. The systematic and comprehensive ablation experimental results strongly verify the rationality and effectiveness of RAFS-YOLO's structural design, and highlight its application value and promotion potential in strawberry target detection tasks.

To further quantify the impact of each module on computational efficiency and resource occupancy, this paper conducts a quantitative comparative analysis of the latency (approximated by *FLOPs*) and memory requirements (approximated by *Params* and *Size*) of the CRA, HSFPN, and DySample modules. The results show that while improving detection performance, each module has a relatively limited impact on computational and storage overhead. Specifically, while enhancing the spatial relationship modeling capability, the CRA module reduces *FLOPs* from 6.3 G to 6.1 G, *Params* from 2.58 M to 2.47 M, and *Size* from 5.2 MB to 5.0 MB, demonstrating higher computational efficiency and parameter utilization. The *FLOPs* of the HSFPN module increase slightly to 7.0 G, but *R* improves significantly, indicating that it helps enhance detection coverage and multi-scale feature expression capabilities. Through a dynamic upsampling strategy, the DySample module effectively compresses the model scale, reducing *Params* from 2.58 M to 2.45 M, shrinking *Size* to 4.8 MB, and keeping *FLOPs* at approximately 5.9 G. Thus, it significantly reduces memory occupancy while maintaining stable performance. Overall, CRA and DySample focus more on computation and storage optimization, while HSFPN mainly improves detection coverage and multi-scale feature modeling capabilities. The three modules have good complementarity in terms of latency and memory, collectively forming the efficient structural design advantages of the RAFS-YOLO model.

Figure 9 intuitively compares the performance of six models (A–F) in terms of key performance metrics, presenting the ablation experiment results through normalized bar charts. The results show that the RAFS-YOLO proposed in this paper (Version F) not only significantly improves detection accuracy but also achieves effective compres-

sion of the model structure, thus possessing both high-precision and lightweight advantages. It can be clearly seen from the figure that Version F has reached the optimal level in terms of *P*, *R* and *mAP*@50, while having the lowest number of model parameters, *FLOPs*, and *Size*, fully reflecting the ideal combination of high precision and low computational cost.

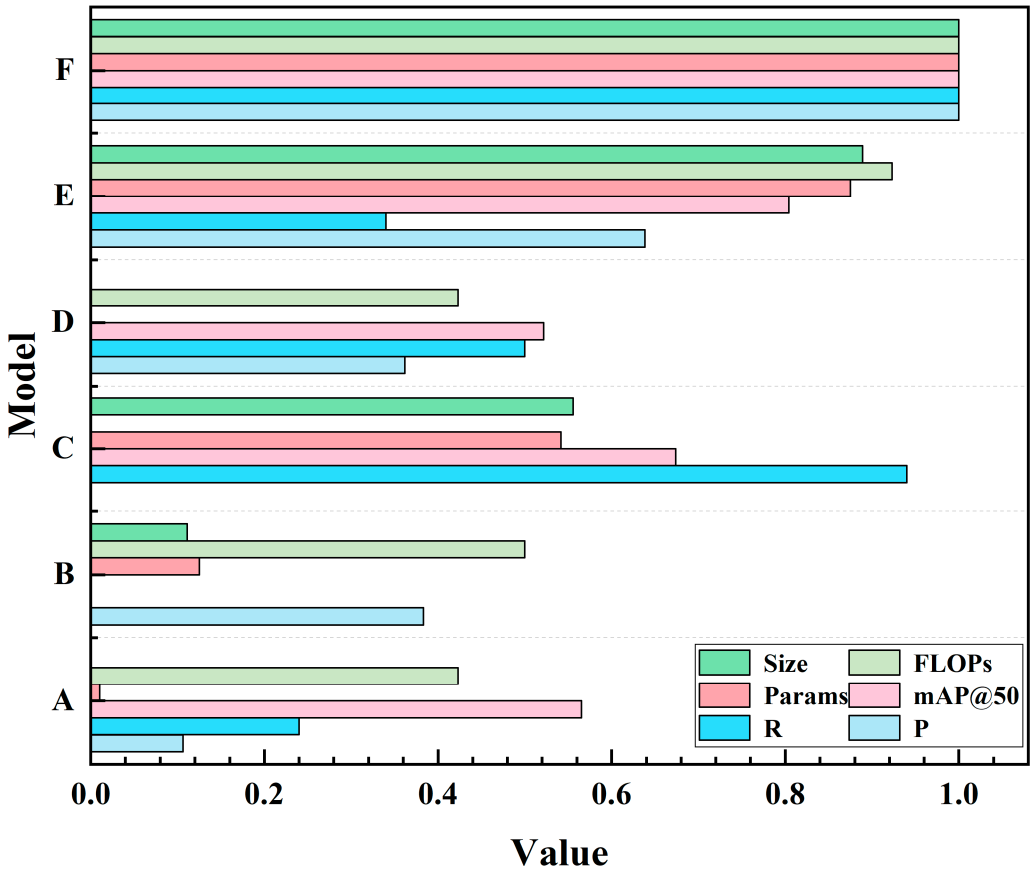

Figure 9. Normalized analysis of ablation experiments (drawn by the authors).

As shown in Figure 10, in simple scenarios, there is little difference in the detection performance of strawberries at different maturity stages between the baseline model and RAFS-YOLO. However, in complex environments, the baseline model often exhibits low or unstable detection confidence, while RAFS-YOLO can consistently maintain high and stable detection performance under the same test conditions, demonstrating its stronger robustness in recognizing targets with low contrast and indistinct differences in shape and color. Combined with the heatmaps, it can be further observed that the activation regions of RAFS-YOLO for strawberries at various maturity stages are more concentrated on the main body of the fruit and have a clearer distribution. In contrast, the activation range of the baseline model is more scattered, and some even fall into the background areas. This indicates that RAFS-YOLO can more effectively capture key features related to strawberry maturity, thereby enhancing its ability to distinguish subtle texture and color differences. In addition, the activation patterns of RAFS-YOLO on fruits of different maturity stages show consistency. That is, regardless of changes in illumination, increases in background complexity, or diversification of fruit postures, its feature extraction always focuses on the core regions, thus ensuring high detection accuracy and robustness even in complex scenarios.

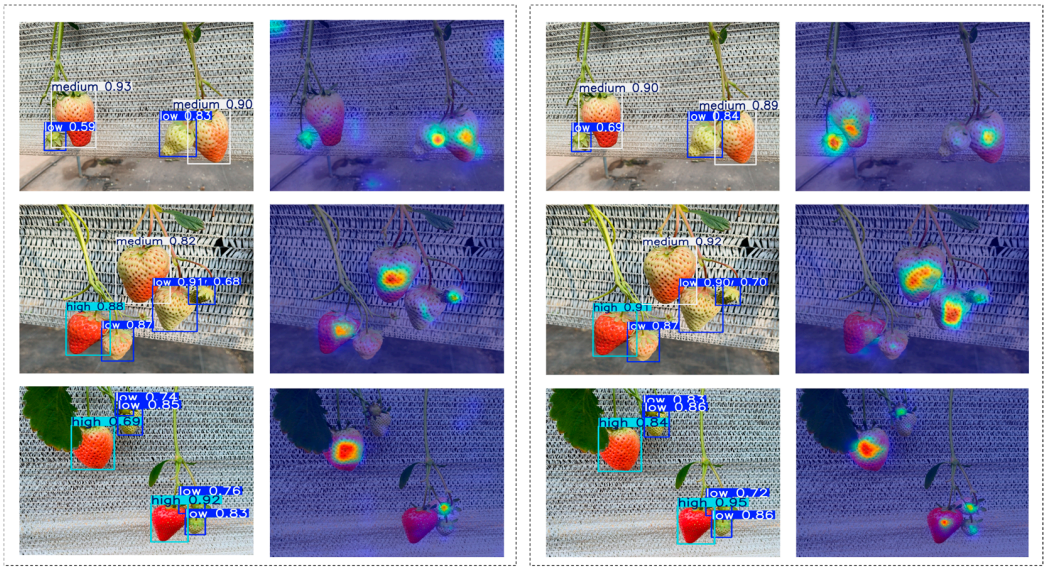

Figure 10. Example detection results and heatmaps on the test set: (**a**) YOLOv11n; (**b**) RAFS-YOLO (drawn by the authors).

