

Article

Lightweight Non-Destructive Detection of Diseased Apples Based on Structural Re-Parameterization Technique

Bo Han ^{1,2,3}, Ziao Lu ^{1,2,3}, Luan Dong ^{1,2,3} and Jingjing Zhang ^{1,2,3,*}

¹ College of Computer and Information Engineering, Xinjiang Agricultural University, Urumqi 830052, China; hanboxjau@163.com (B.H.); luziaoxjau@163.com (Z.L.); dl@xjau.edu.cn (L.D.)

² Engineering Research Center of Intelligent Agriculture Ministry of Education, Urumqi 830052, China

³ Xinjiang Agricultural Informatization Engineering Technology Research Center, Urumqi 830052, China

* Correspondence: zll@xjau.edu.cn

Abstract: This study addresses the challenges in the non-destructive detection of diseased apples, specifically the high complexity and poor real-time performance of the classification model for detecting diseased fruits in apple grading. Research is conducted on a lightweight model for apple defect recognition, and an improved VEW-YOLOv8n method is proposed. The backbone network incorporates a lightweight, re-parameterization VanillaC2f module, reducing both complexity and the number of parameters, and it employs an extended activation function to enhance the model's nonlinear expression capability. In the neck network, an Efficient-Neck lightweight structure, developed using the lightweight modules and augmented with a channel shuffling strategy, decreases the computational load while ensuring comprehensive feature information fusion. The model's robustness and generalization ability are further enhanced by employing the WIoU bounding box loss function, evaluating the quality of anchor frames using outlier metrics, and incorporating a dynamically updated gradient gain assignment strategy. Experimental results indicate that the improved model surpasses the YOLOv8n model, achieving a 2.7% increase in average accuracy, a 24.3% reduction in parameters, a 28.0% decrease in computational volume, and an 8.5% improvement in inference speed. This technology offers a novel, effective method for the non-destructive detection of diseased fruits in apple grading working procedures.

Keywords: artificial intelligence; non-destructive detection; diseased apples; lightweight; YOLOv8



Citation: Han, B.; Lu, Z.; Dong, L.; Zhang, J. Lightweight

Non-Destructive Detection of Diseased Apples Based on Structural Re-Parameterization Technique. *Appl. Sci.* **2024**, *14*, 1907. <https://doi.org/10.3390/app14051907>

Academic Editor: Sungho Kim

Received: 26 January 2024

Revised: 17 February 2024

Accepted: 21 February 2024

Published: 26 February 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The apple, a vital element of China's fruit industry, offers a broad planting area and high total output [1]. Significant progress in automatic apple grading technology offers advantages in speed, efficiency, and human error reduction over traditional manual grading. Developing high-efficiency and stable automatic apple grading equipment necessitates high-performance, lightweight grading models, fostering the application of computer vision in apple grading [2]. This paper concentrates on recognizing and screening diseased apples to replicate manual grading scenarios, excluding substandard fruits, curtailing grading losses, and initiating lightweight recognition model research for diseased apples.

Traditional target detection methods, such as sliding windows and manual feature extraction, are exemplified by techniques like Haar [3], HOG [4], Hu moment [5], SIFT [6], SURF [7], and DPM [8]. The evolution of computer vision and deep learning has ushered target detection into agricultural production prominence, with algorithms bifurcated into single-stage (e.g., YOLO series [9–11], SSD series [12–14], RetinaNet series [15,16]) and two-stage detection algorithms (e.g., RCNN series [17], FasterRCNN series [18]). Apple target detection, melding computer vision and agriculture, automates apple identification and localization in images. Fan et al. [19] refined the YOLOv4 detection algorithm for apple defect identification, employing channel and layer pruning strategies and an L1-paradigm non-extreme value suppression method to prune redundant detection frames,

achieving a 93.9% average detection accuracy. Sun et al. [20] integrated the Res2Net module into the RetinaNet algorithm, coupling a weighted bi-directional feature pyramid network with a focus-based loss and efficient intersection and concatenation ratio in the joint loss function for apple target detection. Zhang et al. [21] implemented GhostNet in the improved YOLOv4 apple target detection task, reconstructing the feature extraction network with a depth-separable convolutional lightweight necking network and detection head, integrating coordinate attention into the feature pyramid, and reconfiguring the feature extraction network. While enhancing model detection accuracy, such advancements often escalate model complexity. Some lightweight apple target detection models depend on depth-separable methods, which, although reducing weight, forfeit global channel information, leading to a concomitant problem of accuracy loss [22]. At present, in most of the apple target detection tasks, the main research is on the detection of apple fruit, for the apple grading system in the fruit disease identification research is relatively small. At the same time, most of the detection tasks focus on improving the detection accuracy of the model, resulting in the complexity of the model becoming high. Some of the lightweight apple target detection models mainly use the depth of separation of lightweight method, and the depth of separation of convolution cuts off the cross-channel communication between the information. Due to the loss of global channel information, the backbone of the network extraction of the features of the information is not adequate. For the model to be lightweight, at the same time it faces the loss of accuracy. If the model is to be lightweight at the same time, the loss of precision is a problem.

Aiming at the identified challenges, the authors of this paper, using the YOLOv8n algorithm, the structural re-parameterization technique, and efficient neck structure, designed a lightweight non-destructive detection of diseased fruits in apple grading, simulated the initial apple screening scenario of manual grading, and propose a lightweight, high-performance diseased apple recognition model. This model focuses on the recognition and screening of four apple appearance features: healthy, speckled, rotted, and scabbed. The model's improvements include several aspects:

(1) To reduce the number and complexity of parameters in the backbone network and facilitate deployment in Apple's hierarchical pipeline devices, a structure re-parameterization of the VanillaC2f module structure is proposed. Utilizing the lightweight VanillaBlock module to construct the C2f module, and forming the lightweight Vanilla-Backbone, this approach decreases the number of parameters and complexity of the backbone network, achieving its lightweighting. The integration of a deep training strategy with an extended activation function ensures both the model's detection accuracy and its capability for real-time detection.

(2) To thoroughly integrate the feature map information extracted by the backbone network, enabling the model to fully learn the differences in features of various apple defects, and to enhance the detection accuracy for the apple grading pipeline task without increasing the model size, an Efficient-Neck lightweight neck network is introduced. This network employs Ghost Shaded Convolution of Blending and Shadows (GSConv) and a one-time aggregation of the cross-graded part of the network module (VoVGSCSP) to construct the Efficient-Neck structure. This setup realizes the neck network's lightweighting while combining it with a channel blending strategy to further enhance the model's detection accuracy.

(3) To augment the robustness and generalization capability of the apple grading pipeline model, the Wise-IoU (Bounding Box Regression Loss with Dynamic Focusing Mechanism) [23] bounding box loss function, featuring an "outlier" quality assessment index and a dynamically updated gradient gain allocation strategy, is incorporated.

Based on the above improvement methods, an enhanced YOLOv8n model is proposed. This improved model reduces the number of parameters by 24.3% and the computational volume by 28.0% compared to the original YOLOv8n, while ensuring an increase in accuracy, making it suitable for deployment in devices for online apple grading. The body of the

article includes four main parts: materials and methods, results and analysis, discussion, and conclusion.

2. Materials and Methods

2.1. Construction and Enhancement of Datasets

Apple grading is mainly performed in an indoor environment, and the initial screening task of defective apples may be accomplished in the orchard picking base. For the task of screening defective apples in the assembly line, while the dataset of healthy fruits is ample, that of diseased fruits is deficient. Given the challenge of constructing a diverse dataset of diseased apples through field shooting, to improve the robustness and generalization of the model, this paper collates apple datasets of varying scales, backgrounds, and brightnesses through manual autonomous shooting and internet collection. The dataset from manual field shooting is sourced from the picturesque Diyarzhimu orchard and town of Yiganqi in Aksu, China, using a Smartphone (iPhone 14 Pro). The shooting environment is shown in Figure 1.



Figure 1. Apples from different orchards in the same region. (a) the town of Yiganqi, (b) the picturesque Diyarzhimu orchard.

Manual collection is based on a relatively simple environmental context. It was carried out in different lighting conditions (low and strong light) and from various shooting angles; this occurred between 10:00 and 16:00. Internet collection involved crawling public images and downloading public datasets, followed by manual screening to retain clear, high-quality images and remove any images with low resolution or overly complex backgrounds. The constructed dataset, divided in an 8:1:1 ratio for the training set, test set, and validation set, underwent data enhancement processing. Image enhancement techniques, such as rotation by angle and the addition of random noise, were applied to the dataset images. The final dataset comprised 2000 images, manually labeled in VOC format with XML files, which were then converted to the TXT format of YOLO using custom code. The dataset included four classes: (1) HEALTHY, (2) BLOTCH, (3) ROT, (4) SCAB. The original dataset, categories, and an example plot of the enhanced data are illustrated in Figure 2.

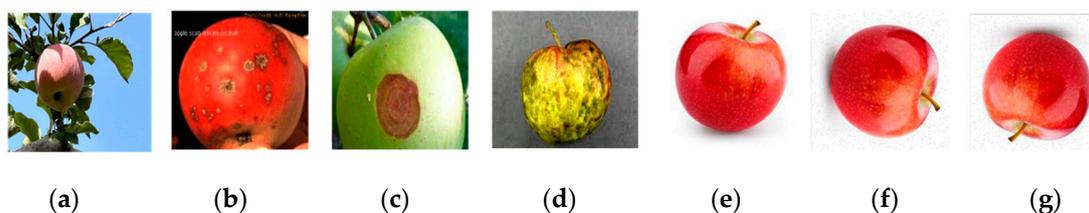


Figure 2. Apple Dataset. The dataset included (a) HEALTHY apples, (b) SCAB-affected apples, (c) ROT-affected apples, (d) BLOTCH-affected apples, (e) original images, (f) images rotated 90 degrees with added noise, and (g) images rotated 180 degrees with added noise.

