agronomy

MDPI

Article

Strawberry Maturity Recognition Based on Improved YOLOv5

Zhiqing Tao [1], Ke Li [1], Yuan Rao [1], Wei Li [2] and Jun Zhu [1,*]

[1] School of Information and Computer Science, Anhui Agricultural University, Hefei 230036, China; taozhiqing@stu.ahau.edu.cn (Z.T.); like@ahau.edu.cn (K.L.); raoyuan@ahau.edu.cn (Y.R.)
[2] Hefei Institutes of Physical Science, Chinese Academy of Sciences, Hefei 230031, China; liwei@iim.ac.cn
* Correspondence: zhujun@ahau.edu.cn

Abstract: Strawberry maturity detection plays an essential role in modern strawberry yield estimation and robot-assisted picking and sorting. Due to the small size and complex growth environment of strawberries, there are still problems with existing recognition systems' accuracy and maturity classifications. This article proposes a strawberry maturity recognition algorithm based on an improved YOLOv5s model named YOLOv5s-BiCE. This algorithm model is a replacement of the upsampling algorithm with a CARAFE module structure. It is an improvement on the previous model in terms of its content-aware processing; it also widens the field of vision and maintains a high level of efficiency, resulting in improved object detection capabilities. This article also introduces a double attention mechanism named Biformed for small-target detection, optimizing computing allocation, and enhancing content perception flexibility. Via multi-scale feature fusion, we utilized double attention mechanisms to reduce the number of redundant computations. Additionally, the Focal_EIOU optimization method was introduced to improve its accuracy and address issues related to uneven sample classification in the loss function. The YOLOv5s-BiCE algorithm was better at recognizing strawberry maturity compared to the original YOLOv5s model. It achieved a 2.8% increase in the mean average precision and a 7.4% increase in accuracy for the strawberry maturity dataset. The improved algorithm outperformed other networks, like YOLOv4-tiny, YOLOv4-lite-e, YOLOv4-lite-s, YOLOv7, and Fast RCNN, with recognition accuracy improvements of 3.3%, 4.7%, 4.2%, 1.5%, and 2.2%, respectively. In addition, we developed a corresponding detection app and combined the algorithm with DeepSort to apply it to patrol robots. It was found that the detection algorithm exhibits a fast real-time detection speed, can support intelligent estimations of strawberry yield, and can assist picking robots.

Keywords: strawberry; maturity detection; Biformed; DeepSort; YOLOv5

Citation: Tao, Z.; Li, K.; Rao, Y.; Li, W.; Zhu, J. Strawberry Maturity Recognition Based on Improved YOLOv5. *Agronomy* **2024**, *14*, 460. https://doi.org/10.3390/agronomy14030460

Academic Editor: Roberto Marani

Received: 4 January 2024
Revised: 23 February 2024
Accepted: 24 February 2024
Published: 26 February 2024

1. Introduction

Strawberry, a perennial herb belonging to the Rosaceae berry family, is the second-most cultivated and produced berry fruit worldwide. With a high nutritional value, strawberries are popular in daily life. However, mature strawberries are thin-skinned and easily damaged, making storage and sales difficult. Therefore, it is necessary to classify strawberries according to maturity for the purposes of selling or storing them in different ways [1,2]. Fully mature strawberries cannot be preserved and must be eaten immediately after picking. Strawberries in the medium well can be transported over short distances, and strawberries in the medium can be transported over long distances. [3]. With modern agriculture's large-scale production, machinery can replace manpower to save time and costs while improving efficiency. Artificial sensory recognition is still the main method used for picking and sorting, but it has many requirements and high costs. The maturity recognition of strawberries is an essential step towards integrating intelligent picking and sorting methods for modern strawberry production [4]. This recognition process plays a crucial role in achieving the intelligent production of strawberries via efficient sorting and picking methods at reduced costs.

With the sustained advancement of deep learning and its integration across various domains, the agricultural industry is also gradually embracing the use of deep learning technology to address common challenges [5–7]. One noteworthy research area within this domain is object detection. Target detection algorithms can be broadly divided into two categories—the first being the two-step algorithm, which initially identifies a set of candidate regions, followed by their subsequent classification. Common algorithms in this category include Faster R-CNN [8] and various algorithms that use a CNN as the backbone network [9,10]. In [11], the leaves of sweet potato plants were identified based on an improved Faster R–CNN model. Additionally, [12] identified and judged apple blossoms and flowers based on the Mask R–CNN detection model. The second type of object detection is end-to-end, utilizing a single network to process input images step-by-step, outputting the presence of objects and their respective locations. Common algorithms employed in this methodology include YOLO [13] and SSD [14]. In [15,16], an improved YOLOv5 model was used to identify the maturity of four types of tomatoes in a greenhouse; [17] combined the dark channel algorithm and YOLOv5 to achieve the maturity recognition of strawberries with an accuracy rate of 85%; [18] combined YOLOv5 with an attention mechanism to detect targets (apples); and [19] implemented an improved SSD algorithm for the detection of jujube maturity.

From the above literature, it can be seen that, compared with one-step object detection algorithms, two-step object detection methods have disadvantages such as their poor real-time performance and difficulty in small-object detection. In addition, there are still issues in the recognition and detection of strawberry maturity, such as limited classification systems of strawberry maturity, the failure to distinguish between mature strawberries and rotten fruits, and the low level of maturity detection. In order to improve the recognition speed of strawberries, ensure the accuracy of strawberry maturity classification and recognition systems, and meet requirements for strawberry maturity detection, after taking into account various factors such as training time, accuracy, ease of use, community support, and deployment requirements, we selected YOLOv5s from the various one-step detection algorithms as a base model and made improvements to it. The algorithm effectively addresses issues of low accuracy, imperfect maturity classifications, and the lack of universality. It incorporates a CARAFE module structure to enhance content-aware processing, expand the field of view, and maintain model efficiency. Additionally, it improves small-object detection through a dual-attention mechanism and enhances the level of accuracy with Focal_EIOU optimization. Additionally, in multi-scale feature fusion, BiFPN is introduced to reduce the number of redundant computations. The experimental results demonstrate the algorithm's superiority over existing solutions.

2. Materials and Methods

2.1. Data Acquisition

The dataset for this study was taken from the Smart Agriculture Valley Strawberry Demonstration Park in Changfeng County, Hefei City, Anhui Province, where over 300 types of strawberries were planted. Multiple categories of strawberries were selected from the dataset, such as "hongyan", "tianxianzui", "yuexiu", etc., and photographed in the demonstration park to eliminate data bias caused by different varieties in the detection results. Images were taken under different environmental conditions, including sunny weather and cloudy weather, and the images captured were of single fruits. In total, 1446 high-resolution RGB images were acquired, which were then saved as 640 × 480 pixel images. After that, each image data point was enhanced once by brightening it, increasing contrast, rotating it, flipping it, adding noise, and other methods; finally, 2892 images were obtained. After segmentation according to the ratio of 7:2:1, 2024 training sets, 578 verification sets, and 290 test sets are obtained. In this article, the maturity levels of strawberries were divided into five categories based on their surface color, namely: immature, medium, medium well, mature. Strawberries with irregular shapes and sizes were classified as malformed fruits, which correspond to 1–5 in Figure 1 below. The

labeling tool used in this paper was Labellmg, which manually labeled different types of strawberries and finally saved the labeled data in TXT format. The final strawberry maturity dataset contains approximately 16,000 annotated bounding boxes, with an average of 4–7 bounding boxes per image, and the image with the most bounding boxes has 15 annotations.

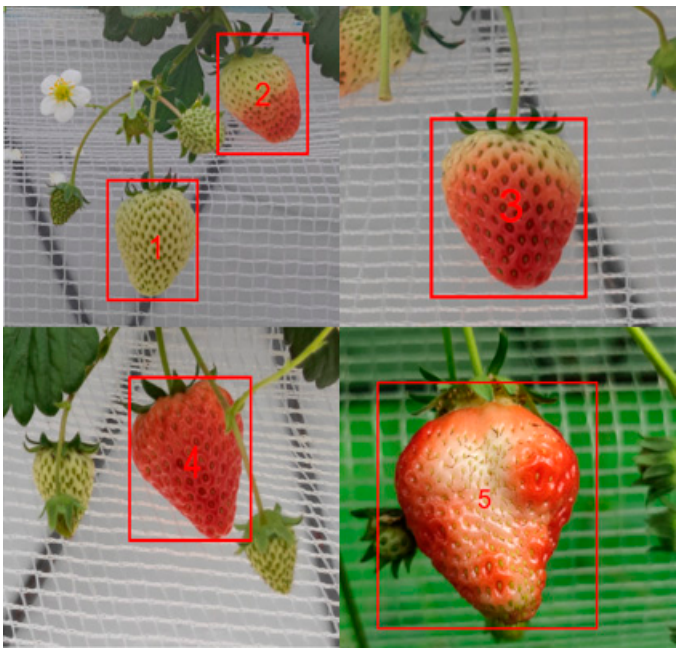

Figure 1. Strawberries at different stages of maturity.