To further analyze the model's detection performance across different maturity categories, a statistical analysis of the model's detection results was conducted during the testing phase, with the results shown in Figure 11. The confusion matrix presents the prediction distribution of the model across low, medium, high maturity, and background categories; the diagonal elements represent the number of correctly detected samples, while the off-diagonal parts reflect the confusion relationships between different categories. The results indicate that the model can accurately distinguish strawberry fruits at different maturity stages, with most samples detected correctly. The main errors stem from confusion between low-maturity fruits and the background category, primarily because immature fruits have a green surface—their color and texture features are similar to those of leaves or background areas, leading the model to easily misdetect when distinguishing boundaries. In addition, there is a small amount of confusion between adjacent maturity stages, mainly due to the subtle differences in color and texture features of fruits during the transition phase, which makes the model prone to ambiguous judgments when distinguishing features. Overall, the model maintains high detection accuracy and stability across all categories, verifying the effectiveness and robustness of RAFS-YOLO in strawberry maturity detection tasks.

In addition, to verify the real-time performance and resource occupancy of the model on the actual deployment hardware platform, inference performance tests were conducted in this study on an Intel i5-8300H CPU and an NVIDIA GTX 1050 Ti GPU. The test resolution was consistent with that in the training phase (640 × 640). The results show that the average inference latency of RAFS-YOLO is approximately 118.48 ms (about 8.44 FPS), with an overall memory occupancy of about 117.89 MB; in contrast, YOLOv11n has an inference latency of 214.6 ms (about 4.66 FPS) and an overall memory occupancy of approximately 135.38 MB. It can be seen from this that RAFS-YOLO outperforms the baseline model in both inference speed and resource utilization efficiency, verifying its application potential in real-time detection and edge deployment tasks.

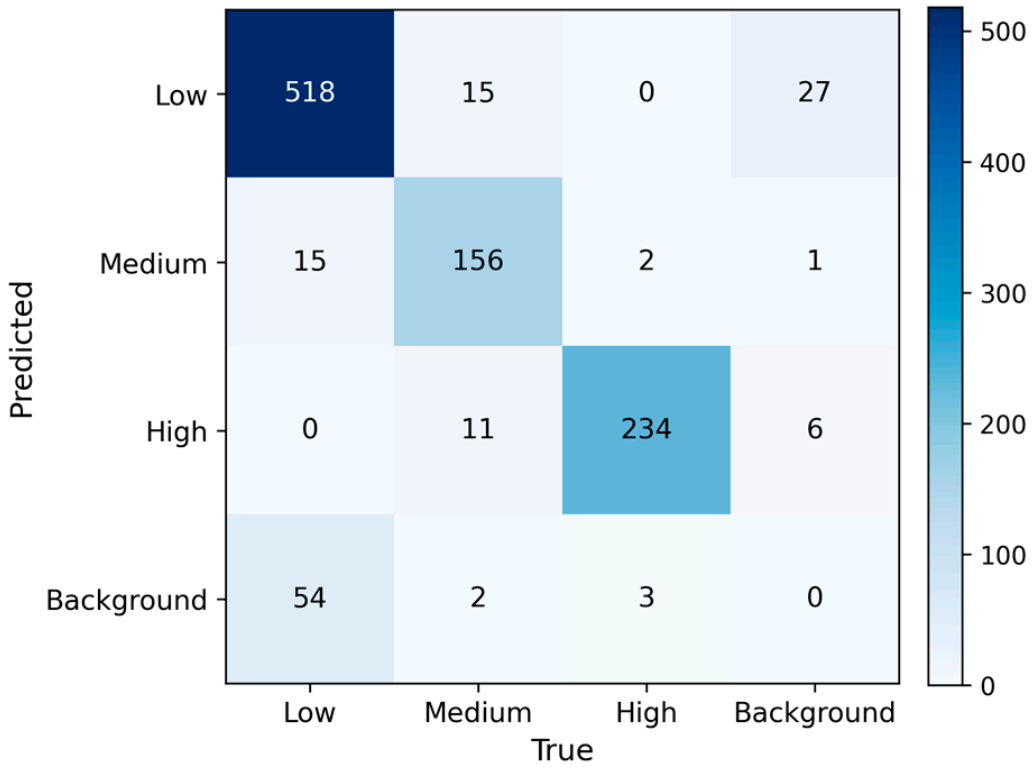

Figure 11. Confusion matrix (drawn by the authors).

3.3. Analysis of Detection Performance for Different Maturity Categories

To further evaluate the impact of category sample imbalance on model detection performance and verify whether the proposed RAFS-YOLO model has recognition bias across different maturity categories, this study calculated evaluation metrics including P, R, F_1, and $mAP@50$ for three types of samples (low maturity, medium maturity, and high maturity), respectively, and conducted a comparative analysis with the baseline model YOLOv11n. The experimental results are shown in Figure 12.

As shown in the results of Figure 12a, the P of RAFS-YOLO across all maturity categories is higher than that of the baseline model. Specifically, the P for low-maturity category increases from 0.892 to 0.925, for medium-maturity category from 0.844 to 0.895, and for high-maturity category from 0.891 to 0.933. The P of the three sample categories all show improvements to varying degrees, and their overall values are relatively close. This indicates that the model maintains consistency in discriminative ability for both majority-class and minority-class targets, with no performance degradation caused by category sample imbalance.

As shown in the results of Figure 12b, RAFS-YOLO also demonstrates stable and balanced detection capability in terms of the R metric. The R for low-maturity category increases from 0.841 to 0.882, for medium-maturity category from 0.802 to 0.847, and for high-maturity category from 0.950 to 0.977—with R showing an overall upward trend across all three categories. The results indicate that the model does not exhibit significant fluctuations in detection coverage due to differences in the number of samples across categories, and can maintain stable recognition performance among targets of different maturity levels.

The F_1 metric further validates the conclusion of the aforementioned performance improvement, as shown in Figure 12c. The F_1 of RAFS-YOLO for low-, medium-, and high-maturity targets reach 0.903, 0.871, and 0.954, respectively, all showing significant increases compared to the baseline model's 0.866, 0.822, and 0.920. Among them, the F_1 improvement

for the medium-maturity category is the most notable, reaching 4.9%. Meanwhile, the F_1 of all categories remain at a high level, indicating that the model achieves a good balance between detection precision and recall capability, with no performance degradation for minority classes caused by category sample imbalance.

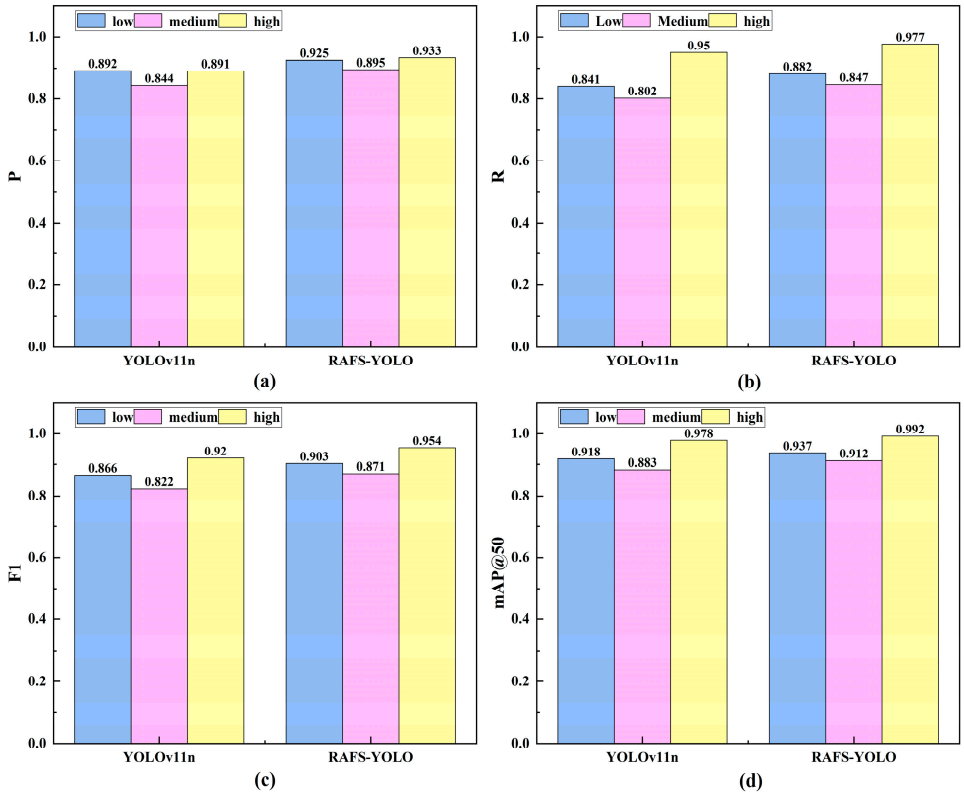

Figure 12. Comparison results of detection performance between RAFS-YOLO and YOLOv11n under different maturity categories: (**a**) P; (**b**) R; (**c**) F_1; (**d**) mAP@50 (drawn by the authors).