2.2. YOLOv8n Target Detection Algorithm

YOLOv8n [24], a lightweight variant of YOLOv8, features fewer parameters and reduced computational demands. Its network structure, depicted in Figure 3, comprises three main components: the backbone network (Backbone), the neck network (Neck), and the prediction head (Head). The backbone network primarily extracts feature information, incorporating the C2f module, which merges the C3 module of YOLOv5 and the ELAN of YOLOv7 [25] for enriched gradient flow information. The neck network, responsible for feature information fusion, utilizes a multi-scale feature fusion structure (FPN-APN). The prediction head adopts the decoupled-head (Decoupled-Head) from YOLOX [26], comprising a classification head and a regression head. The regression head's loss function includes CioU and Distribution Focal Loss (DFL), while the classification head uses the biclassification cross-entropy loss function (BCE) [27]. Sample matching employs the TaskAlignedAssigner [28] for positive and negative sample allocation and the Anchor-Free [29] strategy.

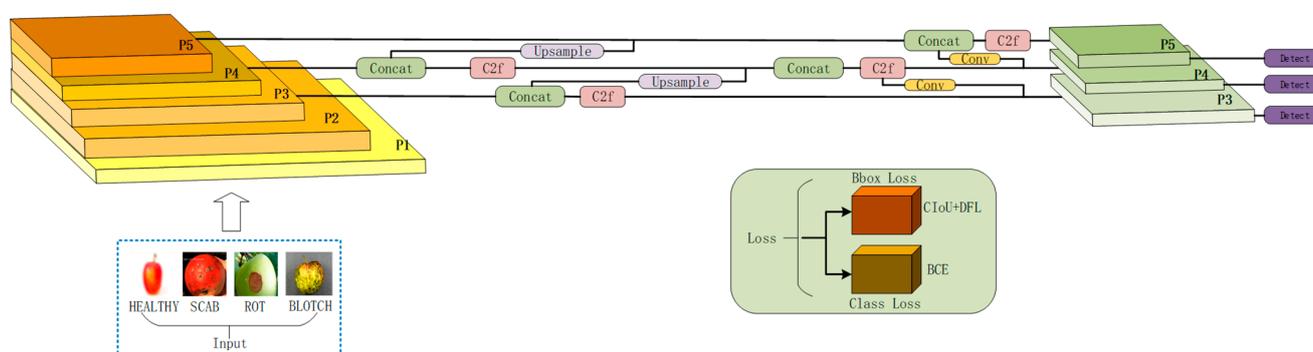


Figure 3. Network structure of YOLOv8n.

2.3. Lightweight Backbone Network

The efficacy of apple grading tasks hinges on the efficiency and stability of edge control devices. Utilizing embedded devices in the apple grading pipeline, the research and deployment of the primary apple screening model aim for high performance and lightweight design. However, for detecting and identifying diseased apple fruit, the YOLOv8n's backbone network faces challenges due to its large number of parameters and complexity, complicating deployment in cost-effective apple grading pipelines. A lightweight VanillaC2f module was proposed, which reduces both complexity and the number of parameters in the backbone network. Additionally, an extended activation function was employed to enhance the model's nonlinear expression capability.

2.3.1. VanillaNet Neural Network

VanillaNet [30] is an efficient, lightweight neural network employing a deep training strategy and an extended activation function to enhance model inference speed while maintaining performance:

(1) The deep training strategy involves initially training two convolutional layers with one activation function. As training iterations increase, the convolution operation simplifies into a constant mapping. Ultimately, at the inference phase's end, the two convolutions merge into a single convolution, reducing inference time and enabling reparametrization structure.

(2) The extended activation function uses parallelized stacked activation functions to replace successive stacked functions, avoiding latency issues in excess computational scenarios. This function modifies neighboring inputs on the feature map, learning global

information. The expression of the scalability activation function for an input size of is $Input \in R^{C \times H \times W}$ shown in Equation (1):

$$A_s(input_{h,w,c}) = \sum_{i,j \in \{-n,n\}} a_{i,j,c} A(input_{i+h,j+w,c} + b_c) \tag{1}$$

where n represents number of stacked activation functions, a, b represents the scale and bias of each activation function, $h \in \{1, 2, 3, \dots, H\}, w \in \{1, 2, 3, \dots, W\}, c \in \{1, 2, 3, \dots, C\}$ represents the width, height and the number of channels of the input feature map.

2.3.2. VanillaBlock

The VanillaBlock comprises standard convolution (Conv), batch normalization (BN), LeakReLU activation function, and extended activation function (ImActivation). It uses a 1×1 convolution kernel, preserving feature map information while minimizing computational costs. The activation functions, with applied post-standard convolution and combined with a BN layer, streamline the network training process. In the inference phase, the Conv and BN layers in the Conv_BN module merge first, followed by merging the weights of the two Convs and trimming the LeakyReLU layer. Lastly, the Gconv and BN layers in the extended activation function merge. The VanillaBlock module’s design follows a deep training strategy, with its structure differing in training and inference phases. During the training phase, a BN layer is included to simplify network training, and in the inference phase, the BN layer merges with the convolutional layer, reducing model complexity. The structure of the VanillaBlock module in the training and inference phases is illustrated in Figure 4.

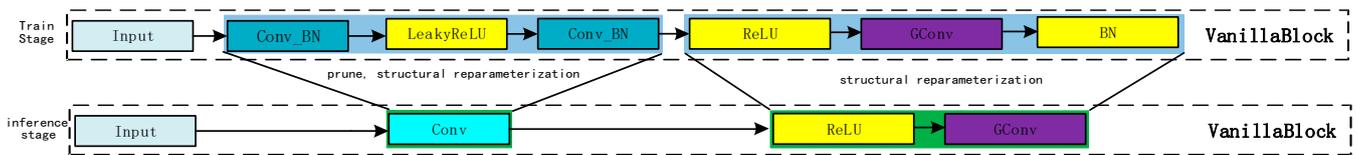


Figure 4. Module structure of VanillaBlock.

The extended activation function (ImActivation) in VanillaBlock is realized through the combination of the ReLU activation function and grouped convolution (Gconv), facilitating parallelized stacked activation functions. Grouped convolution not only reduces the parameter count but also achieves sparse convolution operations, providing a degree of regularization. The structure of ImActivation is detailed in Figure 5.

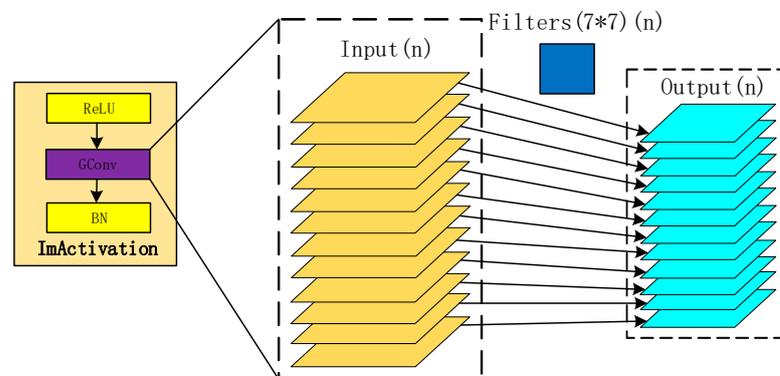


Figure 5. Module structure of ImActivation.

2.3.3. VanillaC2f

The C2f structure in the YOLOv8n model offers rich gradient flow, yet its Bottleneck in the inference stage contains numerous standard volumes and residual modules, leading to

high parameter counts and computational complexity. This paper introduces the efficient VanillaC2f module, based on the C2f module’s internal structure design and integrating VanillaBlock. During inference, the VanillaC2f module reparametrizes by merging all BN layers in VanillaBlock, removing the LeakReLU layer and consolidating the two Convs into one. This reparameterization achieves a lightweight C2f module with reduced parameters and complexity compared to YOLOv8n’s C2f module, as illustrated in Figure 6.

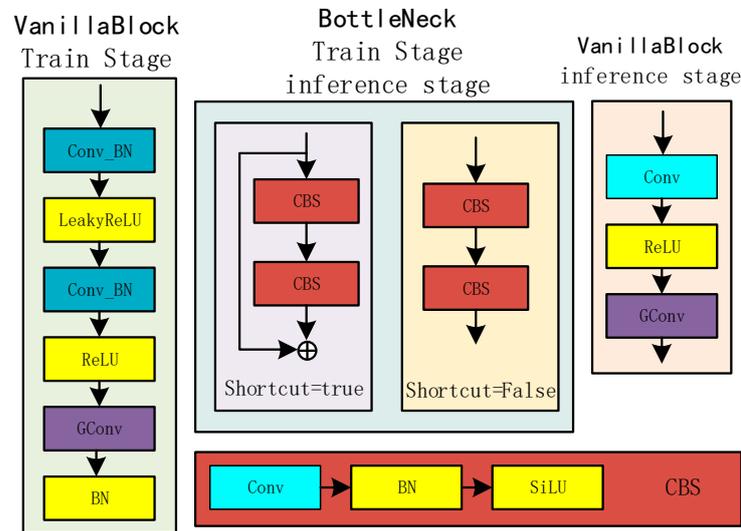


Figure 6. Comparison of VanillaC2f and C2f.