2.2. YOLOv5s Network Structure

When selecting our model, we considered various factors such as training time, accuracy, ease of use, community support, and deployment requirements. Based on the experimental environment in this paper, YOLOv5 had lower computational resource consumption and faster processing speed compared to those of other models. It performed well in real-time detection and lightweight deployment scenarios. This was particularly advantageous when running on devices with limited resources. Therefore, we chose YOLOv5s as our base model for the experiments.

In addition to YOLOv5s, the YOLOv5 series boasts three other versions: YOLOv5m, YOLOv5l, and YOLOv5x. Among these versions, YOLOv5m and YOLOv5l have greater network depth and breadth and achieve higher accuracy than YOLOv5s; however, they also require longer training times and larger model scales. Meanwhile, YOLOv5x is the deepest and widest version, with the best performance in terms of accuracy; however, it necessitates even longer training times and larger memory requirements. Therefore, when selecting an appropriate model for a project, one's specific circumstances must be carefully considered.

For the strawberry maturity recognition task in this article, we chose YOLOv5s as the recognition network, which has the shallowest network depth and width, because the strawberry maturity recognition network needed to meet the requirements for real-time recognition. If the network depth and width were too high, it would lead to a long detection time. Figure 2 displays the structure of this network, which includes four segments: the input module, backbone module, neck module, and prediction module. The input module processes input data before sending them to the backbone module. The backbone module represents the core part of the entire network, as it extracts image features. Among them, the FOCUS block represents the convolutional neural network used for feature extraction, while the CBS block is used to reduce network parameters and improve the efficiency of feature extraction. The neck module connects it with the prediction module while also

playing a role in feature fusion, and Nearest represents the nearest-neighbor interpolation method for upsampling. Finally, the prediction module outputs target detection results with specific compositions, as illustrated in Figure 3.

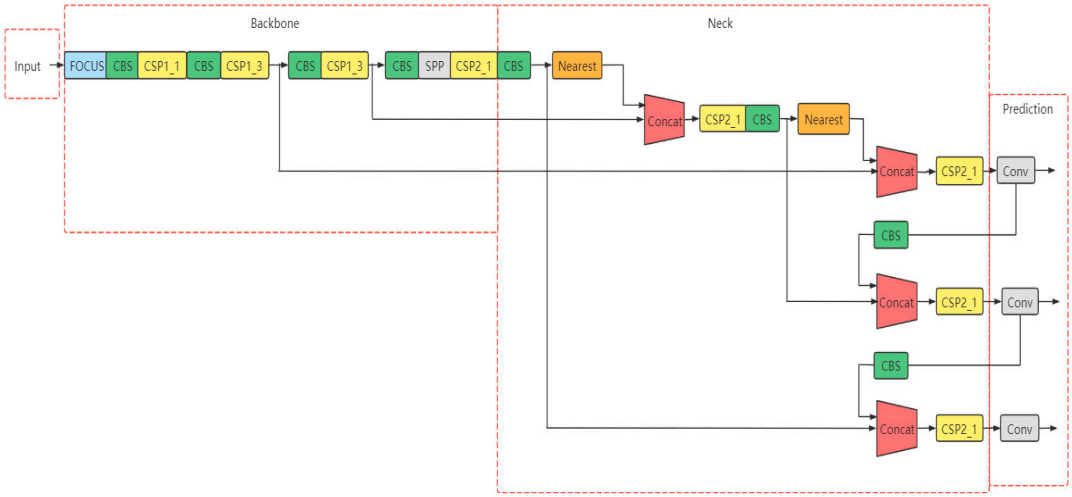

Figure 2. The architecture of the network of YOLOv5s in this paper.

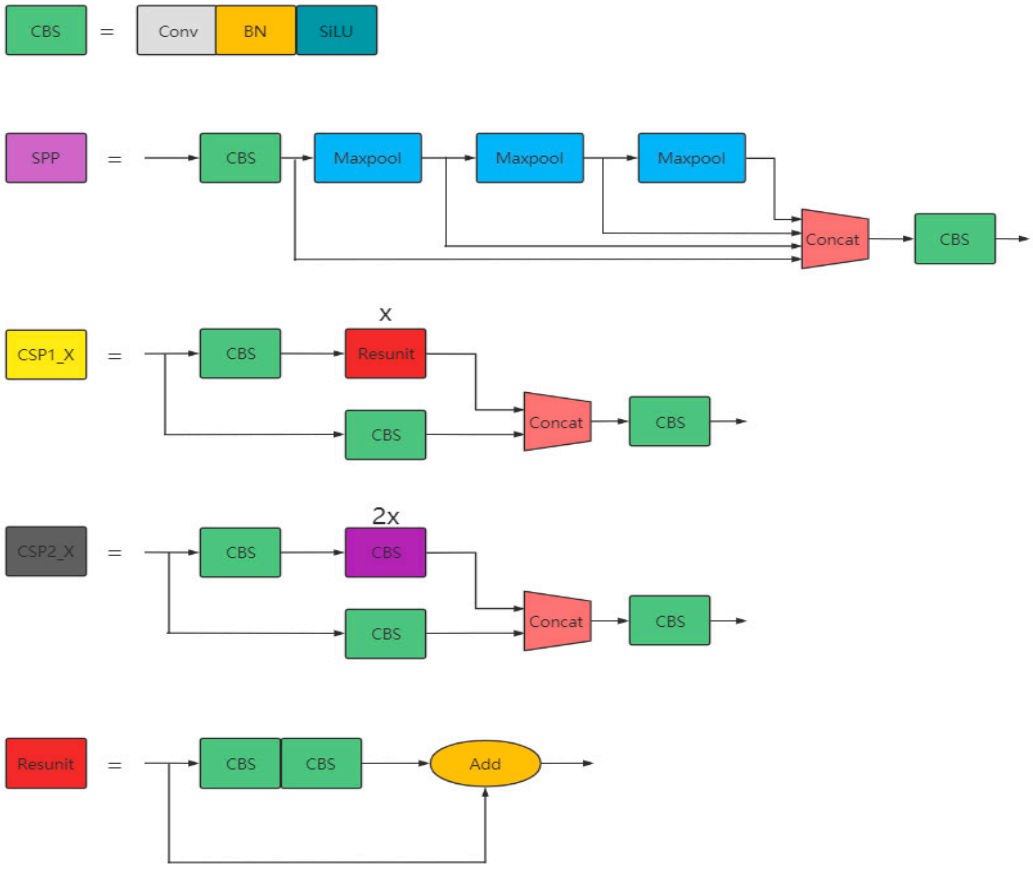

Figure 3. Composition of modules in YOLOv5s.

Parameter Optimization of Anchor Frame

In YOLOv5, the initial anchor box consists of nine boxes obtained from a K-means clustering algorithm based on the COCO dataset. However, since a new dataset was used in this study, customizing the size of the anchor boxes can result in better target detection. To achieve this, the Kmeans++ clustering algorithm was utilized to redesign the anchor

box sizes. Unlike the traditional K-means clustering algorithm, Kmeans++ optimizes the selection of initial clustering centers, which significantly enhances target detection and classification results. This approach results in an improved clustering effect and suitable anchor boxes for small-target datasets that enhance small-target detection accuracy.

To customize the anchor box sizes for strawberries, this study selected three anchor boxes for the large, medium, and small scales, resulting in a total of nine sets of data. The number of clusters K was set to 9, and after 1000 iterations, a new prior anchor box scale was obtained, as shown in Table 1, and normalized.

Table 1. New anchor box scale.

Feature Scale	Anchor Box1/px	Anchor Box2/px	Anchor Box3/px
Small scale	40 × 40	60 × 51	53 × 71
Medium scale	66 × 88	92 × 71	80 × 108
Large scale	95 × 127	134 × 108	115 × 155

Based on the table presented above, it becomes apparent that recalculating the corresponding anchor frame size according to actual data is vital in order to better fulfill the demands of target detection tasks. Through analyzing the size and proportion of strawberry targets in the training set, it was found that they were primarily distributed between 40 × 44 and 115 × 155 pixels, due to a certain size variation. Thus, during the training process, this study employed the newly calculated anchor frame to replace the original settings and further trained the model to reduce redundant information and improve the accuracy of target detection.

2.3. Improved YOLOv5s Network

2.3.1. Biformed Attention Mechanism

The growth environment surrounding the strawberries plays a crucial role in determining the performance of the strawberry maturity detection model. To address this issue, researchers worldwide have been focusing on using attention mechanisms to improve models' performance.