The results of the mAP@50 metric also confirm this conclusion, as shown in Figure 12d. The mAP@50 values of RAFS-YOLO for targets of low-, medium-, and high-maturity reach 0.937, 0.912, and 0.992, respectively—all higher than the baseline model's 0.918, 0.883, and 0.978. Additionally, the performance of all three categories remains at a high level. This result indicates that the improved model also has no bias toward minority classes or performance degradation issues in terms of overall detection accuracy.

Comprehensive analysis indicates that RAFS-YOLO achieves varying degrees of improvement over the baseline model in the four metrics P, R, F_1, and mAP@50. Moreover, the detection performance across all maturity categories shows good overall balance, with no performance bias toward minority classes caused by category sample imbalance. These results fully demonstrate that the proposed method maintains robust discriminative ability and good category generalization ability, even under category sample imbalance, and can achieve balanced performance among low-maturity, medium-maturity, and high-maturity categories. This provides reliable technical support for subsequent fruit grading and automatic harvesting.

3.4. Comparative Experiments

To comprehensively evaluate the performance of the proposed RAFS-YOLO model, this study conducts comparative experiments with mainstream single-stage object detection algorithms, including TOOD, SSD, YOLOv5, YOLOv6, YOLOv7, YOLOv8, YOLOv9,

YOLOv10, and RT-DETR. All models are trained and tested on the same hardware platform under unified training conditions to ensure the fairness and comparability of the results.

To verify the model's stability under different training rates, the experiments are independently run three times with batch sizes of 8, 16, and 32, respectively, while all other training settings except the batch size are kept consistent. The experimental results are presented in the form of mean ± standard deviation (mean ± std), as shown in Table 3. The results indicate that the standard deviations of P, R, and mAP@50 are all lower than 0.005, which verifies the reproducibility of the experiments and the stability of the results. Subsequent analyses are all based on experimental mean values to ensure the accuracy and consistency of model performance evaluation.

Table 3. Experimental results of comparisons between different algorithms.

Model	*P* (Mean ± std)	*R* (Mean ± std)	*mAP*@50 (Mean ± std)	*Params* (M)	*FLOPs* (G)	*Size* (MB)
TOOD	0.917 ± 0.002	0.897 ± 0.002	0.945 ± 0.003	32.02	167.7	244.0
SSD	0.875 ± 0.002	0.884 ± 0.001	0.931 ± 0.004	24.01	30.53	184.0
YOLOv5n	0.9085 ± 0.003	0.858 ± 0.003	0.936 ± 0.001	2.50	7.1	5.0
YOLOv6n	0.8751 ± 0.001	0.894 ± 0.003	0.937 ± 0.002	4.23	11.8	8.3
YOLOv7-tiny	0.875 ± 0.004	0.881 ± 0.002	0.922 ± 0.002	6.01	13.0	11.6
YOLOv7	0.901 ± 0.003	0.892 ± 0.004	0.944 ± 0.003	36.49	103.2	71.3
YOLOv8n	0.8997 ± 0.002	0.871 ± 0.002	0.931 ± 0.003	3.01	8.1	6.0
YOLOv8s	0.8759 ± 0.003	0.898 ± 0.001	0.937 ± 0.002	11.12	28.4	21.5
YOLOv9c	0.8657 ± 0.002	0.893 ± 0.004	0.933 ± 0.003	25.32	102.3	49.2
YOLOv10n	0.8788 ± 0.001	0.869 ± 0.003	0.922 ± 0.003	2.27	6.5	5.5
RTDETR-l	0.8375 ± 0.001	0.802 ± 0.002	0.889 ± 0.002	31.98	103.4	63.1
RAFS-YOLO	0.918 ± 0.001	0.902 ± 0.002	0.946 ± 0.002	1.63	4.8	3.4

As can be seen from the experimental results in Table 3, RAFS-YOLO demonstrates significant performance advantages in this comparative experiment. Its three key indicators—P, R, and mAP@50—reach 0.918, 0.902, and 0.946, respectively, and its overall performance is superior to that of current mainstream single-stage object detection models. In terms of the mAP@50 indicator, RAFS-YOLO outperforms TOOD (0.945), YOLOv7 (0.944), and YOLOv9c (0.933), highlighting its leading position among high-precision object detection models. Additionally, in the category of lightweight models, RAFS-YOLO also performs prominently: its mAP@50 value exceeds that of YOLOv8n (0.931), YOLOv7-tiny (0.922), YOLOv10n (0.922), and YOLOv5n (0.936), while achieving significant improvements in both P and R. These results further verify that RAFS-YOLO possesses excellent robustness and generalization capabilities in scenarios such as complex backgrounds, occlusion interference, and small object detection.

Compared with YOLOv7 and YOLOv9c, the performance improvement of RAFS-YOLO is mainly attributed to its targeted optimizations in multi-scale feature fusion and spatial modeling strategies. Based on the E-ELAN backbone structure and the conventional FPN/PAN fusion mechanism, YOLOv7 exhibits good gradient propagation capability, but there is still information redundancy in cross-layer feature aggregation. YOLOv9c improves path integration through the GELAN backbone and the PGI module; however, its feature fusion mainly relies on general aggregation, lacking selective filtering of high-level semantics and explicit modeling of spatial relationships. To address the above shortcomings, RAFS-YOLO introduces three key modules—CRA, HSFPN, and DySample—based on the YOLOv11 framework. Specifically, the CRA module enhances spatial relationship modeling capability through multi-level contextual feature fusion, enabling the model to achieve accurate localization even in scenarios with dense targets or partial occlusions. The HSFPN structure realizes hierarchical filtering and efficient aggregation of semantic information during multi-scale feature fusion, which effectively reduces cross-layer feature redundancy and improves the detection performance for small targets and edge targets. The DySample

dynamic upsampling module can adaptively adjust the sampling strategy according to input features; while enhancing the expression capability of high-resolution features, it reduces computational redundancy, thereby improving the model's real-time performance and resource utilization efficiency.

Benefiting from the aforementioned structural improvements, RAFS-YOLO significantly reduces model complexity while ensuring high-precision detection. In terms of model complexity and computational resource overhead, the *Params*, *FLOPs*, and *Size* of RAFS-YOLO are 1.63 M, 4.8 G, and 3.4 MB, respectively, all lower than those of all comparative models in the table. It outperforms lightweight models such as YOLOv5n (2.50 M, 7.1 G, 5.0 MB), YOLOv7-tiny (6.01 M, 13.0 G, 11.6 MB), and YOLOv8n (3.01 M, 8.1 G, 6.0 MB). It can be concluded that RAFS-YOLO achieves an excellent balance among accuracy, speed, and resource occupancy, demonstrating stronger model compression capability and deployment flexibility. It is particularly suitable for resource-constrained embedded platforms and edge computing scenarios.