2.3.4. Vanilla-Backbone

The improved Vanilla-Backbone network comprises VanillaC2f, CBS, and SPPF modules. The CBS module, adopting two-dimensional standard convolution (BatchNorm2D) and SiLU activation, focuses on feature extraction. The SPPF module follows YOLOv8n’s backbone network structure. The design of Vanilla-Backbone, depicted in Figure 7, prioritizes efficient feature extraction while maintaining a lightweight architecture.

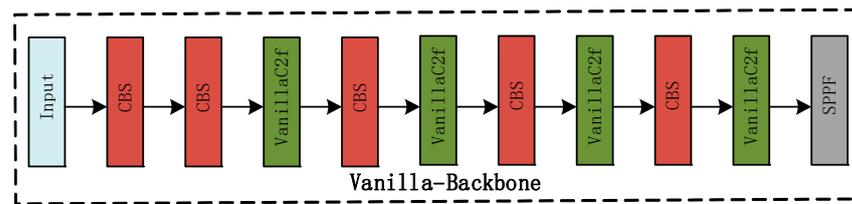


Figure 7. Improved backbone network.

2.4. Lightweight Neck Network

2.4.1. GSConv Module

Addressing the limitations of depth-separable convolution 1×1 , which neglects channel information leading to reduced accuracy, the GSConv module merges standard convolution (Conv), depth-separable convolution (DWConv), and shuffle operations. This combination achieves module lightweighting while fully extracting and fusing feature information, thus, enhancing model accuracy. The GSConv module structure is shown in Figure 8.

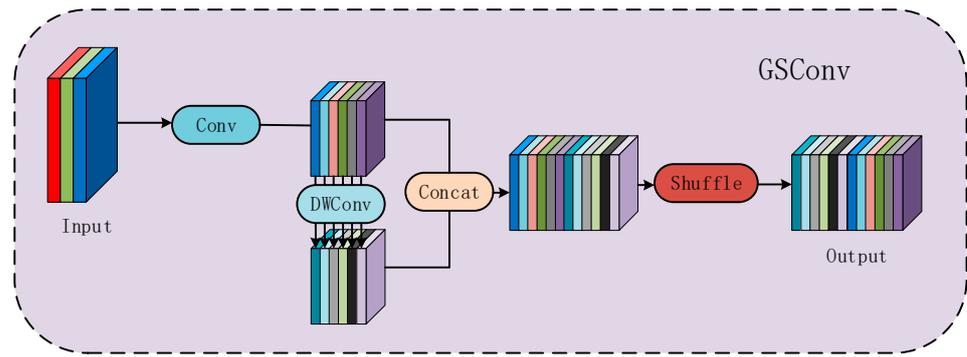


Figure 8. GSConv module structure.

2.4.2. VoVGSCSP Module

Incorporating the GSBottleneck from the ghost blending module, the neck network’s VoVGSCSP module fuses features effectively through a one-time aggregated cross-stage localized network (Figure 8). It utilizes dual branches for feature extraction, aggregates the feature maps, and applies convolution to the multi-channel aggregated maps for richer information. Additionally, it integrates residue-like cross-stage operations to enhance the neck network’s nonlinear expression ability. The VoVGSCSP module structure is depicted in Figure 9.

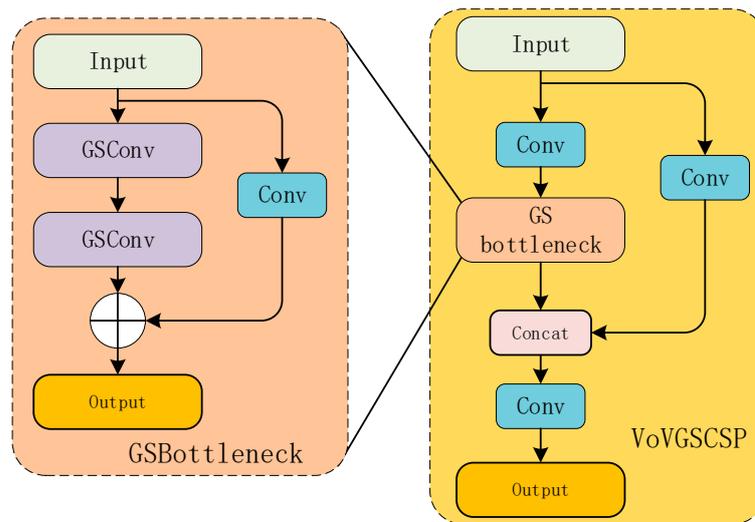


Figure 9. VoVGSCSP module structure.

2.4.3. Efficient-Neck

The YOLOv8n’s neck network, comprising CBS, Upsample, and C2f modules, faced challenges due to a high parameter count from numerous standard convolutions. The improved Efficient-Neck network utilizes a lightweight ghost shuffling module to replace the CBS module and the VoVGSCSP module to substitute the C2f module. This approach fully integrates feature map information from the backbone network, effectively extracting features of different diseased apple types, and improving model detection accuracy. The pre-improvement and post-improvement neck network structures are presented in Figure 10.

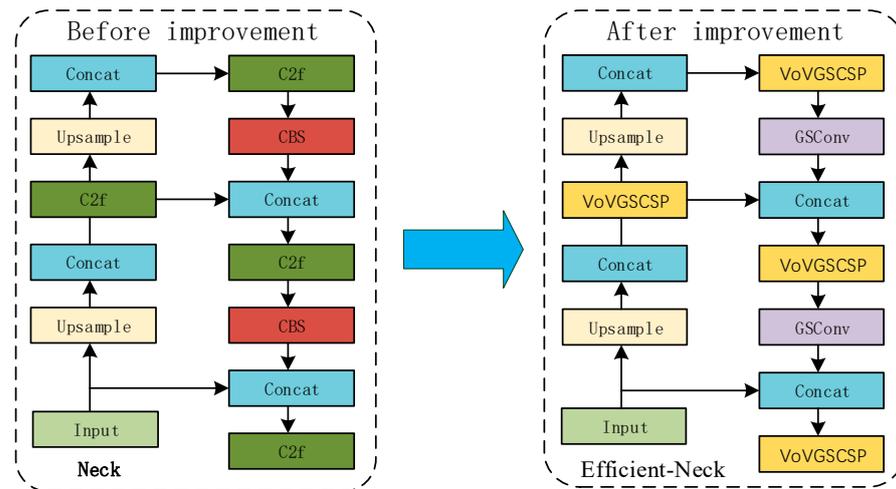


Figure 10. Improvement of the neck network.

2.5. Improved YOLOv8n Model (VEW-YOLOv8n)

Building on the YOLOv8n model, this study introduces the VEW-YOLOv8n, a lightweight apple target detection method. The improved model structure, shown in Figure 11, surpasses YOLOv8n in accuracy while adhering to lightweight design principles of low parameter count and minimal computational demand.

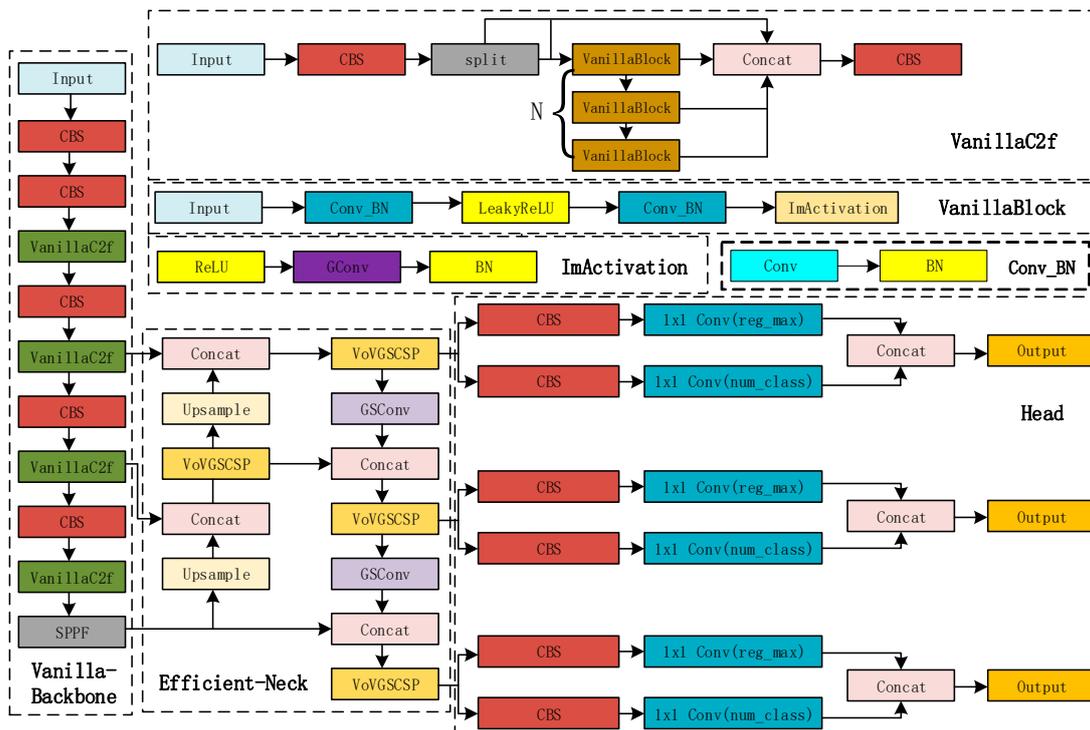


Figure 11. Network structure of VEW-YOLOv8n.