The SE (Squeeze-and-Excitation) attention mechanism includes a compression operation that aggregates feature maps into channel descriptors and an incentive operation that learns a set of weight vectors to selectively emphasize meaningful channels [20]. This enhances mapping channels that are useful for the current task and suppresses those that are not. The CBAM (Convolutional Block Attention Module) attention mechanism improves upon the SE model by integrating channel and spatial attention [21]. It comprises two modules, CAM (Channel Attention Module) and SAM (Spatial Attention Module). The former selectively emphasizes critical channels by learning the weightage of each channel, whereas the latter improves the visibility of local features by determining the spatial significance of the feature map. The CA (Coordinate Attention) attention mechanism selectively emphasizes meaningful channels by weighting their importance. It employs two fully connected layers to calculate the importance weight for each channel and applies it to the feature map, thereby enhancing the model's recognition of crucial features [22].

Overall, these attention mechanisms have shown great potential in improving strawberry maturity detection models under varying growth environments.

However, the use of attention mechanisms commonly employed in neural networks suffers from certain drawbacks, such as poor interpretability and inability to capture long-term dependencies. Additionally, stacking layers continuously is necessary to obtain a larger receptive field, but this approach fails to capture global information effectively. To address these issues and enhance our model's target detection capabilities, this article introduces the BiFormer attention mechanism [23].

While traditional transformers can easily capture global information, they suffer from computational explosion and slow training. In contrast, BiFormer is a dynamic, sparse-

attention, double-layer routing method that focuses on a small number of related tags in an adaptive manner without distracting attention from other unrelated tags. This results in improved performance and high computational efficiency. Figure 4 illustrates the complete structure of BiFormer, while Figure 5 shows the details of the BiFormer block.

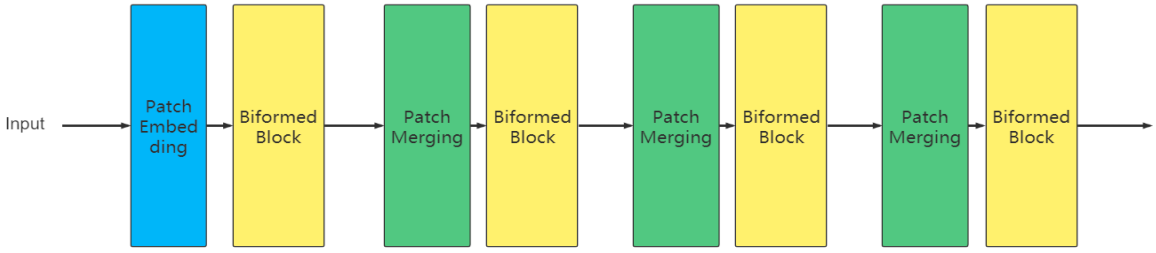

Figure 4. BiFormer structure.

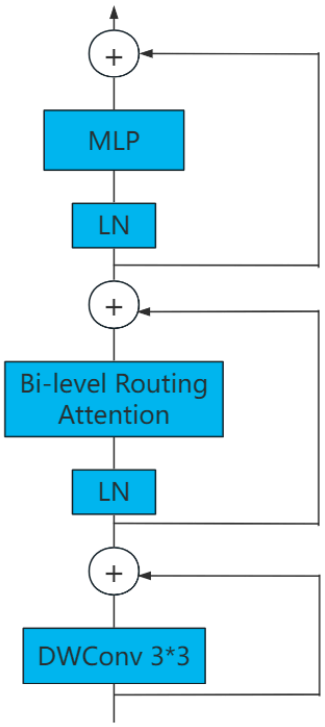

Figure 5. BiFormer block.

2.3.2. Improvement of Upsampling Algorithm

Feature upsampling algorithms play a pivotal role in determining the quality of object detection results. Various upsampling techniques can have differing impacts on object detection outcomes. In YOLOv5, nearest-neighbor interpolation is utilized for upsampling, which requires negligible amount of computational resources. This method compares the target point with known data points and assigns a value to the closest data point to the target point. While this approach is fast and efficient due to its lack of calculation requirements, it only considers the closest data point without utilizing feature semantic information, resulting in a small perception domain of only 1 pixel $\times$ 1 pixel.

To improve feature upsampling for detecting targets, this paper introduces a lightweight module, CARAFE, to address these issues [24]. CARAFE is capable of effectively utilizing surrounding information and has a larger perception domain than nearest-neighbor interpolation. It processes perception based on input content and dynamically generates an adaptive kernel while remaining lightweight, with low computational overhead and quick computing speed. CARAFE comprises two modules—one for predicting upsampling kernels and another for reorganizing features—as depicted in Figure 6.

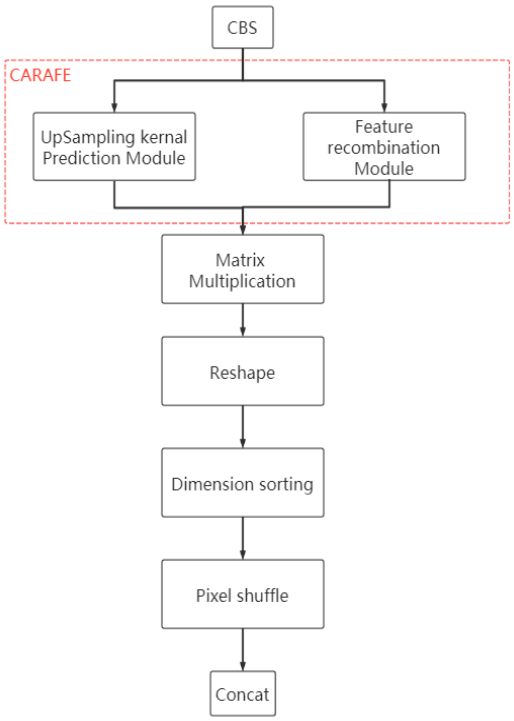

Figure 6. New network structure.

The upsampling kernel prediction module comprises multiple sub-modules, such as feature mapping, channel compression, content coding, upsampling kernel prediction, and upsampling kernel normalization. The content encoder is responsible for encoding the feature map, which first undergoes channel compression to generate a new recombined kernel. Subsequently, the kernel normalizer utilizes the softmax function to normalize each of the recombined kernels. The feature recombination module is accountable for mapping each position's information in the output feature map to the corresponding input feature map. It extracts an upsampling kernel size region centered on that position and calculates the dot product between the region and point prediction upsampling kernel to produce an output value. Since it concentrates more on local related points' information, the reconstructed feature map has stronger semantic significance than the original one.

In summary, CARAFE's architecture comprises two modules that work together synergistically—one for predicting upsampling kernels and another for reorganizing features. This approach enables the more effective utilization of surrounding information, with a larger perception domain than nearest-neighbor interpolation while remaining lightweight, with low computational overhead and quick computing speed.

2.3.3. BIFPN Feature Fusion Network

The FPN structure in the YOLOv5 algorithm is a top-down, one-way information flow that retains more features. YOLOv5s implements the PANet network, which includes a bottom-up pathway for transferring essential image details to the feature layer used for prediction. Due to the dual pathways, PANet can simultaneously incorporate semantic information from the top and feature information from the bottom [25]. However, this design also results in a high degree of complexity, which reduces the efficiency of information transmission in the PANet model.

To address this issue and improve feature integration, this paper proposes using an optimized BIFPN based on the FPN mechanism, as shown in Figure 7, that reconstructs the top-down and bottom-up routes using bidirectional fusion [26]. This approach fuses feature information of different scales while unifying the feature resolution scale through upsampling and downsampling. The BIFPN establishes bidirectional connections between feature maps of the same scale while deleting nodes without feature fusion and with small

contributions, adding new channels between original input and output nodes to integrate more feature information while lowering resource consumption. Among them, P3. . .P7 represent the original node, and the arrow represents the output direction.

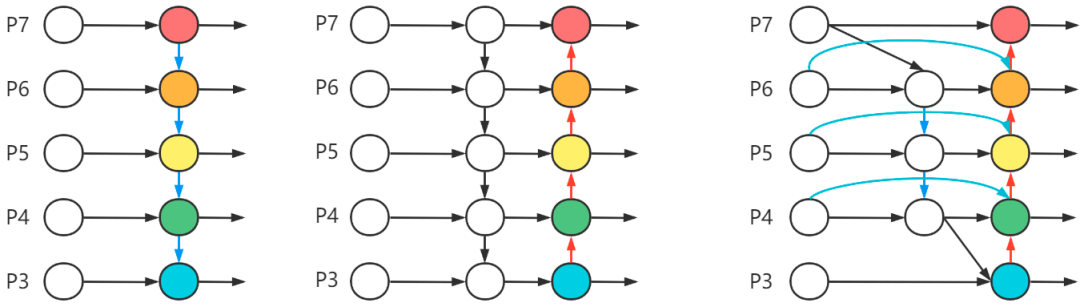

Figure 7. FPN PANet BIFPN structure.

The BIFPN structure performs fast normalized fusion by calculating weight ratios then normalizing them to [0, 1] to enhance target perception ability in different situations. At the end of the prediction, it can fuse the information of feature maps between different levels, effectively addressing interference caused by factors such as image noise.