Figure 13 shows the performance of RAFS-YOLO and various mainstream object detection models in the detection task of strawberries at different maturity stages. It can be seen that RAFS-YOLO maintains excellent detection performance across all maturity stages, and its overall performance remains at the forefront. Especially in the detection of low-maturity and high-maturity strawberries, RAFS-YOLO achieves 0.937 and 0.992, respectively, outperforming a variety of high-precision models including TOOD, YOLOv7, and YOLOv9c. This indicates that it has stronger feature perception ability when dealing with targets with blurred boundaries, small sizes, and indistinct textures. For the detection of medium-maturity strawberries, although the overall accuracy is lower than that of high-maturity ones, RAFS-YOLO still outperforms lightweight models such as YOLOv8n, YOLOv7-tiny, and YOLOv10n, demonstrating good stability and generalization ability. As can be seen from Figure 14, RAFS-YOLO can stably detect all strawberry targets in the images and exhibit higher accuracy in identifying samples at different maturity stages. In contrast, YOLOv9c and YOLOv10n show low confidence or unstable recognition for some targets. Although YOLOv8n has acceptable overall detection accuracy, its accuracy in identifying low-maturity strawberries is insufficient, with a certain risk of missed detection. Comprehensive analysis shows that while maintaining reliable detection of high-maturity strawberries, RAFS-YOLO can more accurately distinguish samples at medium and low maturity stages, highlighting its robustness and adaptability in multi-stage maturity target recognition, thereby further verifying its performance advantages in complex scenarios.

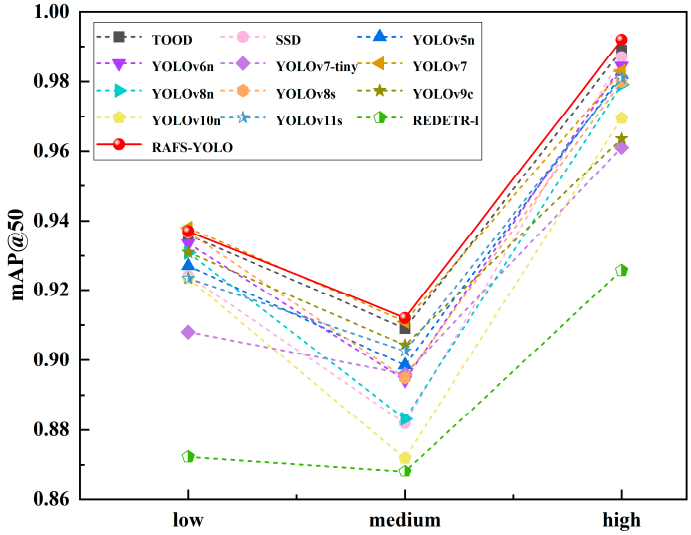

Figure 13. Detection results of strawberries at different maturity stages by various algorithms on the test set (drawn by the authors).

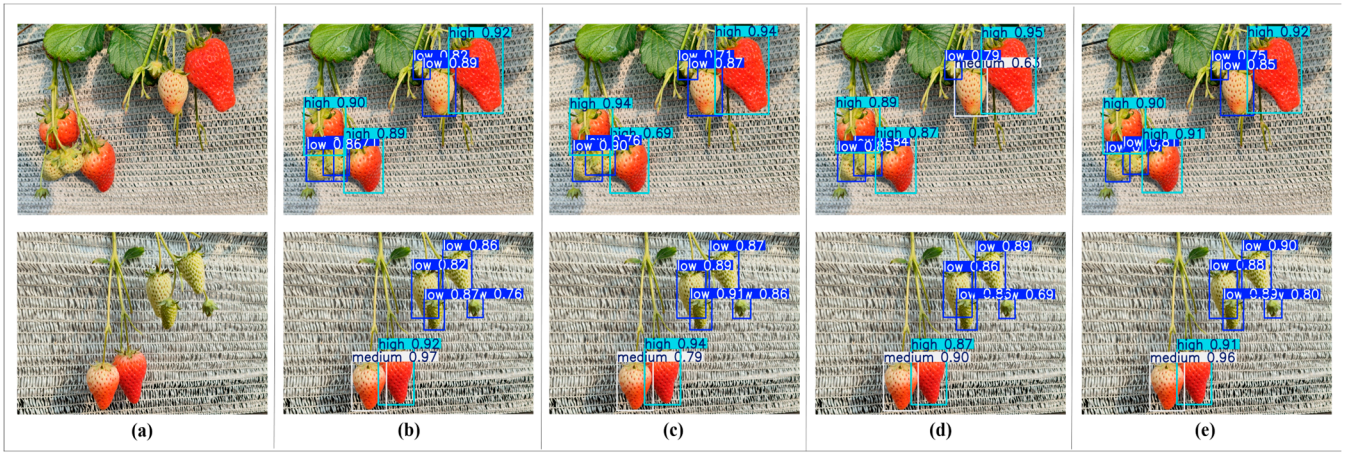

Figure 14. Detection results of partial models: (**a**) Original image; (**b**) RAFS-YOLO; (**c**) YOLOv10n; (**d**) YOLOv9c; (**e**) YOLOv8n (drawn by the authors).

3.5. Results of Spatial Localization Experiment

To improve the accuracy of the spatial localization experiment, this study first calibrated the RGB camera before officially conducting the experiment. As shown in Figure 15a, a chessboard pattern consisting of 6 columns × 9 rows of squares (each square with a side length of 25 mm) was printed, corresponding to 5 columns × 8 rows of internal corners, which was used for subsequent corner detection and calibration. Subsequently, we collected 30 chessboard images at a resolution of 640 × 480. During the collection process, the chessboard was moved to present multiple viewing angles, thereby enhancing the diversity of calibration data and improving the calibration effect. In the calibration process, the camera calibration toolkits of OpenCV [52] and Matlab [53] (Version R2022b) were used, respectively, to calibrate the camera, and finally the result with the smaller average re-projection error was selected. Based on the internal parameter matrix and distortion parameters calculated from this result, subsequent experiments were carried out.

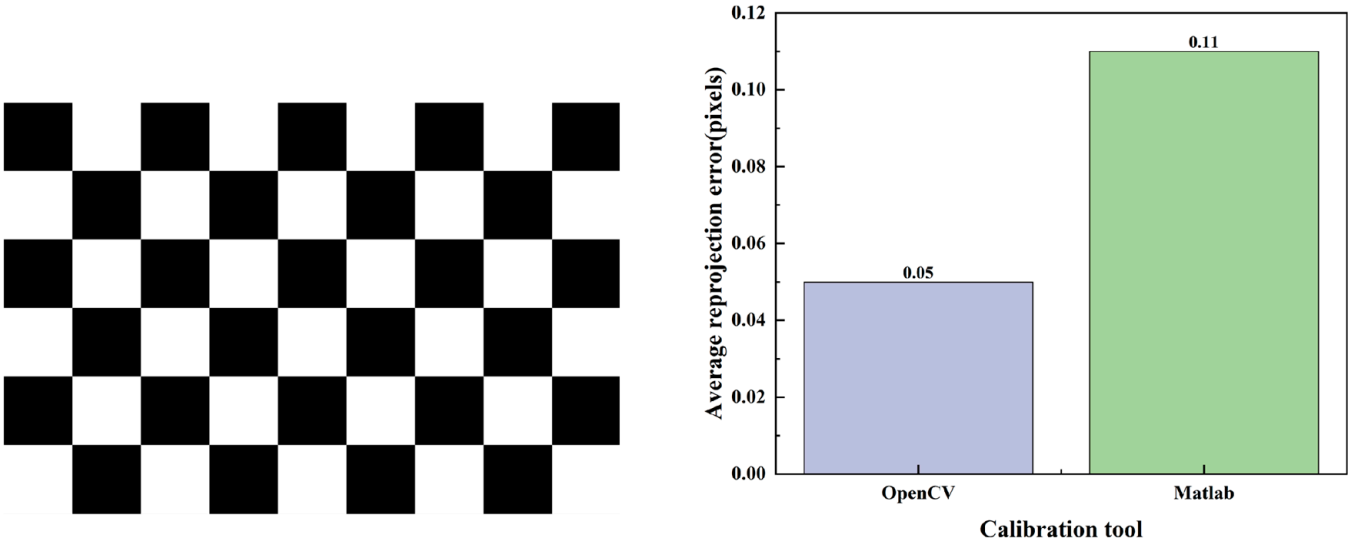

Figure 15. Calibration results: (**a**) chessboard pattern; (**b**) average re-projection error (drawn by the authors).