2.6. WIoU Bounding Box Loss Function

During the manual labeling process of the Apple dataset, human factors may introduce low-quality anchor frames, potentially impacting the IoU loss function’s efficacy during model training. The IoU loss function measures the similarity between predicted and actual bounding boxes by calculating the ratio of their intersection area to the area of their combined region, as depicted in Figure 12.

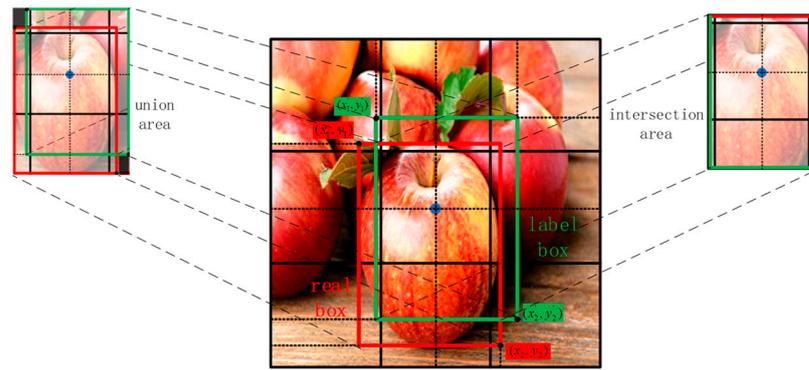


Figure 12. IoU Schematic.

The GioU [31] loss function extends the IoU by including the minimum external rectangle, including both the predicted and actual boxes. It accounts for overlapping and non-overlapping regions, effectively addressing gradient disappearance issues when boxes do not intersect. The GioU loss function reverts to the IoU loss function when the real and predicted boxes overlap or contain each other within the same dimension. The DioU [32] loss function further expands on GioU by considering the Euclidean distance between centroids of the predicted and real boxes, as well as the diagonal distance of their minimum external rectangle. However, it does not account for aspect ratio relationships. The CioU loss function, used in YOLOv8n’s bounding box loss function, builds upon DioU by adding aspect ratio consistency between the predicted and real boxes, including an aspect ratio penalty term. Although CioU addresses several shortcomings, it does not provide uniform gradient signs for anchor box widths and heights and involves a complex computational process that increases computational volume and training time. The SioU [33] loss function combines angle, distance, and shape costs but does not consider dynamic gradient updating. The WioU loss function addresses these issues by weighting the IoU based on the predicted and real frame region, introducing a dynamic non-monotonic focusing mechanism. It uses “outlier” as a new quality assessment criterion for anchor frames and dynamically updates the gradient gain allocation strategy.

2.6.1. WioUv1

WioUv1, a bounding box loss function with an attentional mechanism, is formulated in Equations (2)–(4).

$$L_{IoU} = 1 - IoU = 1 - \frac{W_i H_i}{wh + w_{gt} h_{gt} - W_i H_i} \overline{IoU} \tag{2}$$

$$R_{WIoU} = \exp\left(\frac{(x - x_{gt})^2 + (y - y_{gt})^2}{(W_g^2 + H_g^2)^*}\right) \tag{3}$$

$$L_{WIoUv1} = R_{WIoU} L_{IoU} \tag{4}$$

where W_g and H_g are the width and height of the smallest outer rectangle and are also stripped from the computational map to prevent R_{WIoU} from having an effect on the convergence speed, $R_{WIoU} \in [1, e]$, which can significantly amplify the ordinary quality anchor frames L_{IoU} , $L_{IoU} \in [0, 1]$ will significantly reduce the R_{WIoU} for high quality anchor frames, and will pay more attention to the distance between the centroids of the two frames when there is a high degree of overlap between the predicted and real frames.

2.6.2. WioUv2

WioUv2, incorporating the concept of focus loss, effectively reduces the contribution of simpler instances to the loss value. A monotonic focusing coefficient $(L^* IoU)^\gamma$ is added to WioUv1. Since the focusing coefficient decreases as L_{IoU} decreases, which results in

the late convergence becoming slow, a normalization factor of $\overline{L_{IoU}}$ is introduced. The final focusing coefficient can be expressed as $r_1 = (\frac{L^*IoU}{\overline{L_{IoU}}})^\gamma \in [0, 1]$, and the WioUv2 loss function is computed as shown in Equation (5):

$$L_{WIoUv2} = r_1 L_{WIoUv1} \quad (5)$$

Dynamically updating the normalization factor keeps the value of the focusing coefficient r_1 at a high level, effectively solving the problem of slow convergence at the late stage of training.

2.6.3. WioUv3

WioUv3 operates based on the outlier degree of anchor frames. The outlier degree coefficient β is shown in Equation (6):

$$\beta = \frac{L^*IoU}{\overline{L_{IoU}}} \in [0, +\infty) \quad (6)$$

The smaller the outlier degree, this coefficient β ensures that anchor frames of lower quality exert less influence. In WioUv3, a nonmonotonic focusing coefficient is added to WioUv1. This addition assigns smaller gradient gains to normal quality anchor frames and larger gradients are assigned smaller gradient gains. The use of the outlier degree coefficient β prevents the effects of low quality frames.

WioUv3 incorporates a nonmonotonic focusing coefficient r_2 into WioUv1, as delineated in Equations (7) and (8).

$$r_2 = \frac{\beta}{\delta \alpha^{\beta-\delta}} \quad (7)$$

$$L_{WIoUv3} = r_2 L_{WIoUv1} \quad (8)$$

where the anchor frame will receive the maximum gradient gain when $\beta = \mathbb{C}$. The normalization factor $\overline{L_{IoU}}$ in WioUv3 undergoes dynamic transformation and updates, thus, allowing the quality evaluation metrics of the anchor frame to be dynamic. Consequently, WioUv3 can select the most suitable gradient gain allocation strategy for the current sample at varying times.

This study employs the WioUv3 bounding box loss function to supplant the original CioU bounding box loss function. The WioU loss function mitigates the competitive influence of high-quality anchor frames and curbs the adverse impact of low-quality anchor frames. Combined with the dynamic gradient gain allocation strategy, this approach improves network robustness and augments the model's detection capability.

3. Results and Analysis

3.1. Experimental Environment and Parameter Settings

The experiment was conducted on a system with CentOS 7.9.2009, utilizing a 12 vCPU Intel(R) Xeon(R) Platinum 8255C CPU @ 2.50 GHz and an NVIDIA GeForce RTX 3090 graphics card. Anaconda served as the Integrated Development Environment (IDE), the Python version is 3.8.10, the YOLO series of target detection models were constructed and trained using version 1.11.0 of the PyTorch deep learning framework. The single-stage SSD target detection model and the FasterCNN two-stage target detection model were constructed and trained using version 2.25.3 of the mmdetection deep learning framework, with CUDA version 11.3 facilitating the acceleration of the training process. Parameters included an image resolution of 640×640 , the Adam optimizer, an initial learning rate of 1×10^{-3} , the momentum value of 0.937 and weight decay coefficient of 5×10^{-4} , which were incorporated along with a batch size of 64 for network training and 200 training rounds for both the baseline and improved models based on these settings. Mosaic data augmentation was used in the YOLO target detection algorithms.

3.2. Model Performance Evaluation Metrics

For assessing the enhanced model, six prevalent performance evaluation indices were adopted, namely mean average precision (mAP), average precision, Recall, Precision, GFLOPS, and parametric quantities of the model. The evaluation process is described in Equation (9):

$$mAP = \frac{1}{n} \sum_{i=0}^n AP_i \quad (9)$$

where n represents the number of categories, AP_i is the area included by the PR (precision-recall) curve for a category, and the formula for AP is shown in (10).

$$AP = \int_0^1 PR dr \quad (10)$$

The mAP in this experiment reflects the average across four categories: healthy, spotted, decayed, and crusted.

The accuracy rate, defined as the proportion of boxes accurately predicted by the model, measures model misdetection and is computed as per Equation (11).

$$Precision = \frac{TP}{TP + FP} \times 100\% \quad (11)$$

Recall, the ratio of correctly predicted frames to labeled frames by the model, measures model omission and is calculated according to Equation (12).

$$Recall = \frac{TP}{TP + FN} \times 100\% \quad (12)$$

where TP denotes the count of correct prediction results, FP represents incorrect prediction results, and FN refers to the unlabeled boxes that were not predicted.

3.3. Experiments Comparing the Improved Model with the Baseline Model

Under identical experimental conditions, the mAP and loss values of the improved YOLOv8n model (VEW-YOLOv8n) were compared with those of the original YOLOv8n model during training. Curve comparison graphs for mAP and loss values are depicted in Figure 13, with (a) illustrating the $mAP@0.5$ value, (b) showing the $mAP@0.5:0.95$ value, and (c) presenting the loss values. Figure 13 elucidates that both the $mAP@0.5$ and $mAP@0.5:0.95$ values of VEW-YOLOv8n surpassed those of the original YOLOv8n, while the loss value was lower. The paper demonstrates that the proposed VEW-YOLOv8n has superior convergence and detection accuracy in detecting apple diseases. Furthermore, the performance of VEW-YOLOv8n and YOLOv8n on the test set was verified, and the Precision-Recall curves (P-R curves) are delineated in Figure 14, which shows that the VEW-YOLOv8n model achieved a mAP value of 95.6%, while the YOLOv8n model achieved a mAP value of 92.9%. The improved model demonstrated an average accuracy improvement of 2.7% compared to the YOLOv8n model. This comparison indicates that the VEW YOLOv8n model also outperformed the YOLOv8n model on the test set, substantiating the superiority of the improved VEW-YOLOv8n over YOLOv8n.