2.3.4. Improvement of Loss Function

The loss function is a critical component of the training model, as it assesses the precision of the prediction results and gauges the disparity between the actual data and the model. Selecting an appropriate loss function can accelerate convergence and enhance the quality of the training model. YOLOv5 adopts the *CIOU* function as its network loss function, given by the following:

$$L_{CIOU} = 1 - IOU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v \tag{1}$$

The *IOU* ratio measures the degree of overlap between the predicted detection frame and the actual detection frame. Here, α represents the trade-off factor, and v is used to evaluate the uniformity of the aspect ratio. b and b^{gt} denote the center points of the predicted frame and actual frame, respectively. Additionally, ρ denotes the Euclidean distance between these two center points, while c signifies the diagonal distance of the smallest closed area that can contain both frames. Nevertheless, *CIOU* solely takes into account the aspect ratio of the predicted frame, neglecting the actual differences in width and height, their confidence, and sample balance. As a result, it may occasionally impede the effective optimization of the model's similarity. This study utilized the *Focal_EIOU* loss function, which is based on *CIOU*. *Focal_EIOU* optimization is an innovative combination of the focal loss and the *EIOU* (enhanced intersection over union) loss. The significance of *Focal_EIOU* lies in its ability to balance the learning focus between easy and hard examples, with a particular emphasis on misclassified objects. By dynamically adjusting the loss contribution of each example, researchers can drive the model to correct its mistakes more effectively, thereby enhancing its accuracy, especially in complex scenes with numerous overlapping objects or varying scales [27]. This novel approach is utilized to address the issue of unbalanced sample distribution by introducing focal loss. By introducing this technique, researchers are able to better focus on high-quality anchor frames and achieve more accurate results. The formula for this loss function is as follows:

$$L_{EIOU} = L_{IOU} + L_{dis} + L_{asp} = 1 - IOU + \frac{\rho(b, b^{gt})}{c^2} + \frac{\rho(w, w^{gt})}{C_w^2} + \frac{\rho(h, h^{gt})}{C_h^2} \tag{2}$$

$$L_{Focal-EIOU} = IOU^{\gamma} L_{EIOU} \tag{3}$$

Within this formula, L_{IOU}, L_{dis}, and L_{asp}, respectively, correspond to the overlap loss, center distance loss, and aspect ratio loss. Specifically, the *IOU* loss evaluates the overlap between the predicted and actual frames, while the center distance loss assesses the distance between their centers. Additionally, the aspect ratio loss quantifies the variation in aspect ratio between the predicted and actual frames. The parameters b, w, and h correspond to the center point, width, and height of the predicted frame. Similarly, b^{gt}, w^{gt}, and h^{gt} denote the center point, width, and height of the true bounding box. Furthermore, c represents the diagonal length of the smallest bounding rectangle that encompasses both the predicted frame and the real frame. Additionally, C_w and C_h signify its width and height. These parameters are used to compute a single bounding box's relationship with respect to its true counterpart. Finally, γ is a hyperparameter controlling the curves' curvature to adjust contributions of different parts towards the total loss value.

2.3.5. Target Tracking Algorithm

To obtain real-time information on strawberry maturity recognition counts from the inspection robot, it is necessary to track and locate the strawberries. This study utilizes the DeepSort detection-based, multi-target tracking algorithm to achieve this goal. DeepSort is an advanced version of the Sort tracking algorithm designed for multi-target tracking. The algorithm employs a Kalman filter to forecast the positions of the targets in subsequent frames, followed by cascade matching. Finally, the Hungarian algorithm is used for data association to improve tracking accuracy. The overall process is illustrated in Figure 8. The video was first processed by a detection network to obtain the strawberries' positions, which were then fed into the tracking network for data association and matching between frames to produce accurate tracking results.

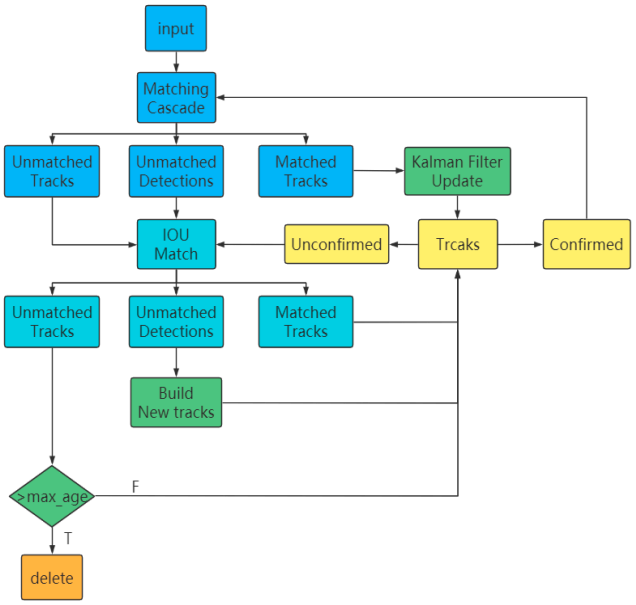

Figure 8. DeepSort flow chart.

The DeepSort algorithm utilizes motion and appearance information of the targets to compute the degree of similarity. To evaluate the correlation between predicted and detected targets in terms of motion information, Mahalanobis distance was employed. The calculation for Mahalanobis distance is provided below:

$$d^{(1)}(i,j) = (d_j - y_i)^T S_i^{-1} (d_j - y_i) \quad (4)$$

This formula proposes a methodology by which we can evaluate the level of agreement between trajectory and prediction frames using the notation $d^{(1)}$. In order to evaluate the motion similarity, a Kalman filter was used to predict the state and detection value of

the target from the detection and tracking frames. Moreover, the Mahalanobis distance was computed to estimate the matching degree. To represent the *j*-th detection frame's position, d_j was used. While y_i represents the position of the *i*-th target prediction frame, S_i *is* also considered to represent covariance between the detection and prediction positions of ith target, which check the measurement stability by determining the standard deviation between both positions. In scenarios in which a target remains hidden for an extended period of time or there are viewing angle inconsistencies, appearance information should be incorporated and cosine distance should be used to address identity conversion issues caused by occlusion. The formula for cosine distance is as follows:

$$d^{(2)}(i,j) = min\left\{1 - r_j^T r_k^{(i)} \middle| r_k^{(i)} \in R_i\right\} \tag{5}$$

In the formula, $d^{(2)}$ represents the cosine distance between two targets and r_j represents the feature vector extracted by each target d_j, $||r_j|| = 1$. In addition, the vector library $R_k = \{r_k^{(i)}\}^{Lk}_{k=1}$ retains the eigenvector of each track k in the nearest L_k frame. The eigenvectors exceeding L_k are not considered, and their contribution to the results decreases with the increase in L_k. When the value of $d^{(2)}$ is lower than the specified threshold, the association can be considered successful.

The ultimate comprehensive matching degree $c_{i,j}$ is weighted by two kinds of information:

$$c_{i,j} = \lambda d^{(1)}(i,j) + (1-\lambda) d^{(2)}(i,j) \tag{6}$$

In the formula, λ is a superparameter used to adjust the influence of two measurement methods on the correlation, and the target correlation is considered if and only if the measurement values $c_{i,j}$ are between $d^{(1)}$ and $d^{(2)}$.

3. Results

3.1. Training of Models

The network training for this study was conducted using the PyTorch framework on an Ubuntu system. The server configuration included an i5-9500 CPU and GeForce RTX 2080ti GPU, with Cuda11.6, cuDNN8.0.5, and Python 3.9 installed.

In this study, all algorithms were trained under the same environmental conditions and hyper-parameter settings. The model underwent training with a batch size of eight, completed 300 iterations, and was fed image inputs that were 640 × 640 pixels. Furthermore, an initial learning rate of 0.01 was set, and a confidence threshold of 0.5 was employed to differentiate between positive and negative samples.

3.2. Model Evaluation

This study employed standard evaluation metrics to evaluate the performance of the target detection model. The evaluation indices include accuracy *P*, recall *R*, mAP_0.5, and mAP_0.5:0.95. The accuracy is the percentage of correct predictions made by the model, while the recall rate represents the ratio of correct predictions to all actual targets. mAP provides an average accuracy measurement for each class. To calculate *mAP*, we first computed the *AP* of each category, which indicates how well the model detects different categories of interest.

$$P = \frac{TP}{TP + FP} \tag{7}$$

$$R = \frac{TP}{TP + FN} \tag{8}$$

$$AP = \int_0^1 P(R)dR \tag{9}$$

$$mAP = \frac{1}{C}\sum_{i=1}^{C} AP_i \tag{10}$$

In this study, we employed a formula that included the following variables: *TP*, *FN*, *FP*, and *TN*. These represent the number of positive classifications that were predicted accurately, the number of positive classifications that were mistakenly identified as negative, the number of negative classifications that were falsely identified as positive, and the number of negative classifications that were predicted correctly. In addition to this, we utilized *P*–*R* to represent how accurately a model performed with respect to recall *R*. This is commonly referred to as the *P*–*R* curve. Lastly, we considered *C* as the total number of classes taken into account for this study. Figure 9 illustrates the changes in the mean average precision during training with threshold values of 0.5 and 0.5:0.95, which were relatively high when using a threshold value of 0.5. Figure 10 shows an optimal training model based on the *P*–*R* curve for immature strawberries, strawberries at medium of maturity, strawberries at medium well of maturity, fully mature strawberries, and malformed strawberries with respective *AP* values of 0.969, 0.961, 0.948, 0.963, and 0.965. A further analysis revealed the average *AP* value for all five categories, represented by the blue curve in Figure 10, to be 0.961.