As shown in Figure 15b, the average re-projection error obtained by OpenCV calibration is 0.05 pixels, which is significantly better than the 0.11 pixels obtained by MATLAB calibration. Therefore, we chose to use the calibration results of OpenCV for subsequent spatial localization experiments. The camera's internal parameter matrix and distortion parameters are detailed in Table 4.

Table 4. Internal Parameters of RGB Camera.

Parameter	RGB Camera
Internal Parameter Matrix	$\begin{bmatrix} 391.365 & 0 & 309.722 \\ 0 & 393.260 & 266.262 \\ 0 & 0 & 1 \end{bmatrix}$
Radial Distortion Coefficients	$[-0.034 \quad 0.0344 \quad -0.008]$
Tangential Distortion Coefficients	$[0.015 \quad -0.006]$

In the spatial positioning experiment, the horizontal distance between the camera and the strawberry simulation model was set within the range of 30–60 cm, and image data were collected within this distance range. During the experiment, the relative position between the strawberry simulation model and the camera was adjusted by moving the model, and 30 target points were randomly selected within this range for measurement. For each measurement point, the proposed positioning algorithm was first used to estimate its 3D coordinate values; subsequently, a tape measure was used to perform two independent measurements under the same conditions, and the average value was taken to obtain the corresponding real 3D coordinates, thereby improving the reliability of the measurement results. During the experiment, to ensure the independence of samples and the stability of data collection, the model inference was run at a fixed frequency, with detection performed every 3 s. It should be noted that this 3 s detection interval was only used to control the sampling frequency for stable acquisition of image samples in a fixed scenario, and did not represent the actual inference speed of the system. Meanwhile, to evaluate the potential impact of external lighting conditions on depth measurement accuracy and spatial positioning performance, this study simulated imaging conditions under different

lighting environments by adjusting the fill light intensity, and completed all experiments under the same distance measurement range and measurement process. To ensure the scientificity and comprehensiveness of error analysis, the 3D coordinate deviations were processed as absolute errors, and the positioning errors in the three directions (X, Y, and Z) were calculated, respectively. At the same time, the Euclidean distance error between the camera and the fruit was introduced as a comprehensive evaluation index to more comprehensively assess the positioning accuracy of the system. The experimental results are shown in Figure 16.

Figure 16 presents the error analysis results of the spatial positioning experiment, where Figure 16a–c correspond to the positioning accuracy along the X, Y, and Z axes, respectively, and Figure 16d reflects the fitting of the overall Euclidean distance error. As can be seen from the figure, all fitting curves are basically coincident with the ideal reference line $y = x$, and the correlation coefficient R^2 exceeds 0.998, indicating a very high consistency between the measured values and the true values. In terms of specific error metrics, the RMSE values in the X, Y, and Z directions are 0.21 cm, 0.23 cm, and 0.30 cm, respectively, with corresponding MAE values of 0.17 cm, 0.18 cm, and 0.27 cm. The positioning deviation in all three coordinate directions is controlled at the millimeter level, and the errors in the X and Y directions are slightly better than those in the Z direction. Meanwhile, the $RMSE$ and MAE of the overall Euclidean distance error are 0.31 cm and 0.27 cm, respectively, with R^2 still reaching 0.9984, further verifying the high accuracy and robustness of the proposed method in 3D spatial positioning tasks.

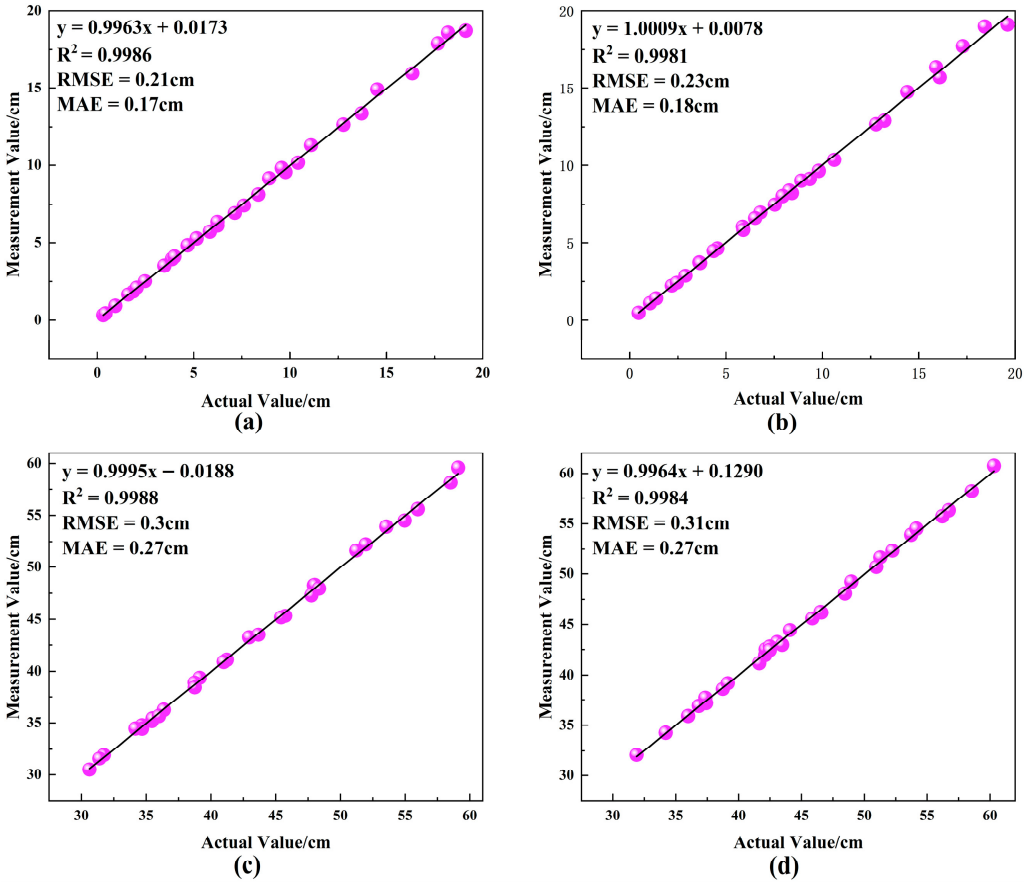

Figure 16. Diagram of localization error analysis: (**a**) X-axis; (**b**) Y-axis; (**c**) Z-axis; (**d**) Euclidean distance (drawn by the authors).

To further analyze the positioning performance of the system under different distance measurement ranges, the horizontal distance between the camera and the target was

divided into three intervals: 30–40 cm, 40–50 cm, and 50–60 cm, and the positioning error metrics for each interval were calculated separately, as shown in Table 5. Overall, the 3D Euclidean distance error remains at a low level across all distance intervals, indicating that the system can maintain high positioning stability under different working distances; within the 30–40 cm interval, the positioning accuracy is the highest, with the Euclidean distance $RMSE$ of 0.18 cm and MAE of 0.15 cm; as the distance increases to 50–60 cm, the positioning error rises slightly, with the $RMSE$ increasing to 0.35 cm and the MAE increasing to 0.33 cm, which is mainly due to the reduction in effective pixel density and the increase in noise at longer distances. From the perspective of the error variation trend along each coordinate axis, the X-axis error is generally small and decreases gradually with increasing distance, with the $RMSE$ decreasing from 0.26 cm to 0.13 cm and the MAE decreasing from 0.22 cm to 0.11 cm, indicating that horizontal positioning is less affected by distance changes and maintains stable accuracy; the Y-axis error rises slightly within the 40–50 cm interval, with the $RMSE$ increasing from 0.11 cm to 0.28 cm and the MAE increasing from 0.10 cm to 0.22 cm, but falls slightly within the 50–60 cm interval, with the $RMSE$ of 0.24 cm and the MAE of 0.19 cm, suggesting that this direction is more sensitive to local lighting and viewing angle; the Z-axis error increases significantly with increasing distance, with the $RMSE$ rising from 0.20 cm to 0.39 cm and the MAE rising from 0.18 cm to 0.38 cm, indicating that the measurement accuracy in the depth direction is most significantly affected by distance changes. Although there are slight fluctuations in errors across different directions, the errors in all dimensions still remain at the millimeter level, demonstrating good stability and engineering applicability.