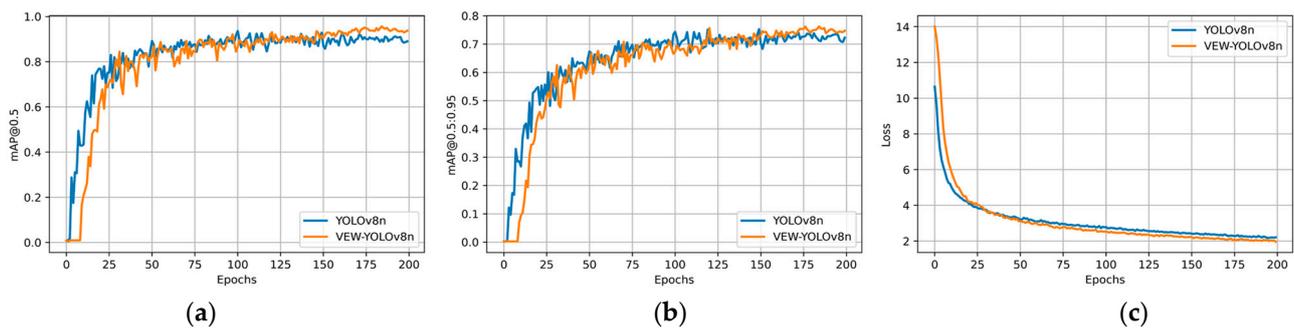


Figure 13. Comparison of mAP and loss value curves: (a) mAP@0.5 line graph; (b) mAP@0.5:0.95 line graph; (c) loss line graph.

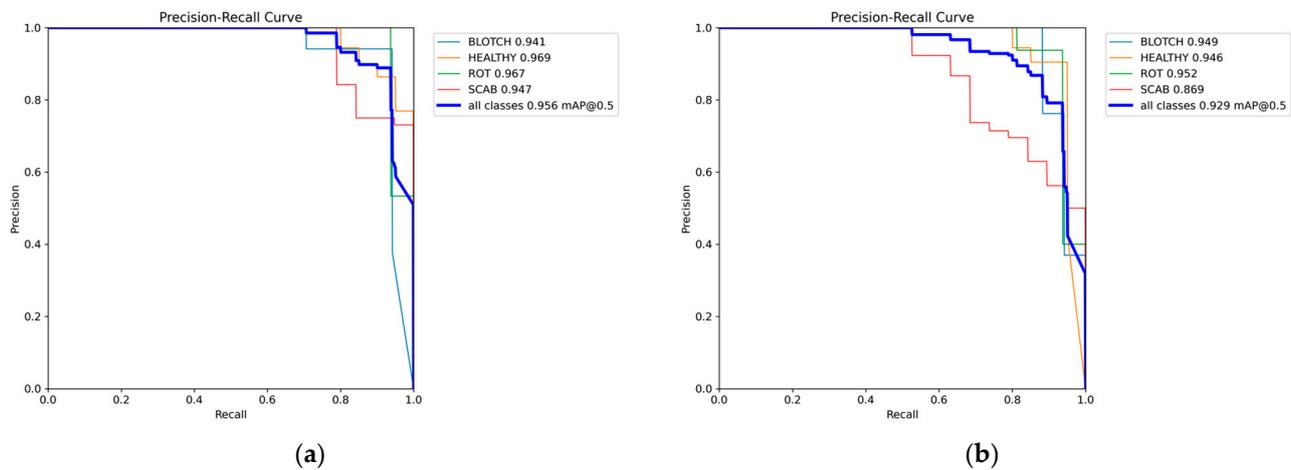


Figure 14. Comparison of P-R curve: (a) VEW-YOLOv8n P-R line graph; (b) YOLOv8n P-R line graph.

3.4. Ablation Experiments for Improved Processes

To ascertain the efficacy of the improvement methods, ablation experiments were conducted. These experiments assessed the contributions of three distinct improvement strategies and their combinations to model performance enhancement. The results are presented in Table 1. Experiment (1) evaluated the original YOLOv8n model. In Experiment (2), VanillaC2f was utilized to streamline YOLOv8n's backbone network, forming the Vanilla-Backbone lightweight network. This, along with deep training and an extended activation function, resulted in a 15% reduction in model size, a 14.6% decrease in parameters, and a 17.1% reduction in GFLOPS, while maintaining mAP values comparable to the baseline YOLOv8n model. Experiment (3) involved the use of a one-time aggregated cross-stage localized network module (VoVGSCSP) and the ghost shuffling module (GSConv) to optimize the neck network, creating the Efficient-Neck lightweight Neck network. This led to a 13.3% reduction in both model size and parameters, and, coupled with channel blending, slightly enhanced mAP compared to YOLOv8n. Experiment (4) improved model detection accuracy by substituting the original CIoU bounding-box loss function with the WIoU bounding-box loss function, resulting in mAP@0.5 values consistent with the baseline model and a slightly higher mAP@0.5:0.95. Experiment (5) combined the strategies from Experiments (2) and (3), integrating the Vanilla-Backbone and Efficient-Neck networks. This approach reduced the model size by 23.3%, parameters by 24.3%, and GFLOPS by 28.1%, while increasing the mAP@0.5 value by 1.3% compared to the baseline model. Experiment (6) combined the improvements of Experiments (3) and (4), combining the Vanilla-Backbone and Efficient-Neck networks with the WIoU bounding box loss function. This yielded a 2.7% increase in mAP@0.5 and a 1.2% increase in mAP@0.5:0.95 compared to the baseline model, without additional increases in parameters, model size, or GFLOPS.

Table 1. Ablation experiments for improved processes.

Algorithm	Precision/%	Recall/%	mAP@0.5/%	mAP@0.5:0.95/%	GFLOPS	Params/10 ⁶	Size/MB
YOLOv8n (1)	86.7	83.2	92.9	75.0	8.2	3.01	6.0
YOLOv8n + Vanilla-Backbone (2)	87.5	86.3	93.1	74.9	6.8	2.57	5.1
YOLOv8n + Efficient-Neck (3)	84.0	90.7	93.7	73.9	6.6	2.61	5.2
YOLOv8n + Vanilla-Backbone + WIoU (4)	90.6	86.5	93.9	75.8	6.7	2.57	5.1
YOLOv8n + Vanilla-Backbone + Efficient-Neck (5)	88.7	88.4	94.2	75.2	5.9	2.28	4.6
YOLOv8n + Vanilla-Backbone + EfficientNeck + WIoU (6)	89.0	89.5	95.6	76.2	5.9	2.28	4.6

3.5. Bounding Box Loss Function Side-by-Side Comparison Experiment

The effectiveness of the WIoU loss function for enhancing the bounding box loss function was evaluated through a cross-sectional comparison experiment. This experiment compared the performance of five bounding box loss functions—DIoU, SIoU, GIoU, CIoU, and WIoU—using the Vanilla-Backbone lightweight network, as delineated in Experiment (2) in Table 2. Results are presented in Table 2. The CIoU loss function served as the baseline for this comparison. It was observed that the model’s average accuracy decreased when the GIoU loss function was employed, suggesting that the prediction box and the actual box either shared an inclusion relationship or had overlapping dimensions, which were not suitable for the dataset in this study. The average accuracies of DIoU and SIoU were comparable to that of CIoU, nearly maintaining the same level. However, DIoU did not account for the aspect ratio of the bounding box, and SIoU lacked a strategy for gradient dynamic updating. While the GFLOPS slightly increased, the computational volume was larger. WIoU demonstrated the highest average accuracy, with mAP@0.5 reaching 93.9% and mAP@0.5:0.95 at 75.8%, surpassing the other bounding box loss functions in the experiment.

Table 2. Comparison of different bounding box loss functions.

Bounding Box Loss Functions	mAP@0.5/%	mAP@0.5:0.95/%	GFLOPS	Params/10 ⁶	Size/MB
+DIoU	93.0	74.6	6.7	2.57	5.1
+SIoU	93.4	75.2	6.8	2.57	5.1
+GIoU	91.4	73.8	6.7	2.57	5.1
+CIoU	93.1	74.9	6.8	2.57	5.1
+WIoU	93.9	75.8	6.7	2.57	5.1

Figure 15 illustrates the training process performance comparison for the five bounding box loss functions, indicating WIoU’s superior results.

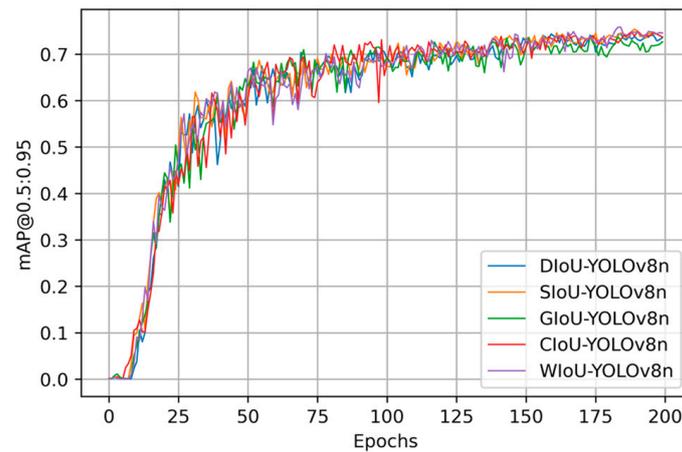


Figure 15. Comparison of the performance of the bounding box loss function.