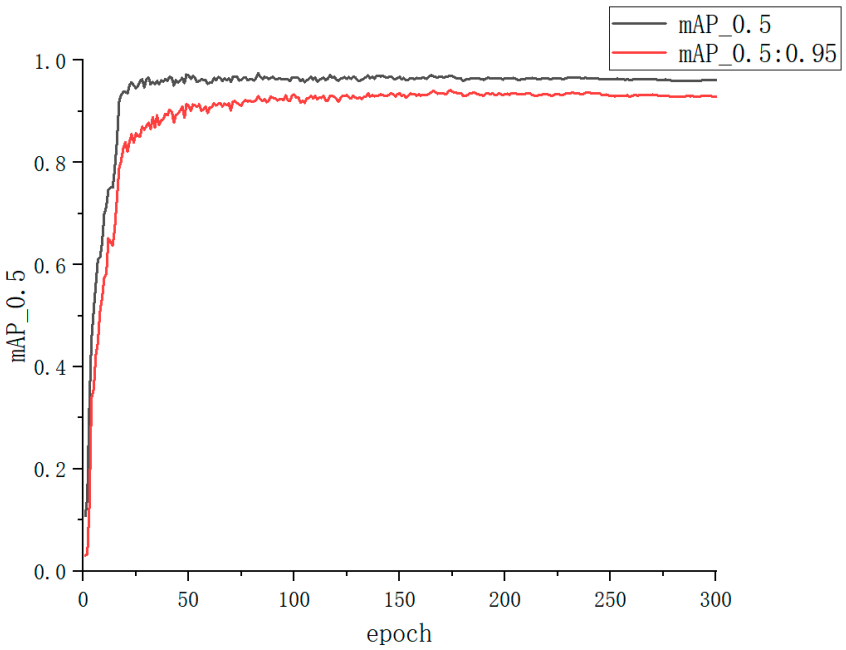

Figure 9. mAP curve.

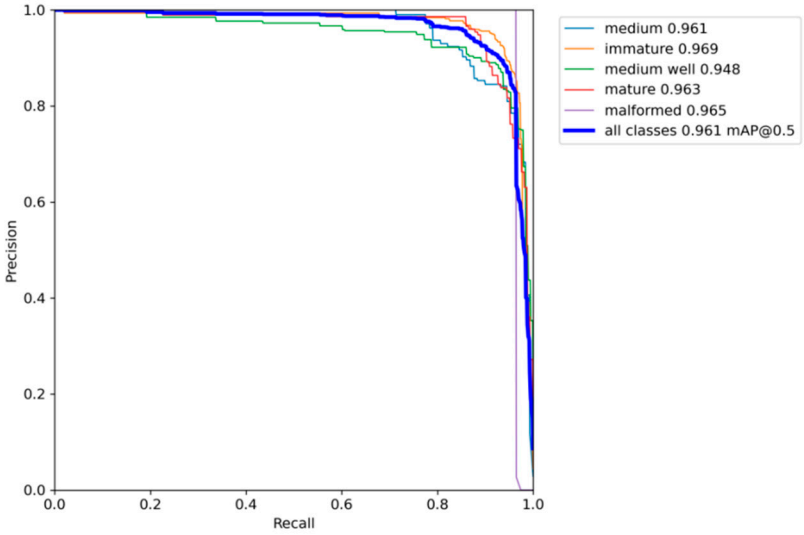

Figure 10. PR curve.

4. Discussion

4.1. Evaluation of the Model's Performance Pre- and Post-Improvement

We evaluated the performance of the trained model using a test set, which consisted of 475 immature strawberries, 140 strawberries in medium of maturity, 209 strawberries in medium well of maturity, 173 mature strawberries, and 84 malformed strawberries. The improved algorithm employed in this paper effectively discerned the quantity of strawberries in each classification, as demonstrated below: 453 immature strawberries, 129 strawberries in medium of maturity, 193 strawberries in medium well of maturity, 164 fully mature strawberries, and 83 malformed strawberries. The recognition effect before and after implementing the improved algorithm is depicted in Figure 11. The figure demonstrates that the YOLOv5s-BiCE model can effectively eliminate targets that were incorrectly identified by the YOLOv5s model. Furthermore, it is capable of detecting small, occluded, and overlapping targets with a higher level of accuracy.

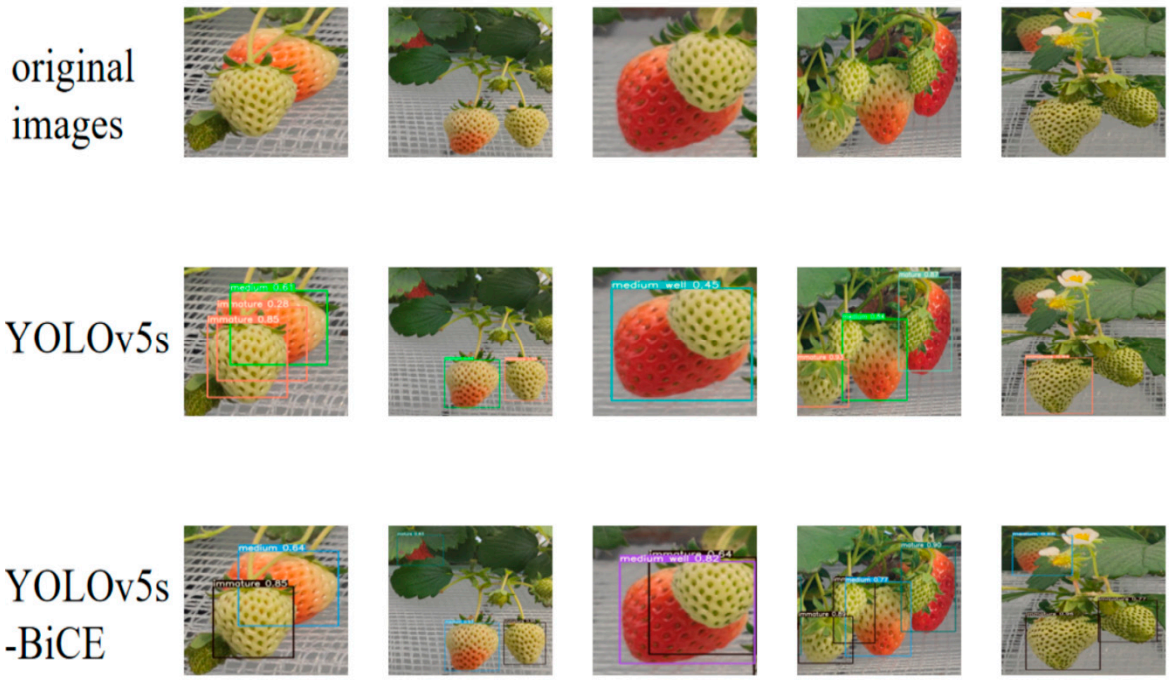

Figure 11. Evaluation of the model's performance pre- and post-improvement.

4.2. Comparison of Performance between This Algorithm and Several Target Detection Algorithms

4.2.1. Comparison between the Improved Algorithm and Other Algorithms

In this study, YOLOv5s-BiCE was compared to other models using a strawberry maturity data set. The evaluation models included a lightweight model and a common target detection model. Table 2 summarizes the results.

Table 2. Comparison of our model's performance to that of other models.

Algorithm	P/%	R/%	mAP_0.5/%	mAP_0.5:0.95/%	Size/MB
YOLOv4-tiny	91.2	88.2	91.5	79.5	6.7
YOLOv5-lite-e	89.8	81.4	87.5	71.4	1.6
YOLOv5-lite-s	90.3	85.7	90.3	77.5	3.2
YOLOv7	93	90.7	93.9	88.7	74.5
Faster RCNN	92.3	89.6	91.8	83.3	107.57
YOLOv5s-BiCE	94.5	93.4	96.1	92.9	15.3

Table 2 illustrates the impressive performance of the YOLO-BiCE model, achieving an mAP_0.5 of 96.4% and an mAP_0.5:0.95 ratio of 92.9%, all while maintaining a model size of

just 15.3 MB. Compared to other lightweight models, such as YOLOv4-tiny, YOLOv5-lite-e, and YOLOv5-lite-s, the YOLO-BiCE model boasts a significant increase in mAP_0.5 by 4.5%, 8.6%, and 5.8%, respectively, as well as an increase in mAP_0.5:0.95 by 13.4%, 21.5%, and 15.4%, correspondingly.