Table 5. Errors in different distance intervals.

Distance Interval (cm)	$RMSE_X$	MAE_X	$RMSE_Y$	MAE_Y	$RMSE_Z$	MAE_Z	$RMSE_{ED}$	MAE_{ED}
30–40	0.26	0.22	0.11	0.1	0.2	0.18	0.18	0.15
40–50	0.21	0.18	0.28	0.22	0.27	0.25	0.33	0.3
50–60	0.13	0.11	0.24	0.19	0.39	0.38	0.35	0.33

Under different lighting conditions within a controllable range, the system still maintains excellent measurement accuracy and stability, with the overall error remaining within the millimeter-level range. This fully verifies the feasibility and application value of the proposed method for precise fruit positioning in complex agricultural environments. This result further indicates that in short-range measurement scenarios such as strawberry picking, changes in external light intensity have an extremely limited impact on the depth measurement quality of the RGB-D camera, so their impact on positioning errors can also be considered negligible. In this case, the overall positioning accuracy of the system mainly depends on the measurement accuracy of the algorithm rather than external lighting conditions.

Meanwhile, to verify the real-time performance of the system, end-to-end processing latency was tested on the used hardware platform. The results show that the average latency of the system from image acquisition to detection, 3D positioning, and output is approximately 138.89 ms (about 7.2 FPS), which verifies that the proposed method maintains high accuracy while possessing real-time processing capability and can meet the real-time operation requirements of intelligent picking tasks.

4. Discussion

This paper proposes a visual recognition and 3D localization method, which can achieve high-precision detection of strawberry fruit maturity and accurate spatial localization. Experimental results show that RAFS-YOLO achieves better performance than

YOLOv11n in the strawberry maturity detection task: its P, R, and $mAP@50$ are improved by 4.2%, 3.8%, and 2%, respectively, demonstrating stronger robustness and recognition capability in complex scenarios. Meanwhile, the model reduces the *Params* by 36.82%, lowers the *FLOPs* by 23.81%, and compresses the *Size* by 34.62%, exhibiting excellent lightweight characteristics and providing strong support for the optimization of operation efficiency and embedded deployment. Overall, RAFS-YOLO achieves a better balance between detection accuracy and model complexity, demonstrating comprehensive performance superior to current mainstream object detection models. In terms of 3D localization, this method also exhibits high accuracy and consistency: the R^2 in the X, Y, and Z directions are 0.9986, 0.9981, and 0.9988, respectively, with corresponding $RMSE$ of 0.21 cm, 0.23 cm, and 0.30 cm, and MAE of 0.17 cm, 0.18 cm, and 0.27 cm; the R^2 of the Euclidean distance between two points is 0.9984, with $RMSE$ of 0.31 cm and MAE of 0.27 cm. The overall results indicate that this method can achieve millimeter-level error control in all dimensions and maintain extremely high consistency ($R^2 > 0.998$), providing reliable spatial coordinate support for path planning and grasping operations of picking robotic arms.

Based on YOLOv11, this study proposes an improved object detection model, which designs and integrates three core improvements: the CRA feature extraction module, the HSFPN high-level screening feature fusion module, and the DySample lightweight dynamic upsampling module, to effectively enhance detection performance and efficiency. Among them, the CRA improves the model's ability to delineate target boundaries under complex background conditions by strengthening position awareness and feature expression capabilities; the HSFPN effectively enhances the model's ability to characterize the maturity of strawberry fruits at different scales through selective screening and efficient fusion of multi-scale features, thereby improving detection accuracy and robustness; the DySample optimizes the upsampling process based on a point sampling mechanism, balancing lightweight properties with the reconstruction accuracy of edge and fine-grained features. Relying on the above designs, the model achieves a good balance between detection accuracy and computational efficiency, demonstrating potential in meeting both real-time and high-precision requirements in agricultural scenarios. Furthermore, combined with the millimeter-level depth information provided by the RGB-D camera, a joint integrated method of detection and localization is realized, which shows better environmental adaptability and application potential compared with methods relying solely on single-modal vision.

Compared with traditional 3D localization methods, the RGB-D perception scheme adopted in this paper has significant advantages in terms of structural simplicity and application cost. Binocular vision systems rely on disparity matching to calculate depth, which is easily affected by illumination, occlusion, and texture changes, resulting in insufficient ranging stability. Although lidar (Light Detection and Ranging) has high precision and strong robustness, its high equipment cost, large size, and high energy consumption make it difficult to be promoted in small agricultural machinery. In contrast, RGB-D cameras can directly output dense depth information within short-to-medium distance ranges, avoiding complex stereo matching processes, and have good fusion compatibility with visual detection models. Combined with the lightweight RAFS-YOLO model, this study realizes efficient 3D localization that balances precision and real-time performance, achieving a good balance among performance, cost, and system complexity. It is particularly suitable for intelligent perception tasks in greenhouses and short-distance picking scenarios.

This study verifies the feasibility of combining deep learning detection algorithms with task-specific customized modules. When traditional general detection models are directly applied to agricultural scenarios, their performance is often limited due to insuf-

ficient generalization ability. To address this issue, we conducted targeted optimizations on the model by incorporating the unique visual characteristics of strawberry crops (such as maturity discrimination dominated by color, small target size and susceptibility to occlusion). Favorable application results have been achieved in practical tasks. This outcome suggests that for other crops, customized model optimization may also be one of the important approaches to improve recognition performance. In addition, while balancing detection accuracy and computational efficiency, the proposed method also demonstrates good deployment potential in terms of embedded device deployment, providing valuable references for equipping agricultural machinery with intelligent vision systems. With the rapid development of smart agriculture, visual models that balance high precision and lightweight properties are expected to have broader application demands in the future, and this study provides valuable exploration and practical experience for this purpose.

Despite the excellent performance of the combined scheme of RAFS-YOLO and RGB-D camera in this experiment, its application still has certain limitations. First, the experimental scenarios of this study are mainly focused on short-distance operation tasks for strawberry picking, and the experimental environment is indoor controlled conditions. Systematic evaluation has not been conducted under environments such as strong illumination, specular reflection, and complex outdoor lighting. Therefore, the depth measurement accuracy and stability of RGB-D cameras under such conditions remain uncertain. Although the experimental results show that the measurement error is small under conventional illumination, in strong light, strong reflection, or complex outdoor environments, external infrared interference may weaken the quality of depth signals, leading to a decline in positioning accuracy. Future research will conduct performance verification in complex outdoor scenarios to address the above issues, and consider introducing optical filtering and protection designs (such as infrared filters and light shields) to reduce illumination and reflection interference. At the same time, combined with multi-sensor fusion strategies (such as lidar, multispectral, or polarization imaging), the robustness and depth measurement stability of the system in complex natural environments will be further improved.

Second, the detection performance of the model largely depends on the scale and diversity of training data. However, the dataset used in this study is mainly derived from indoor red strawberry collection scenarios, which cannot fully cover the complex and variable environmental conditions in agricultural production. Therefore, the generalization capability of the model may be limited to a certain extent. To alleviate this issue, an online data augmentation strategy is introduced in the training phase. By means of random rotation, flipping, brightness adjustment, and color perturbation, the diversity of samples is expanded, thereby improving the model's robustness under different lighting and strawberry posture conditions. Nevertheless, since the data source still focuses primarily on red strawberry samples, the applicability of the model to other varieties (such as white strawberries) and different picking environments needs further verification. Future research will consider combining transfer learning and multi-source domain adaptation methods to further improve the model's cross-variety and cross-environment detection performance, thereby enhancing its practicality and stability.

In addition, the classification of strawberry ripeness levels mainly relies on manual visual judgment (with red color proportion $<$30%, 30–80%, and $>$80%). Although double independent labeling and cross-verification have improved consistency, certain subjective biases may still exist due to differences in individual visual perception. Different labelers may have inconsistent judgments under changing lighting conditions or in the color transition areas of strawberries, thereby affecting the category boundaries of some samples. Future research will consider introducing objective measurement tools (such as colorimeters

or spectral analyzers) to verify some samples, or establishing a color perception consistency calibration and evaluation process, so as to further improve the standardization and portability of ripeness labeling.