4. Discussion

To ascertain the superior efficiency of the enhanced VEW-YOLOv8n algorithm, derived from YOLOv8n, it was compared with prevailing two-stage and single-stage target detection algorithms. The two-stage category included FasterRCNN, while the single-stage category included lightweight algorithms like YOLOv3-tiny [34], YOLOv5n [35], YOLOv6n [36], and YOLOv8n, along with high-precision, medium-large algorithms such as YOLOv8 m [37], and YOLOv3. Algorithms with larger convolutional kernels, namely YOLOv8n-InceptionNext and SSD target detection algorithms, were also compared. The YOLO-series algorithms used in these comparisons, similar to VEW-YOLOv8n, employed the same training strategy, incorporating methods like DFL, Anchor Free, Decoupled-Head, CiIoU, and TaskAlignedAssigner. The comparative performance evaluation results of VEW-YOLOv8n with mainstream and improved algorithms are displayed in Table 3 (two-stage target detection algorithm) and Table 4 (single-stage target detection algorithm). In Table 3, the two-stage models, based on migration learning and using pre-trained weights, recorded lower AP values compared to VEW-YOLOv8n, with a larger number of parameters and slower inference speeds. Table 4 shows the comparison among single-stage target detection algorithms. VEW-YOLOv8n stood out with the smallest model size (4.6 MB), least number of parameters (2.28×10^6), lowest GFLOPS (5.9), while achieving the highest average accuracy (mAP@0.5 at 93.9%, mAP@0.5:0.95 at 75.8%). Compared to high-precision, medium-large target detection algorithms, VEW-YOLOv8n attained similar average accuracy but with far fewer parameters, lower GFLOPS, and smaller model size. Against algorithms with large convolutional kernels, VEW-YOLOv8n surpassed YOLOv8n-InceptionNext in all aspects.

Table 3. Comparison of VEW-YOLOv8n with two-stage target detection algorithms.

Algorithm	mAP@0.5/%	mAP@0.5:0.95/%	GFLOPS	Size/MB	Speed/ms
SSD (Pretrained)	92.4	70.9	344.4	184.3	8.9
FasterRCNN (Pretrained)	91.5	70.5	206.68	315.1	25.8
VEW-YOLOv8n	95.6	76.2	5.9	4.6	7.5

The performance comparison of various YOLO family algorithms, including YOLOv3-tiny, YOLOv3-tiny, YOLOv5n, YOLOv6n, YOLOv8n, YOLOv8n, YOLOv8n, and YOLOv8n-InceptionNext, with VEW-YOLOv8n on two evaluation metrics, GFLOPS and mAP@0.5, is illustrated in Figure 16, highlighting VEW-YOLOv8n's minimal computational intensity and maximal average accuracy.

Table 4. Comparison of VEW-YOLOv8n with single-stage target detection algorithms.

Algorithm	mAP@0.5/%	mAP@0.5:0.95/%	GFLOPS	Params/10 ⁶	Size/MB	Speed/ms
YOLOv3	95.4	75.8	282.2	103.67	198.1	18.2
YOLOv3-tiny	91.1	68.9	19.1	12.1	23.2	5.1
YOLOv5n	92.1	74.1	7.2	2.51	5.3	7.3
YOLOv6n	91.9	73.5	11.8	4.23	8.7	6.3
YOLOv8n	92.9	75.0	8.2	3.01	6.0	8.2
YOLOv8n-InceptionNext	90.4	70.2	12.5	4.80	9.4	8.0
YOLOv8m	95.4	75.5	78.7	25.84	52.0	8.8
VEW-YOLOv8n	95.6	76.2	5.9	2.28	4.6	7.5

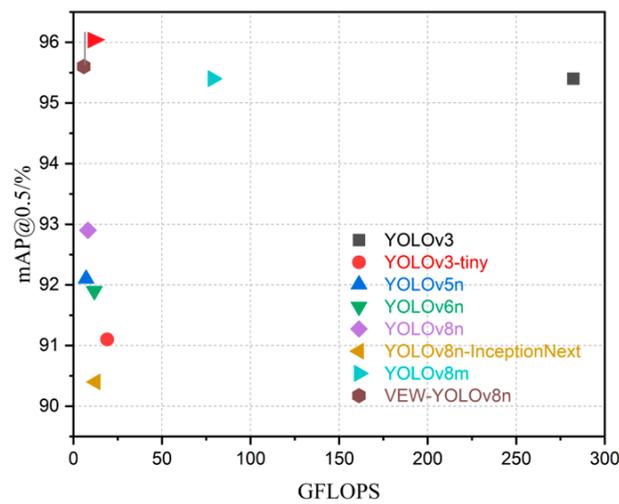


Figure 16. Performance comparison of VEW-YOLOv8n with various detection algorithms.

The effectiveness of VEW-YOLOv8n for the initial screening of bad apple fruits was further validated by comparing its performance with mainstream two-stage and single-stage target detection algorithms using the mAP@0.5:0.95 metric. VEW-YOLOv8n exhibited the highest values and efficacy, as demonstrated in Figure 17.

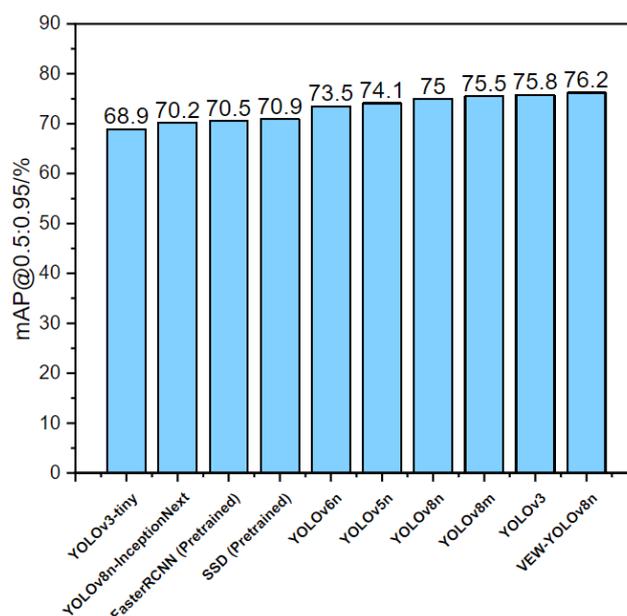


Figure 17. Performance comparison of VEW-YOLOv8n with various detection algorithms at mAP@0.5:0.95.

From three performance indicators—the number of parameters (params), computational volume (GFLOPS), and model size—the apple appearance grading detection algorithm (VEW-YOLOv8n) proposed in this study is confirmed as a low-complexity, lightweight target detection algorithm. Figure 18 shows VEW-YOLOv8n’s superiority with the smallest number of params, least computational volume, and smallest model size, rendering it suitable for edge-end deployment in initial screening and categorization of apples.

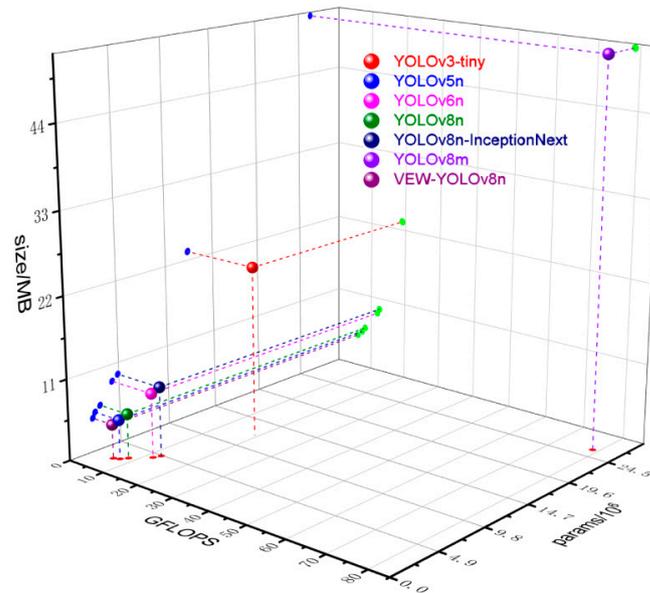


Figure 18. Comparison of VEW-YOLOv8n with single-stage target detection algorithm in terms of GFLOPS, params, and size performance indicators.

In order to have a more intuitive feeling of the detection performance of the VEW-YOLOv8n algorithm, standard test images of four apple types (HEALTHY, BLOTCH, ROT, and SCAB) were used. The detection effect of VEW-YOLOv8n was compared with YOLOv3-tiny, YOLOv5n, YOLOv6n, and YOLOv8n, and as depicted in Figure 19, the results indicate that VEW-YOLOv8n excels in detecting all four apple types. Specifically, in single-target detection, VEW-YOLOv8n demonstrates higher confidence than the other models. In multi-target detection, VEW-YOLOv8n outperforms the alternatives, with YOLOv5n, YOLOv6n, and YOLOv3-tiny exhibiting varying levels of detection omission. Analyzing Table 3, Figures 18 and 19 reveal that the VEW-YOLOv8n method offers low parameter count, minimal computational demand, high detection accuracy, and speed, meeting the real-time requirements of apple grading tasks.

The overall work is shown in Figure 20.