Moreover, compared to commonly used models such as the YOLOv7 and Faster RCNN models, the YOLOv5s-BiCE model still outperformed them, with increases in mAP_0.5 of 2.2% and 4.3%, respectively, while also achieving increases in mAP_0.5:0.95 of 4.2% and 9.6%, respectively.

Overall, these outcomes demonstrate that the YOLOv5s-BiCE model is highly effective compared to other models across various performance metrics, while maintaining a relatively smaller model size for easier deployment on resource-limited devices and platforms.

4.2.2. Comparison of Actual Recognition Effects of Test Sets

In this study, we evaluated the target detection performance of each algorithm by calculating the number of strawberries in each category identified by the algorithm and measuring the proportion of correctly identified strawberries to the total number of strawberries. To detect the test set, different algorithms were used with a threshold of 0.5, and the detection results for each algorithm are presented in Table 3.

Table 3. Comparison of actual effects with those of other models.

Algorithm	Number of Immature	Number of Medium	Number of Medium Well	Number of Mature	Number of Malformed	Recognition Accuracy Rate/%	Detection Time/s
YOLOv5s	426	164	262	158	74	89.7	2.6
YOLOv5s-B	435	153	189	178	79	92.4	2.7
YOLOv5s-Bi	440	151	187	176	92	92.7	2.6
YOLOv5s-C	445	123	185	169	78	92.5	2.6
YOLOv5s-E	430	125	178	188	90	90.6	2.6
YOLOv5s-BiCE	453	129	193	164	83	94.5	2.6
YOLOv4-tiny	434	127	183	183	86	91.5	1.17
YOLOv5-lite-e	428	124	184	160	88	90.3	3.01
YOLOv5-lite-s	431	125	184	179	74	90.8	3.27
YOLOv7	443	126	187	181	80	93.4	3.92
Faster RCNN	441	125	224	163	88	92.8	57.7

Table 3 demonstrates that the proposed algorithm exhibited a significant improvement in recognition accuracy, with an increase of 6.3% compared to its pre-improvement performance. It also outperformed the YOLOv5s-B, YOLOv5s-Bi, YOLOv5s-C, and YOLOv5s-E algorithms by 2.1%, 1.8%, 2%, and 3.9%, respectively. Additionally, compared to lightweight networks, the proposed algorithm displayed increases in recognition accuracy of 3%, 4.2%, and 3.7%, respectively. Furthermore, when compared with the commonly used algorithm models, the model's accuracy of recognition was 1.1% and 1.7% higher, respectively.

4.2.3. Ablation Experiment

To evaluate the efficacy of the improved model, we conducted an ablation experiment using a dataset of strawberry maturity. The enhanced YOLOv5s-BiCE model was compared with existing models, and Table 4 summarizes the corresponding algorithm combinations represented by YOLOv5s-B, YOLOv5-C, and other models proposed in this article. The mAP_0.5 curve and mAP_0.5:0.95 were used to assess performance, and the average precision (mAP) values for various models using the improved method are displayed in Figures 12 and 13, respectively.

Table 4. The composition of the algorithm.

Algorithm 1	Algorithm 2	Algorithm 3
YOLOv5s	BiFormer	YOLOv5s-B
YOLOv5s	BiFPN	YOLOv5s-Bi
YOLOv5s	CARAFE	YOLOv5s-C
YOLOv5s	Focal_EIOU	YOLOv5s-E

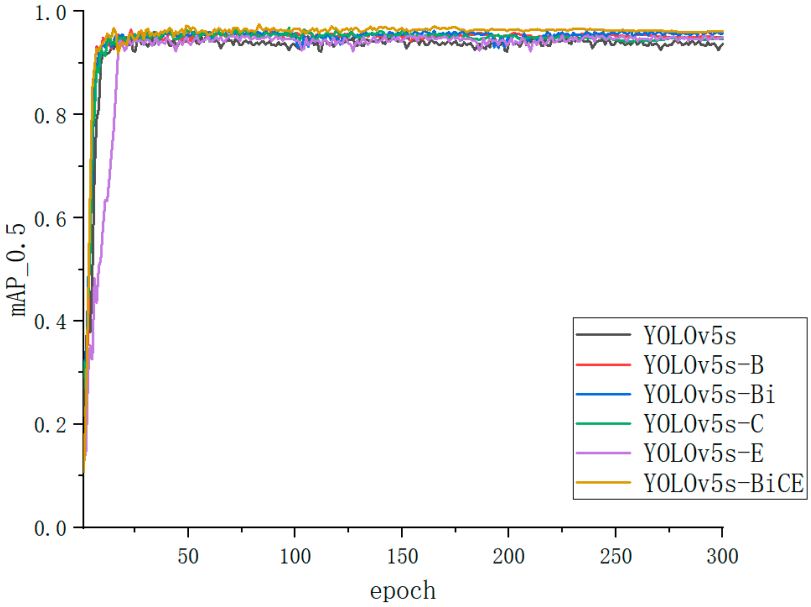

Figure 12. mAP_0.5 curves of different improved methods.

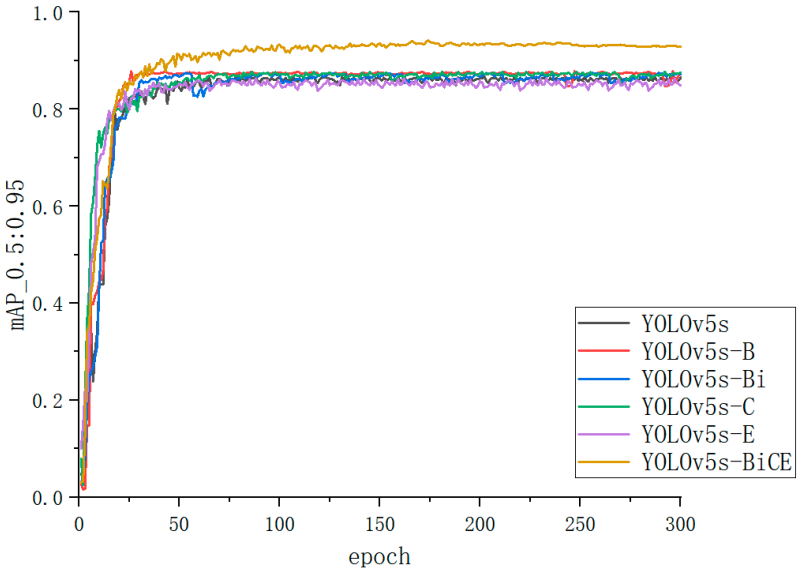

Figure 13. mAP_0.5:0.95 curves of different improved methods.

Table 5 presents the training outcomes of various models. The results demonstrate that the enhanced algorithm outperformed the other models.

Table 5. Results of ablation experiment.

Algorithm	P/%	R/%	mAP_0.5/%	mAP_0.5:0.95/%	Size/MB
YOLOv5s	88.2	90.9	93.3	85.5	14.4
YOLOv5s-B	92.8	90.1	95	87.7	14.9
YOLOv5s-Bi	92.3	91	94.6	87.1	14.5
YOLOv5s-C	92.1	90.6	94.7	87.3	14.7
YOLOv5s-E	89.9	89.8	94.8	86.5	14.4
YOLOv5s-BiCE	94.5	93.4	96.1	92.9	15.3

Table 5 illustrates that the improved YOLOv5s-BiCE model achieved an mAP_0.5 of 96.1% and an mAP_0.5:0.95 ratio of 92.9%, with a model size of only 15.3 MB. Comparing these results with those of the original model, we observed a 2.8% improvement in mAP_0.5, a 7.4% increase in mAP_0.5:0.95, and only a slight increase in model size by 0.9 MB.

4.3. Android Deployment Testing

In-depth learning parameters trained using a PC are often stored in a specified model and cannot be applied to all hardware platforms. A model transplanted to a mobile phone needs to perform parameter extraction and format conversion first. Figure 14 depicts a schematic diagram illustrating the deployment process of the strawberry maturity recognition model to an Android terminal. Torchscript is a high-performance reasoning framework suitable for deep learning on mobiles, with a low level of precision loss and fast calculation speed, which is suitable for model transplantation to mobiles. The strawberry maturity recognition model was successfully transferred to an Android phone. First, the PTH model file trained by PyTorch needed to be transformed into the Torchscript.ptl model, and we checked whether the model was wrong. Finally, according to the app design requirements, an Android project was built to deploy the Torchscript model to the handset for accuracy testing.