Finally, although the model's lightweight design has significantly reduced the number of parameters and computational complexity, its real-time performance and energy efficiency on extremely resource-constrained embedded platforms still need further optimization to meet the application requirements of ultra-low power consumption and low latency.

5. Conclusions

Aiming at the application requirements of intelligent picking, this paper proposes a high-precision visual recognition and 3D localization method to improve the accuracy of strawberry maturity identification and spatial localization precision. First, a lightweight RAFS-YOLO object detection model is designed, which introduces CRA, HSFPN, and DySample based on YOLOv11. In this model, CRA enhances the model's position awareness and expression capabilities in object detection; HSFPN strengthens the model's perception and representation abilities for targets of different scales; DySample constructs an upsampler from the perspective of point sampling, improving the efficiency of the upsampler. Experimental results show that the model designed in this paper effectively improves the recognition accuracy of strawberry targets while significantly reducing *Params*, *FLOPs*, and *Size*, demonstrating excellent lightweight advantages. The model still exhibits good robustness under conditions of partial occlusion, light changes, and complex background interference. Meanwhile, by fusing the trained model with depth information collected by an RGB-D camera, a visual localization system is constructed, which can output the spatial 3D coordinates of fruit targets accurately while detecting them in real time. The localization precision reaches the millimeter level, providing reliable and efficient visual perception support for intelligent picking. Furthermore, the model has undergone inference performance verification on the actual deployment hardware platform. The results show that while ensuring detection accuracy, it also has high operational efficiency and low resource occupancy, indicating that this method has good implementability and engineering promotion potential on edge computing devices. In future research, the recognition stability of the model under complex lighting conditions can be improved by fusing multi-spectral information, and its cross-scene adaptability can be enhanced by combining methods such as multi-scene data augmentation, domain adaptation, and meta-learning. At the same time, lightweight technologies such as model pruning can be introduced to reduce computational complexity, thereby further improving the flexibility of the model in practical deployment and laying a technical foundation for the deep coupling of visual recognition and mechanical control. In addition, future work will further verify the model's robustness and generalization ability in real field picking environments and on external datasets, so as to comprehensively evaluate its performance and promotability in practical application scenarios.

Author Contributions: Conceptualization, K.L. and W.Z.; methodology, K.L.; software, X.W.; validation, K.L., Q.W. and X.W.; formal analysis, W.Z.; investigation, Q.W.; resources, W.Z.; data curation, K.L.; writing—original draft preparation, K.L.; writing—review and editing, Q.W.; visualization, X.W.; supervision, W.Z.; project administration, W.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The article contains original data from this study, which should be available upon reasonable request by contacting the corresponding author.

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

References

1. Zhang, J.; Kang, N.; Qu, Q.; Zhou, L.; Zhang, H. Automatic fruit picking technology: A comprehensive review of research advances. *Artif. Intell. Rev.* **2024**, *57*, 54. [CrossRef]
2. Yu, Y.; Xie, H.; Zhang, K.; Wang, Y.; Li, Y.; Zhou, J.; Xu, L. Design, development, integration, and field evaluation of a ridge-planting strawberry harvesting robot. *Agriculture* **2024**, *14*, 2126. [CrossRef]
3. Santos, A.A.; Schreurs, C.; da Silva, A.F.; Pereira, F.; Felgueiras, C.; Lopes, A.M.; Machado, J. Integration of artificial vision and image processing into a pick and place collaborative robotic system. *J. Intell. Robot. Syst.* **2024**, *110*, 159. [CrossRef]
4. Pal, A.; Leite, A.C.; From, P.J. A novel end-to-end vision-based architecture for agricultural human–robot collaboration in fruit picking operations. *Robot. Auton. Syst.* **2024**, *172*, 104567. [CrossRef]
5. 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. [CrossRef]
6. Wei, C.Z.; Zaman, M.M.; Ibrahim, M.F. Visual servo algorithm of robot arm for pick and place application. *J. Kejuruter.* **2024**, *36*, 891–898. [CrossRef]
7. He, Z.; Liu, Z.; Zhou, Z.; Karkee, M.; Zhang, Q. Improving picking efficiency under occlusion: Design, development, and field evaluation of an innovative robotic strawberry harvester. *Comput. Electron. Agric.* **2025**, *237*, 110684. [CrossRef]
8. Yang, S.; Wang, W.; Gao, S.; Deng, Z. Strawberry ripeness detection based on YOLOv8 algorithm fused with LW-Swin Transformer. *Comput. Electron. Agric.* **2023**, *215*, 108360. [CrossRef]
9. Rosada, R.; Hussein, Z.M.; Novamizanti, L. Evaluating YOLO Variants for Real-Time Multi-Object Detection of Strawberry Quality and Ripeness. In Proceedings of the 2025 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT), Bali, Indonesia, 3–5 July 2025; pp. 483–490.
10. Huang, F.; Zheng, J.; Liu, X.; Shen, Y.; Chen, J. Polarization of road target detection under complex weather conditions. *Sci. Rep.* **2024**, *14*, 30348. [CrossRef] [PubMed]
11. Daza, A.; Ramos, K.Z.; Paz, A.A.; Dario, R.; Rivera, M. Deep learning and machine learning for plant and fruit recognition: A systematic review. *J. Syst. Manag. Sci.* **2024**, *14*, 226–246. [CrossRef]
12. Maheswari, P.; Raja, P.; Karkee, M.; Raja, M.; Baig, R.U.; Trung, K.T.; Hoang, V.T. Performance analysis of modified deeplabv3+ architecture for fruit detection and localization in apple orchards. *Smart Agric. Technol.* **2025**, *10*, 100729. [CrossRef]
13. Naito, H.; Shimomoto, K.; Fukatsu, T.; Hosoi, F.; Ota, T. Interoperability analysis of tomato fruit detection models for images taken at different Facilities, Cultivation Methods, and Times of the Day. *AgriEngineering* **2024**, *6*, 1827–1846. [CrossRef]
14. Subeesh, A.; Kumar, S.P.; Chakraborty, S.K.; Upendar, K.; Chandel, N.S.; Jat, D.; Dubey, K.; Modi, R.U.; Khan, M.M. UAV imagery coupled deep learning approach for the development of an adaptive in-house web-based application for yield estimation in citrus orchard. *Measurement* **2024**, *234*, 114786. [CrossRef]
15. Endo, K.; Hiraguri, T.; Kimura, T.; Shimizu, H.; Shimada, T.; Shibasaki, A.; Suzuki, C.; Fujinuma, R.; Takemura, Y. Estimation of the amount of pear pollen based on flowering stage detection using deep learning. *Sci. Rep.* **2024**, *14*, 13163. [CrossRef] [PubMed]
16. Du, L.; Zhang, R.; Wang, X. Overview of two-stage object detection algorithms. *J. Phys. Conf. Ser.* **2020**, *1544*, 012033. [CrossRef]
17. Juntao, X.; Zhen, L.; Linyue, T.; Rui, L.; Rongbin, B.; Hongxing, P. Visual detection technology of green citrus under natural environment. *Trans. Chin. Soc. Agric. Eng.* **2018**, *49*, 46–52.
18. Wang, P.; Niu, T.; He, D. Tomato young fruits detection method under near color background based on improved faster R-CNN with attention mechanism. *Agriculture* **2021**, *11*, 1059. [CrossRef]
19. Kong, X.; Li, X.; Zhu, X.; Guo, Z.; Zeng, L. Detection model based on improved faster-RCNN in apple orchard environment. *Intell. Syst. Appl.* **2024**, *21*, 200325. [CrossRef]
20. Kamat, P.; Gite, S.; Chandekar, H.; Dlima, L.; Pradhan, B. Multi-class fruit ripeness detection using YOLO and SSD object detection models. *Discov. Appl. Sci.* **2025**, *7*, 931. [CrossRef]
21. Pandey, A.; Kumar, S.; Verma, A.; Baijal, S.; Dutta, C.; Choudhury, T.; Patni, J.C. Enhancing Fruit Recognition with YOLO v7: A Comparative Analysis Against YOLO v4. In Proceedings of the International Conference on Information Systems and Management Science, St. Julian, Malta, 18–19 December 2023; pp. 330–342.
22. Diwan, T.; Anirudh, G.; Tembhurne, J.V. Object detection using YOLO: Challenges, architectural successors, datasets and applications. *Multimed. Tools Appl.* **2023**, *82*, 9243–9275. [CrossRef]
23. Liang, Q.; Zhu, W.; Long, J.; Wang, Y.; Sun, W.; Wu, W. A real-time detection framework for on-tree mango based on SSD network. In Proceedings of the International Conference on Intelligent Robotics and Applications, Newcastle, NSW, Australia, 9–11 August 2018; pp. 423–436.
24. 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. [CrossRef]