In the context of apple grading, the diseased fruits of the apple recognition process during primary screening utilizes images of multiple rolling apples, indicating a need for enhanced real-time recognition capabilities. Apples deemed healthy in the primary screening phase are directly channeled into the apple grading assembly line, conforming to the GBT10651–2008 [38] Fresh Apple standard, to ascertain their grade. Conversely, apples exhibiting defects such as rot, spots, and deformities, often resulting from diseases and pests, are promptly excluded from the grading process. Further research is planned to refine the apple grading task in subsequent stages. This research will be implemented and applied within the apple grading assembly line. Such advancements aim to automate and optimize the efficiency of the apple grading process, thereby reducing the time and cost associated with manual grading. This enhancement is anticipated to improve production efficiency, decrease production costs, and fulfill the demands of large-scale production and supply.

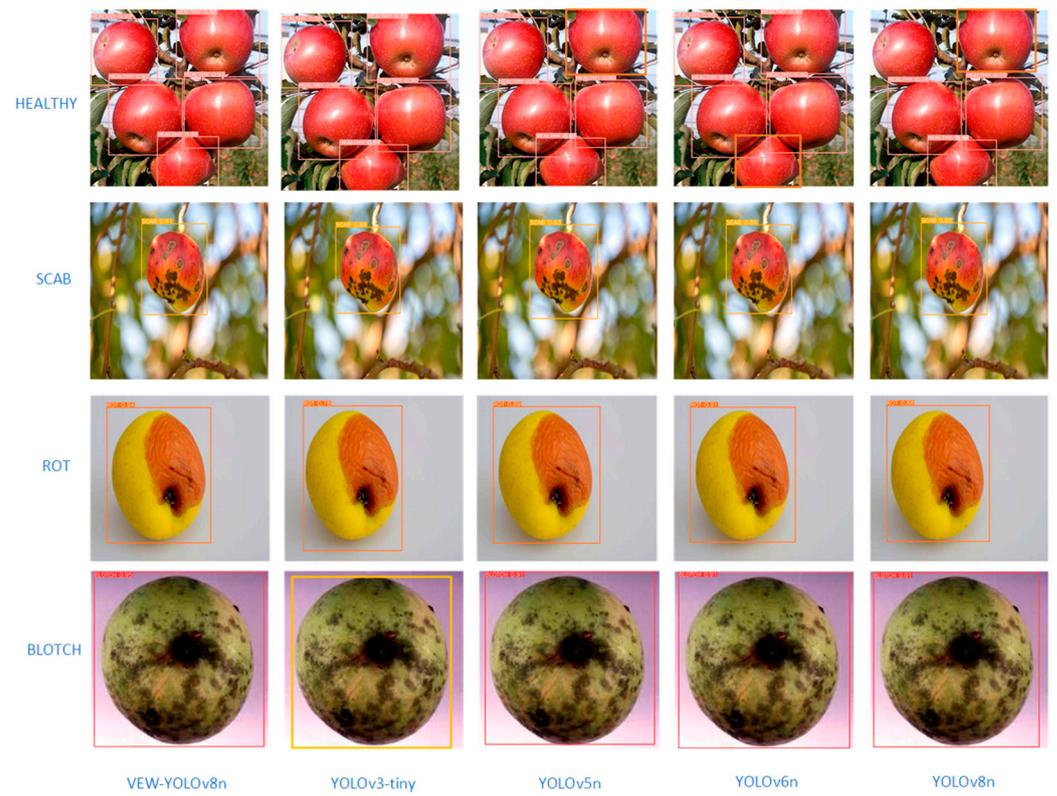


Figure 19. Comparison of detection results of VEW-YOLOv8n with various detection algorithms.

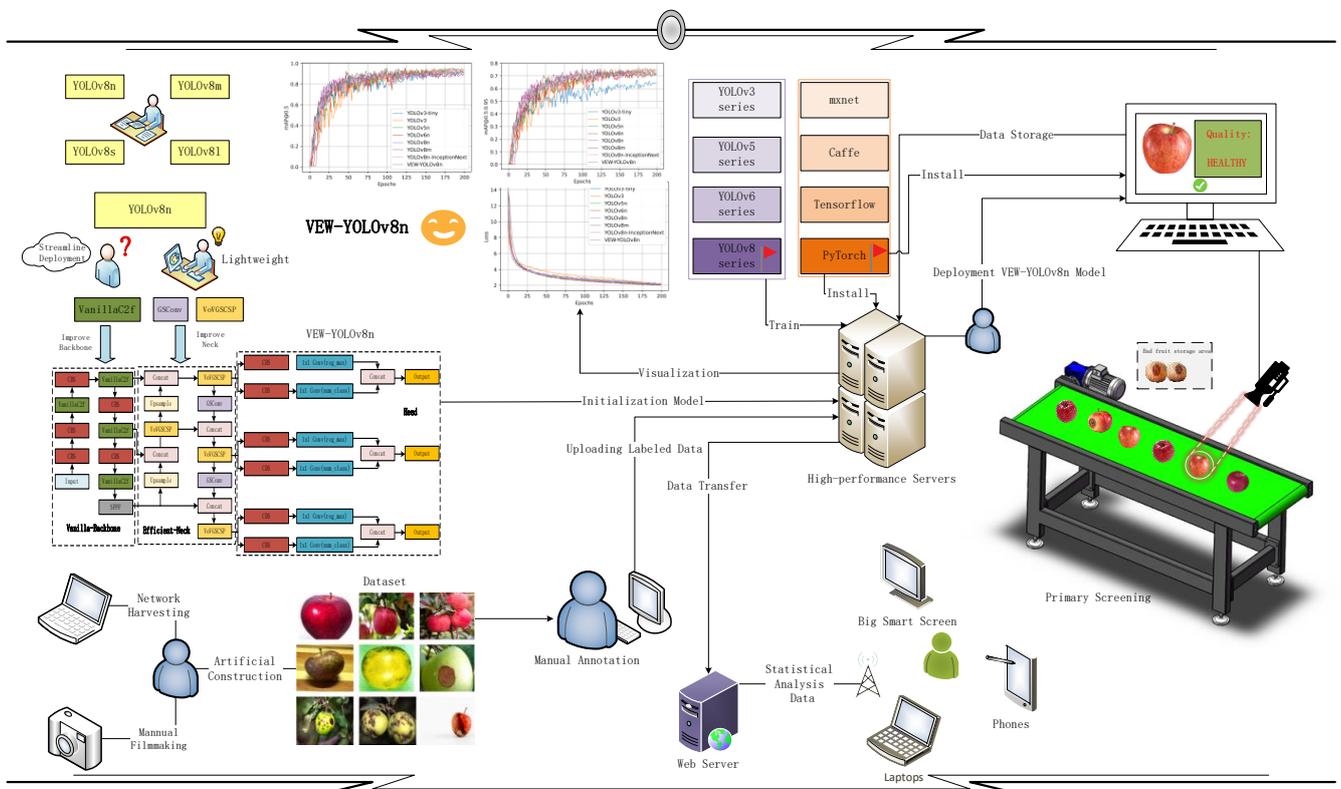


Figure 20. Overall work.

5. Conclusions

Considering the stability, real-time, and high-efficiency requirements of apple grading, the non-destructive detection of diseased fruits of apple adopts a lightweight detection approach to prevent grading loss. Consequently, this paper introduces the VEW-YOLOv8n apple grading detection algorithm. This algorithm involves a structural reparameterization of the VanillaC2f module, which lightens the backbone network and integrates an extended activation function to enhance the model's nonlinear expression capability. Additionally, the paper describes the development of an Efficient-Neck thin-neck structure, incorporating the lightweight GSConv module and VoVGSCSP module. This structure lightens the neck network, reduces the parameter count, and applies a channel mixing and washing strategy to efficiently extract feature information while elevating model detection accuracy. Furthermore, the implementation of the WIoU bounding-box loss function, combined with a strategy that diminishes the quality assessment index of outlier degree by using the competitiveness of high-quality frames, suppresses the adverse effects of low-quality frames. This approach effectively addresses the deviation issue of traditional IoU anchor frames and dynamically updates the gradient gain allocation strategy; thus, enhancing the model's robustness and generalization capability. The paper substantiates the efficacy of these algorithmic enhancements through ablation experiments and comparative experiments with prevalent target detection algorithms. The results demonstrate that the VEW-YOLOv8n model, compared to the YOLOv8n model, increases the average accuracy by 2.7%, reduces parameter count by 24.3%, decreases computation volume by 28.0%, shrinks model size by 23.3%, and boosts inference speed by 8.5%. Under the premise of maintaining accuracy, the VEW-YOLOv8n model achieves reductions in parameters and computation volume to various extents, markedly enhancing the model's lightweight effect, and more time can be set aside for the grading process. This offers a more advantageous identification method for the detection of diseased apples in apple grading. Further research will be carried out on the apple grading task and deployed and applied in the apple grading assembly line at a later stage.

Author Contributions: B.H.: Data collection, algorithm design, experiments, and writing; J.Z.: Supervision and writing review; Z.L.: Data collection and data organization; L.D.: investigation and validation. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Natural Science Foundation of Xinjiang Uygur Autonomous Region [2022D01A202]; Xinjiang Uygur Autonomous Region Colleges and Universities Research Program Project [XJEDU2020Y020]; Special Program for Central-Guided Local Science and Technology Development "Construction of Smart Agricultural Innovation Platform" [ZY2022B12].

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to privacy.