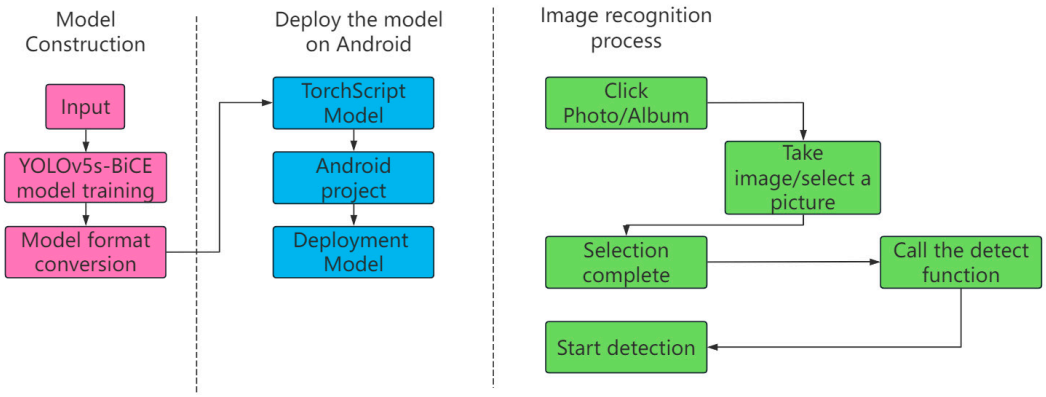

Figure 14. Flowchart of strawberry maturity model's deployment to an Android.

The main functions of the strawberry maturity recognition APP include the following: image acquisition, image storage, strawberry maturity detection, detection result category count display, and real-time detection. Users can capture strawberry images with their mobile phones through the image acquisition module or use their own images. The strawberry maturity detection module analyzes the maturity of the target strawberry and outputs the maturity information, location, and number of different strawberry maturity levels in the target image. Real-time detection can allow mobile cameras to detect the real-time maturity of strawberries within the scope of the camera used and output the maturity information. The effect diagram is shown in Figure 15. The save module can save the required detected image to a phone album after the detection is performed.

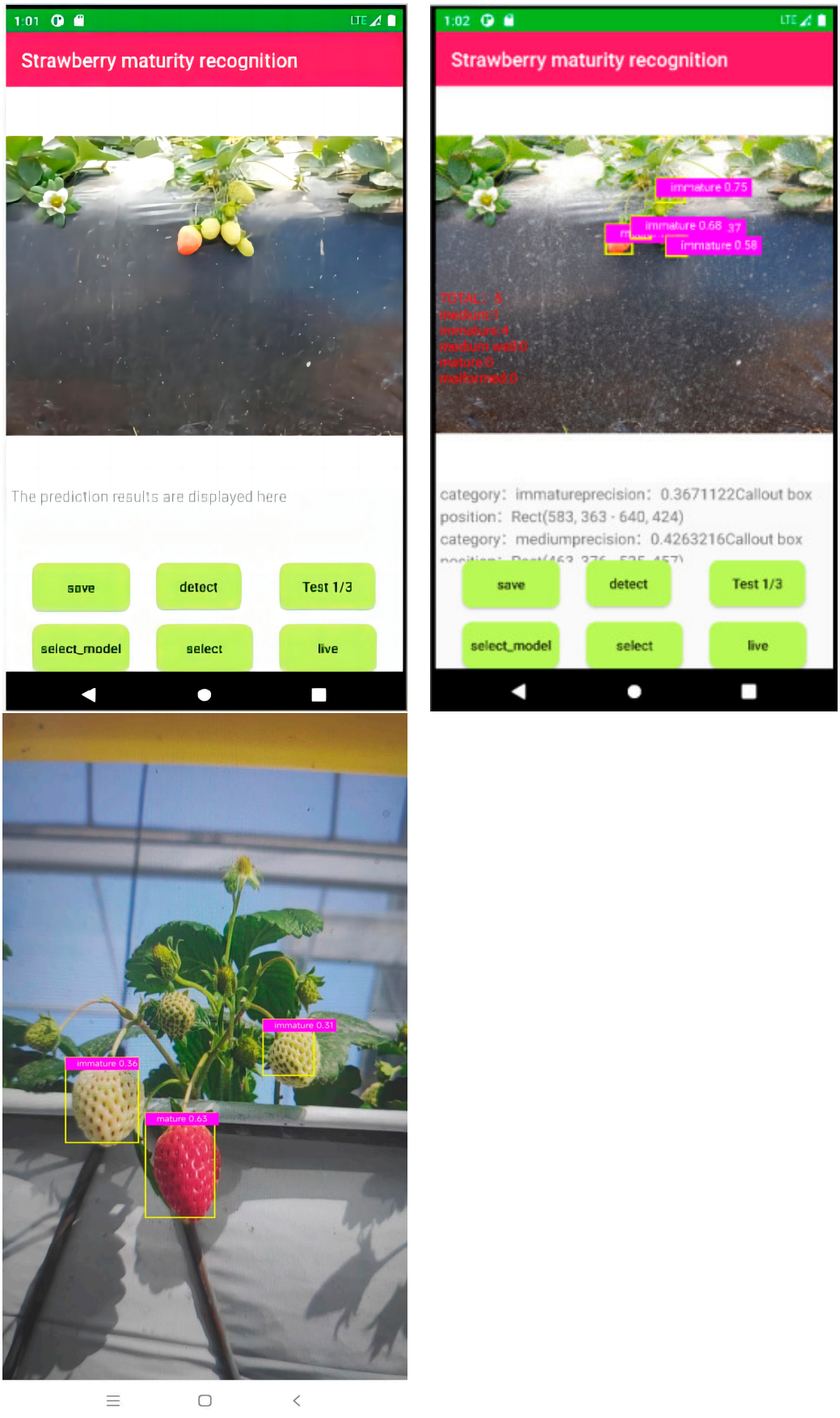

Figure 15. Effective picture for strawberry maturity detection app.

The deployment of the strawberry ripeness model on the Android platform allowed for real-time detection and provided advantages in terms of mobility, user-friendliness, data transmission, and system integration. This can be convenient for and benefit agricultural management and the quality control of agricultural products. It also laid the foundation for strawberry yield estimation and the development of strawberry harvesting and sorting robots.

4.4. Combination Experiment with Detection Robots

To evaluate the efficacy of this proposed algorithm in strawberry maturity recognition and tracking, it was tested on a self-built strawberry detection and tracking dataset. The parameter settings of the original YOLOv5s-DeepSort algorithm were utilized to compare the model's performance before and after the improvement. The comparative results of the model's performance are presented in Table 6.

Table 6. Model performance comparison.

Algorithm	MOTA/%	MOTP/%	FPS
YOLOv5s-DeepSort	84.4	82.6	25
YOLOv5s-BiCE-DeepSort	91.3	90.1	51

Table 6 demonstrates that the algorithm proposed in this paper effectively enhanced the detection accuracy and speed of strawberry maturity in target tracking. The model achieved an accuracy rate of 91.3% and a speed of 51 fps, satisfying demands for the real-time detection and tracking of strawberry maturity.

The tracking accuracy evaluation indices used in this study were *MOTA* and *MOTP*. The calculation method for these evaluation indices is as follows:

$$MOTA = 1 - \frac{\sum_t FN + FP + IDS}{\sum_t GT_t} \tag{11}$$

$$MOTP = \frac{\sum_{t,i} d_{t,i}}{\sum_t c_t} \tag{12}$$

In the *MOTA* formula for multi-target tracking accuracy, *FN* represents the number of positive classifications that were mistakenly identified as negative, *FP* represents the number of negative classifications that were falsely identified as positive, *IDS* represents the number of switch events, and *GT* refers to the number of actual targets present in the video. In the formula for multi-target tracking accuracy *MOTP*, $d_{t,i}$ represents the average distance between predicted and true bounding boxes, and c_t represents the number of successfully matched targets at frame t.

In order to provide a clearer demonstration of the practical application of the proposed algorithm in this study, we used an inspection robot for verification, as shown in Figure 16. The inspection robot was able to perform the stable detection and tracking of strawberries while also accurately assessing their levels of maturity. This demonstrated that deploying the strawberry ripeness model on the inspection robot enabled the real-time detection and monitoring of the strawberries' ripeness, improving the efficiency and accuracy of the detection. Additionally, it also enabled automation and data collection, providing strong support for strawberry cultivation management. This proved the effectiveness and potential usefulness of YOLOv5s-BiCE in real-world scenarios, laying the foundation for the future development of strawberry picking and sorting robots.