25. Sun, Q.; Li, P.; He, C.; Song, Q.; Chen, J.; Kong, X.; Luo, Z. A lightweight and high-precision passion fruit YOLO detection model for deployment in embedded devices. *Sensors* **2024**, *24*, 4942. [CrossRef]
26. Wu, H.; Mo, X.; Wen, S.; Wu, K.; Ye, Y.; Wang, Y.; Zhang, Y. DNE-YOLO: A method for apple fruit detection in Diverse Natural Environments. *J. King Saud Univ.-Comput. Inf. Sci.* **2024**, *36*, 102220. [CrossRef]
27. Wang, L.; Wang, S.; Wang, B.; Yang, Z.; Zhang, Y. Jujube-YOLO: A precise jujube fruit recognition model in unstructured environments. *Expert Syst. Appl.* **2025**, *291*, 128530. [CrossRef]
28. Nan, Y.; Zhang, H.; Zeng, Y.; Zheng, J.; Ge, Y. Intelligent detection of Multi-Class pitaya fruits in target picking row based on WGB-YOLO network. *Comput. Electron. Agric.* **2023**, *208*, 107780. [CrossRef]
29. Wang, D.; He, D. Channel pruned YOLO V5s-based deep learning approach for rapid and accurate apple fruitlet detection before fruit thinning. *Biosyst. Eng.* **2021**, *210*, 271–281. [CrossRef]
30. Shen, S.; Duan, F.; Tian, Z.; Han, C. A novel deep learning method for detecting strawberry fruit. *Appl. Sci.* **2024**, *14*, 4213. [CrossRef]
31. Du, X.; Cheng, H.; Ma, Z.; Lu, W.; Wang, M.; Meng, Z.; Jiang, C.; Hong, F. DSW-YOLO: A detection method for ground-planted strawberry fruits under different occlusion levels. *Comput. Electron. Agric.* **2023**, *214*, 108304. [CrossRef]
32. Wang, Y.; Yan, G.; Meng, Q.; Yao, T.; Han, J.; Zhang, B. DSE-YOLO: Detail semantics enhancement YOLO for multi-stage strawberry detection. *Comput. Electron. Agric.* **2022**, *198*, 107057. [CrossRef]
33. Wang, C.; Wang, H.; Han, Q.; Zhang, Z.; Kong, D.; Zou, X. Strawberry detection and ripeness classification using yolov8+ model and image processing method. *Agriculture* **2024**, *14*, 751. [CrossRef]
34. Liu, J.; Guo, J.; Zhang, S. YOLOv11-HRS: An Improved Model for Strawberry Ripeness Detection. *Agronomy* **2025**, *15*, 1026. [CrossRef]
35. Ye, R.; Shao, G.; Gao, Q.; Zhang, H.; Li, T. CR-YOLOv9: Improved YOLOv9 multi-stage strawberry fruit maturity detection application integrated with CRNET. *Foods* **2024**, *13*, 2571. [CrossRef]
36. Sorour, S.E.; Alsayyari, M.; Alqahtani, N.; Aldosery, K.; Altaweel, A.; Alzhrani, S. An Intelligent Management System and Advanced Analytics for Boosting Date Production. *Sustainability* **2025**, *17*, 5636. [CrossRef]
37. Lammers, K.; Zhang, K.; Zhu, K.; Chu, P.; Li, Z.; Lu, R. Development and evaluation of a dual-arm robotic apple harvesting system. *Comput. Electron. Agric.* **2024**, *227*, 109586. [CrossRef]
38. Karim, M.R.; Ahmed, S.; Reza, M.N.; Lee, K.-H.; Jin, H.; Ali, M.; Chung, S.-O.; Sung, J. A review on stereo vision for feature characterization of upland crops and orchard fruit trees. *Precis. Agric. Sci. Technol.* **2024**, *6*, 104–122.
39. Rayamajhi, A.; Jahanifar, H.; Asif, M.; Mahmud, M.S. Measuring Ornamental 3D Canopy Volume and Trunk Diameter Using Stereo Vision for Precision Spraying and Assessing Tree Maturity. In Proceedings of the 2024 ASABE Annual International Meeting, Anaheim, CA, USA, 28–31 July 2024; p. 1.
40. Thakur, M.; Belwal, T. *Advances in Postharvest and Analytical Technology of Horticulture Crops*; Springer: Berlin/Heidelberg, Germany, 2024.
41. Steward, B.L.; Tekeste, M.Z.; Gai, J.; Tang, L. Modeling, Simulation, and Visualization of Agricultural and Field Robotic Systems. In *Fundamentals of Agricultural and Field Robotics*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 297–334.
42. Li, Y.; Wang, W.; Guo, X.; Wang, X.; Liu, Y.; Wang, D. Recognition and positioning of strawberries based on improved YOLOv7 and RGB-D sensing. *Agriculture* **2024**, *14*, 624. [CrossRef]
43. Ge, Y.; Xiong, Y.; From, P.J. Three-dimensional location methods for the vision system of strawberry-harvesting robots: Development and comparison. *Prec. Agric.* **2023**, *24*, 764–782. [CrossRef]
44. Whitaker, V.M.; Boyd, N.S.; Peres, N.; Desaeger, J.; Lahiri, S.; Agehara, S. *Chapter 16. Strawberry Production*; HS736; University of Florida, IFAS Extension: Gainesville, FL, USA, 2023; Available online: https://edis.ifas.ufl.edu/publication/CV134 (accessed on 17 October 2025).
45. He, L.-h.; Zhou, Y.-z.; Liu, L.; Cao, W.; Ma, J.-H. Research on object detection and recognition in remote sensing images based on YOLOv11. *Sci. Rep.* **2025**, *15*, 14032. [CrossRef] [PubMed]
46. Shi, J.; Ruan, S.; Tao, Y.; Rui, Y.; Deng, J.; Liao, P.; Mei, P. Improved YOLO algorithm based on multi-scale object detection in haze weather scenarios. *CHAIN* **2025**, *2*, 183–197. [CrossRef]
47. Fan, Q.; Huang, H.; Chen, M.; Liu, H.; He, R. Rmt: Retentive networks meet vision transformers. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–22 June 2024; pp. 5641–5651.
48. Chen, Y.; Zhang, C.; Chen, B.; Huang, Y.; Sun, Y.; Wang, C.; Fu, X.; Dai, Y.; Qin, F.; Peng, Y. Accurate leukocyte detection based on deformable-DETR and multi-level feature fusion for aiding diagnosis of blood diseases. *Comput. Biol. Med.* **2024**, *170*, 107917. [CrossRef]
49. Liu, W.; Lu, H.; Fu, H.; Cao, Z. Learning to upsample by learning to sample. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Paris, France, 2–6 October 2023; pp. 6027–6037.
50. Yan, Y.; Yang, P.; Yan, L.; Wan, J.; Sun, Y.; Tansey, K.; Asundi, A.; Zhao, H. Automatic checkerboard detection for camera calibration using self-correlation. *J. Electron. Imaging* **2018**, *27*, 033014. [CrossRef]

51. Juarez-Salazar, R.; Zheng, J.; Diaz-Ramirez, V.H. Distorted pinhole camera modeling and calibration. *Appl. Opt.* **2020**, *59*, 11310–11318. [CrossRef] [PubMed]
52. Culjak, I.; Abram, D.; Pribanic, T.; Dzapo, H.; Cifrek, M. A brief introduction to OpenCV. In Proceedings of the 2012 Proceedings of the 35th International Convention MIPRO, Opatija, Croatia, 21–25 May 2012; pp. 1725–1730.
53. Gilat, A. *MATLAB: An Introduction with Applications*; John Wiley & Sons: Hoboken, NJ, USA, 2017.