Acknowledgments: The authors are very grateful to the editor and reviewers for their valuable comments and suggestions to improve the paper.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Wang, X.; Zhao, Z.; Wang, J. Measurement of concentration of apple production in China's main production areas and analysis of their competitiveness. *J. Hebei Agric. Sci.* **2023**, *27*, 83–86.
2. Chen, Q.; Yin, C.; Guo, Z.; Wang, J.; Zhou, H.; Jiang, X. Current status and future development of the key technologies for apple picking robots. *Trans. Chin. Soc. Agric. Eng.* **2023**, *39*, 1–15.
3. Ma, F.; Liu, F.; Li, W. Jet tagging algorithm of graph network with Haar pooling message passing. *Phys. Rev. D* **2023**, *108*, 072007. [[CrossRef](#)]

4. Ji, P.; Feng, J.; Ma, F.; Wang, X.; Li, C. Fingertip Detection Algorithm Based on Maximum Discrimination HOG Feature in Complex Background. *IEEE Access* **2023**, *11*, 3160–3173. [[CrossRef](#)]
5. Dong, Y.; Guo, B. Railway track detection algorithm based on Hu invariant moment feature. *J. China Railw. Soc.* **2018**, *40*, 64–70. [[CrossRef](#)]
6. Madake, J.; Raje, A.; Rajurkar, S.; Rakhe, R.; Bhatlawande, S.; Shilaskar, S. Vision-based distracted driver detection using a fusion of SIFT and ORB feature extraction. In Proceedings of the International Conference on Security, Privacy and Data Analytics, Surat, India, 13–15 December 2022; Springer: Singapore, 2022; pp. 163–178.
7. ÇİLtaŞ, Y.; Funda, A. Copy move forgery detection with SURF and MSER combination. *Niğde Ömer Halisdemir Univ. Muhendis. Bilim. Derg.* **2022**, *11*, 513–521. [[CrossRef](#)]
8. Zeng, J.; Chen, X. Pedestrian detection combined with single and couple pedestrian DPM models in traffic scene. *Acta Electronica Sin.* **2016**, *44*, 2668–2675. [[CrossRef](#)]
9. Gu, B.; Wen, C.; Liu, X.; Hou, Y.; Hu, Y.; Su, H. Improved YOLOv7-Tiny Complex Environment Citrus Detection Based on Lightweighting. *Agronomy* **2023**, *13*, 2667. [[CrossRef](#)]
10. Ren, R.; Sun, H.; Zhang, S.; Wang, N.; Lu, X.; Jing, J.; Xin, M.; Cui, T. Intelligent Detection of Lightweight “Yuluxiang” Pear in Non-Structural Environment Based on YOLO-GEW. *Agronomy* **2023**, *13*, 2418. [[CrossRef](#)]
11. Lyu, S.; Zhao, Y.; Liu, X.; Li, Z.; Wang, C.; Shen, J. Detection of Male and Female Litchi Flowers Using YOLO-HPFD Multi-Teacher Feature Distillation and FPGA-Embedded Platform. *Agronomy* **2023**, *13*, 987. [[CrossRef](#)]
12. Zhang, L.; Zhou, S.; Li, N.; Zhang, Y.; Chen, G.; Gao, X. Apple Location and Classification Based on Improved SSD Convolutional Neural Network. *Trans. Chin. Soc. Agric. Mach.* **2023**, *54*, 223–232.
13. Tian, L.; Zhang, H.; Liu, B.; Zhang, J.; Duan, N.; Yuan, A.Y.; Huo, Y. VMF-SSD: A Novel V-Space based Multi-scale Feature Fusion SSD for Apple Leaf Disease Detection. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2022**, *20*, 2016–2028. [[CrossRef](#)]
14. Hu, G.; Zhang, E.; Zhou, J.; Zhao, J.; Gao, Z.; Sugirbay, A.; Jin, H.; Zhang, S.; Chen, J. Infield apple detection and grading based on multi-feature fusion. *Horticulturae* **2021**, *7*, 276. [[CrossRef](#)]
15. Su, H.; Ma, S. Study on the stability of high and steep slopes under deep bench blasting vibration in open-pit mines. *Front. Earth Sci.* **2022**, *10*, 990012. [[CrossRef](#)]
16. Liu, Y.; Liu, X.; Zhang, B. RetinaNet-vline: A flexible small target detection algorithm for efficient aggregation of information. *Clust. Comput.* **2023**, 1–13. [[CrossRef](#)]
17. Liu, S.; Fu, S.; Hu, A.; Ma, P.; Hu, X.; Tian, X.; Zhang, H.; Liu, S. Research on Insect Pest Identification in Rice Canopy Based on GA-Mask R-CNN. *Agronomy* **2023**, *13*, 2155. [[CrossRef](#)]
18. Zhang, X.; Wang, C.; Jin, J.; Huang, L. Object detection of VisDrone by stronger feature extraction FasterRCNN. *J. Electron. Imaging* **2023**, *32*, 013018. [[CrossRef](#)]
19. Fan, S.; Liang, X.; Huang, W.; Zhang, V.J.; Pang, Q.; He, X.; Li, L.; Zhang, C. Real-time defects detection for apple sorting using NIR cameras with pruning-based YOLOV4 network. *Comput. Electron. Agric.* **2022**, *193*, 106715. [[CrossRef](#)]
20. Sun, J.; Qian, L.; Zhu, W.; Zhou, X.; Dai, C.; Wu, X. Apple detection in complex orchard environment based on improved RetinaNet. *Trans. Chin. Soc. Agric. Eng.* **2022**, *38*, 314–322.
21. Zhang, C.; Kang, F.; Wang, Y. An improved apple object detection method based on lightweight YOLOv4 in complex backgrounds. *Remote Sens.* **2022**, *14*, 4150. [[CrossRef](#)]
22. Li, H.; Li, J.; Wei, H.; Liu, Z.; Zhan, Z.; Ren, Q. Slim-neck by GSConv: A better design paradigm of detector architectures for autonomous vehicles. *arXiv* **2022**, arXiv:2206.02424. [[CrossRef](#)]
23. Tong, Z.; Chen, Y.; Xu, Z.; Yu, R. Wise-IoU: Bounding Box Regression Loss with Dynamic Focusing Mechanism. *arXiv* **2023**, arXiv:2301.10051. [[CrossRef](#)]
24. Aboah, A.; Wang, B.; Bagci, U.; Adu-Gyamfi, Y. Real-time multi-class helmet violation detection using few-shot data sampling technique and yolov8. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 5349–5357.
25. Ou, J.; Zhang, R.; Li, X.; Lin, G. Research and Explainable Analysis of a Real-Time Passion Fruit Detection Model Based on FSOne-YOLOv7. *Agronomy* **2023**, *13*, 1993. [[CrossRef](#)]
26. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. Yolox: Exceeding yolo series in 2021. *arXiv* **2021**, arXiv:2107.08430. [[CrossRef](#)]
27. Jadon, S. A survey of loss functions for semantic segmentation. In Proceedings of the 2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB), Via del Mar, Chile, 27–29 October 2020; pp. 1–7.
28. Feng, C.; Zhong, Y.; Gao, Y.; Scott, M.R.; Huang, W. Tood: Task-aligned one-stage object detection. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, BC, Canada, 11–17 October 2021; pp. 3490–3499.
29. Cheng, G.; Wang, J.; Li, K.; Xie, X.; Lang, C.; Yao, Y.; Han, J. Anchor-free oriented proposal generator for object detection. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5625411. [[CrossRef](#)]
30. Chen, H.; Wang, Y.; Guo, J.; Tao, D. VanillaNet: The Power of Minimalism in Deep Learning. *arXiv* **2023**, arXiv:2305.12972. [[CrossRef](#)]
31. Rezatofighi, H.; Tsoi, N.; Gwak, J.; Sadeghian, A.; Reid, I.; Savarese, S. Generalized intersection over union: A metric and a loss for bounding box regression. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2019, Long Beach, CA, USA, 16–20 June 2019; pp. 658–666.

32. Chen, D.; Miao, D. Control Distance IoU and Control Distance IoU Loss for Better Bounding Box Regression. *Pattern Recogn.* **2023**, *137*, 109256. [[CrossRef](#)]
33. Gevorgyan, Z. SIoU loss: More powerful learning for bounding box regression. *arXiv* **2022**, arXiv:2205.12740. [[CrossRef](#)]
34. Fu, L.; Feng, Y.; Wu, J.; Liu, Z.; Gao, F.; Majeed, Y.; Al-Mallahi, A.; Zhang, Q.; Li, R.; Cui, Y. Fast and accurate detection of kiwifruit in orchard using improved YOLOv3-tiny model. *Precis. Agric.* **2021**, *22*, 754–776. [[CrossRef](#)]
35. Sun, Y.; Zhang, D.; Guo, X.; Yang, H. Lightweight Algorithm for Apple Detection Based on an Improved YOLOv5 Model. *Plants* **2023**, *12*, 3032. [[CrossRef](#)]
36. Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W. YOLOv6: A single-stage object detection framework for industrial applications. *arXiv* **2022**, arXiv:2209.02976. [[CrossRef](#)]
37. Ranjan Sapkota, D.A.; Churuvija, M.; Karkee, M. Immature green apple detection and sizing in commercial orchards using YOLOv8 and shape fitting techniques. *arXiv* **2023**, arXiv:2401.08629.
38. GB/T 10651-2008; Fresh Apple. All China Federation of Supply and Marketing Cooperatives: Beijing, China, 2008.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.