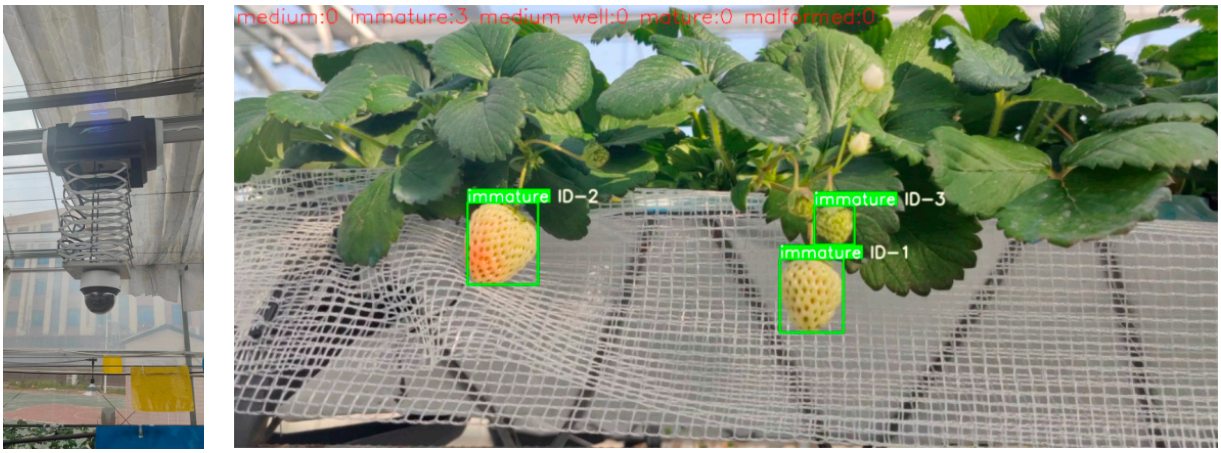

Figure 16. Identification and assessment of strawberry maturity via inspection robot.

5. Conclusions

This paper introduces an advanced YOLOv5s-BiCE algorithm for strawberry maturity recognition based on the improved YOLOv5s model. The algorithm effectively addresses issues of low accuracy, imperfect maturity classifications, and the lack of universality. It incorporates a CARAFE module structure to enhance content-aware processing, expand the field of view, and maintain model efficiency. Additionally, it improves small-object detection through a dual-attention mechanism and enhances accuracy with Focal_EIOU optimization. Additionally, in multi-scale feature fusion, BiFPN is introduced to reduce the number of redundant computations.

We proposed the YOLOv5s-BiCE model for strawberry maturity detection, which achieved an mAP of 96.1%. In the experimental environment, a single strawberry image was detected within 9 ms, and the entire test set was detected within 2.6 s, almost instantaneously detecting maturity information and counting the number of strawberries within each category. Furthermore, our ablation experiments and comparisons with other target recognition models demonstrated that our proposed algorithm model has a higher accuracy, faster detection speed, and better robustness than other models, such as YOLOv7. We also transferred this algorithm to the Android platform and used it within inspection robots, achieving real-time detection, both of which are very suitable for applications such as strawberry yield estimations and auxiliary picking robots.

Although the proposed improvement algorithm in this paper has achieved excellent results to some extent, there are still certain limitations, such as data dependency and more complex environmental detection. In future works, we will study its potential expansion in specific domains, expand it to a more diverse dataset, and integrate the model with other hardware and emerging technologies. In conclusion, the research results of this paper demonstrate substantial progress in applying the strawberry ripeness model to real-world problems. However, we also indicate a need to further improve the algorithm's robustness, speed, and transferability to keep up with the rapid development in precision agriculture and autonomous robotics.

Author Contributions: Data curation, Z.T. and W.L.; formal analysis, Z.T. and Y.R.; investigation, K.L.; methodology, Z.T. and K.L.; project administration, K.L.; resources, J.Z. and W.L.; supervision, J.Z. and Y.R.; validation, Z.T.; writing—original draft preparation, Z.T.; writing—review and editing, J.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by Guizhou Province Science and Technology Plan Project (No. Guizhou Province Science and Technology Contract Achievements 2021-119), the University Synergy Innovation Program of Anhui Province (No. GXXT-2022-041), the National Natural Science Foundation of China (No. 32272498), the Anhui Provincial Quality Engineering Project of Higher Education Institutions (2022jyxm464), and the Anhui Agricultural University Introduction and Stabilization of Talents Research Funding (No. yj2020-74).

Data Availability Statement: The datasets generated for this study are available upon request from the corresponding author.

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

References

1. Fait, A.; Hanhineva, K.; Beleggia, R.; Dai, N.; Rogachev, I.; Nikiforova, V.J.; Fernie, A.R.; Aharoni, A. Reconfiguration of the achene and receptacle metabolic networks during strawberry fruit development. *Plant Physiol.* **2008**, *148*, 730–750. [CrossRef] [PubMed]
2. Nunes, M.C.N.; Brecht, J.K.; Morais, A.; Sargent, S.A. Physicochemical changes during strawberry development in the field compared to those that occur in harvested fruit during storage. *J. Sci. Food Agric.* **2006**, *86*, 180–190. [CrossRef]
3. Sturm, K.; Koron, D.; Stampar, F. The composition of fruit of different strawberry varieties depending on maturity stage. *Food Chem.* **2003**, *83*, 417–422. [CrossRef]
4. Zhou, C.; Hu, J.; Xu, Z.; Yue, J.; Ye, H.; Yang, G. A novel greenhouse-based system for the detection and plumpness assessment of strawberry using an improved deep learning technique. *Front. Plant Sci.* **2020**, *11*, 559. [CrossRef] [PubMed]
5. Hayashi, S.; Yamamoto, S.; Saito, S.; Ochiai, Y.; Kamata, J.; Kurita, M.; Yamamoto, K. Field operation of a movable strawberry-harvesting robot using a travel platform. *Jpn. Agric. Res. Q. JARQ* **2014**, *48*, 307–316. [CrossRef]
6. Xiong, Y.; Ge, Y.; Grimstad, L.; From, P.J. An autonomous strawberry-harvesting robot: Design, development, integration, and field evaluation. *J. Field Robot.* **2020**, *37*, 202–224. [CrossRef]
7. Xiong, Y.; Peng, C.; Grimstad, L.; From, P.J.; Isler, V. Development and field evaluation of a strawberry harvesting robot with a cable-driven gripper. *Comput. Electron. Agric.* **2019**, *157*, 392–402. [CrossRef]
8. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef] [PubMed]
9. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
10. Cai, Z.; Vasconcelos, N. Cascade r-cnn: Delving into high quality object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6154–6162.
11. Wang, M.; Fu, B.; Fan, J.; Wang, Y.; Zhang, L.; Xia, C. Sweet potato leaf detection in a natural scene based on faster R-CNN with a visual attention mechanism and DIoU-NMS. *Ecol. Inform.* **2023**, *73*, 101931. [CrossRef]
12. Mu, X.; He, L.; Heinemann, P.; Schupp, J.; Karkee, M. Mask R-CNN based apple flower detection and king flower identification for precision pollination. *Smart Agric. Technol.* **2023**, *4*, 100151. [CrossRef]
13. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
14. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.E.; Fu, C.-Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. *arXiv* **2015**, arXiv:1512.02325.
15. Li, R.; Ji, Z.; Hu, S.; Huang, X.; Yang, J.; Li, W. Tomato Maturity Recognition Model Based on Improved YOLOv5 in Greenhouse. *Agronomy* **2023**, *13*, 603. [CrossRef]
16. Li, T.; Sun, M.; He, Q.; Zhang, G.; Shi, G.; Ding, X.; Lin, S. Tomato recognition and location algorithm based on improved YOLOv5. *Comput. Electron. Agric.* **2023**, *208*, 107759. [CrossRef]
17. Fan, Y.; Zhang, S.; Feng, K.; Qian, K.; Wang, Y.; Qin, S. Strawberry Maturity Recognition Algorithm Combining Dark Channel Enhancement and YOLOv5. *Sensors* **2022**, *22*, 419. [CrossRef] [PubMed]
18. Sekharamantry, P.K.; Melgani, F.; Malacarne, J. Deep Learning-Based Apple Detection with Attention Module and Improved Loss Function in YOLO. *Remote Sens.* **2023**, *15*, 1516. [CrossRef]
19. Wang, Y.; Xing, Z.; Ma, L.; Qu, A.; Xue, J. Object Detection Algorithm for Lingwu Long Jujubes Based on the Improved SSD. *Agriculture* **2022**, *12*, 1456. [CrossRef]
20. Jie, H.; Li, S.; Samuel, A.; Gang, S.; Enhua, W. Squeeze-and-Excitation Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *42*. [CrossRef]
21. Woo, S.; Park, J.; Lee, J.-Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
22. Hou, Q.; Zhou, D.; Feng, J. Coordinate attention for efficient mobile network design. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 13713–13722.
23. Zhu, L.; Wang, X.; Ke, Z.; Zhang, W.; Lau, R.W. BiFormer: Vision Transformer with Bi-Level Routing Attention. In Proceedings of the Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023. [CrossRef]
24. Wang, J.; Chen, K.; Xu, R.; Liu, Z.; Loy, C.C.; Lin, D. Carafe: Content-aware reassembly of features. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 3007–3016.
25. Wang, K.; Liew, J.H.; Zou, Y.; Zhou, D.; Feng, J. Panet: Few-shot image semantic segmentation with prototype alignment. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 9197–9206.

26. Tan, M.; Pang, R.; Le, Q.V. EfficientDet: Scalable and efficient object detection. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020.
27. Zhang, Y.-F.; Ren, W.; Zhang, Z.; Jia, Z.; Wang, L.; Tan, T. Focal and efficient IOU loss for accurate bounding box regression. *Neurocomputing* **2022**, *506*, 146–157. [CrossRef]