electronics

MDPI

Article

HR-YOLOv8: A Crop Growth Status Object Detection Method Based on YOLOv8

Jin Zhang [1], Wenzhong Yang [2,3,*], Zhifeng Lu [4,*] and Danny Chen [2]

[1] School of Software, Xinjiang University, Urumqi 830017, China; 107552204844@stu.xju.edu.cn
[2] School of Computer Science and Technology, Xinjiang University, Urumqi 830017, China; kabuoxygen@stu.xju.edu.cn
[3] Xinjiang Key Laboratory of Multilingual Information Technology, Xinjiang University, Urumqi 830017, China
[4] School of Information Science and Technology, Xinjiang Teacher's College, Urumgi 830043, China
* Correspondence: yangwenzhong@xju.edu.cn (W.Y.); xjdxsylb@xju.edu.cn (Z.L.)

Abstract: Crop growth status detection is significant in agriculture and is vital in planting planning, crop yield, and reducing the consumption of fertilizers and workforce. However, little attention has been paid to detecting the growth status of each crop. Accuracy remains a challenging problem due to the small size of individual targets in the image. This paper proposes an object detection model, HR-YOLOv8, where HR means High-Resolution, based on a self-attention mechanism to alleviate the above problem. First, we add a new dual self-attention mechanism to the backbone network of YOLOv8 to improve the model's attention to small targets. Second, we use InnerShape(IS)-IoU as the bounding box regression loss, computed by focusing on the shape and size of the bounding box itself. Finally, we modify the feature fusion part by connecting the convolution streams from high resolution to low resolution in parallel instead of in series. As a result, our method can maintain a high resolution in the feature fusion part rather than recovering high resolution from low resolution, and the learned representation is more spatially accurate. Repeated multiresolution fusion improves the high-resolution representation with the help of the low-resolution representation. Our proposed HR-YOLOv8 model improves the detection performance on crop growth states. The experimental results show that on the oilpalmuav dataset and strawberry ripeness dataset, our model has fewer parameters compared to the baseline model, and the average detection accuracy is 5.2% and 0.6% higher than the baseline model, respectively. Our model's overall performance is much better than other mainstream models. The proposed method effectively improves the ability to detect small objects.

Keywords: crop growth; object detection; DNN; YOLO

Citation: Zhang, J.; Yang, W.; Lu, Z.; Chen, D. HR-YOLOv8: A Crop Growth Status Object Detection Method Based on YOLOv8. *Electronics* **2024**, *13*, 1620. https://doi.org/10.3390/electronics13091620

Academic Editor: George A. Papakostas

Received: 20 March 2024
Revised: 16 April 2024
Accepted: 22 April 2024
Published: 24 April 2024

1. Introduction

With the rapid development of technology, smart agriculture has been an essential part of the intelligent economy. The growth posture of crops is related to the whole process of agricultural production, so agricultural production must carry out reasonable crop-growth-state detection through crop pictures of different periods. By analyzing the growth of crops, we can maximize the judgment of crop growth statuses, rationally deploy production resources, provide timely and reliable growth information for crop production managers or management decision-makers, facilitate the timely collection of effective field management measures, and help farmers, agricultural professionals, and government agencies to make more informed decisions. By monitoring and recognizing the growth status of crops, more precise agricultural management can be carried out. Examples include proper fertilizer application, irrigation, and pest control to maximize yields and minimize resource waste. Understanding the growth status of crops helps to develop more effective production plans and make agricultural production more efficient. By recognizing the growth stage of plants, the best time to harvest can be determined, improving the quality and marketability

of crops. Accurate growth status identification can help farmers better use resources, such as water and fertilizers. Avoiding unnecessary resource waste reduces the cost of agricultural production. The significance of the crop-growth-state recognition task is to improve agricultural production's efficiency, sustainability, and adaptability.

Modern computer vision and machine learning techniques are widely used for crop-growth-state recognition. Computer vision technology can realize the recognition of crop growth states by processing and analyzing farmland images. The traditional method uses manual observation, in which farmers and professionals determine the growth state by directly observing plants in the farmland and judging the growth state based on the appearance, color, morphology, and other characteristics of the plants. It is time-consuming, labor-intensive, and expensive, and due to different experiences, different people may obtain different results from observing the same piece of farmland.

This includes image segmentation, feature extraction, and pattern recognition methods. Deep learning techniques have made significant achievements in image recognition tasks and can be used to automate the identification of crop growth. Convolution neural network features such as color, texture, and shape can be extracted for crop-growth-state recognition. These features help distinguish plants at different growth stages.

Object detection plays a vital role in computer vision. By harnessing the power of deep convolution neural networks, models can learn complex feature representations from images, enabling them to detect and localize objects accurately. Object detection is widely used in smart agriculture, autonomous driving, and industrial robotics. Significant progress has been made in crop-growth-state recognition in recent years. In oil palm-growth-state detection, Zheng et al. [1] proposed a network model based on Faster-RCNN to detect oil palm growth state. It proposed a refined pyramid feature (RPF) module for feature extraction. RDF integrates deep and shallow features to obtain more features from different hierarchical feature maps. Yu et al. [2] investigated two imaging-based automatic detection methods for two critical growth stages of maize using AP-HI. Zhang et al. [3], based on the YOLOv5s, proposed ESPA-YOLOv5s for detecting the growth status of cotton and rape seeds. Yang et al. [4] improved the YOLOv5 model to recognize and monitor the growth status of apple blossoms. They enhanced the model's ability to recognize the growth status of apple blossoms by improving the backbone and feature fusion networks. In contrast, due to its superior detection speed, the YOLO series has a significant advantage in crop-growth-state detection applications.

Despite the rapid development in technology, recognizing and monitoring the growth status of crops are still challenging. As UAVs capture farmland scenes at different heights, the image target scale varies widely and has a high proportion of small targets. For example, the high density and blurred pixels as well as the targets are easily affected by image noise, which brings great challenges in the identification of network optimization and crop localization; moreover, it often leads to detector insensitivity to target objects, and the IoU in the loss function is affected by small target bias in the training of the model. YOLOv8 [5] represents a single-stage detection model with fast training speed and high reliability, but the design of its backbone network and feature fusion network can lead to the loss of some small target feature information. YOLOv8 also has a large number of parameters and operations, which is not suitable for deployment on terminals in agriculture. To alleviate the above problems, we propose an improved model based on YOLOv8, which improves the baseline model from several perspectives, including the design of the feature pyramid HR-FPN, the introduction of a new two-channel self-attention mechanism, the Dual Channel High-Resolution Attention (DCHRA), and the loss function InnerShape(IS)-IoU. Evaluating the model performance in the oilpalmuav [1] dataset and strawberry ripeness dataset, HR-YOLOv8 achieves a good performance in terms of accuracy, recall, and average precision, achieving significant improvement. The contributions of this paper are as follows:

1. We propose a feature fusion module called HR-FPN that references the network structure of HR-Net [6], which connects high-resolution and low-resolution convolution

streams in parallel rather than in series. Our approach maintains high resolution within the module, avoiding the information loss associated with transitioning from low to high resolution. As a result, the representation learned by HR-FPN achieves less loss of information compared to previous feature fusion modules. In addition, we repeated the feature fusion at different resolutions, using the low-resolution representation to improve the high-resolution representation.

2. We added a new dual self-attention mechanism called DCHRA to the YOLOv8 backbone network, including spatial and channel attention branches. The potential loss of high-resolution information in deep neural networks is reduced by pooling/downsampling. Compared to existing attention blocks, DCHRA maintains a high internal resolution inside the attention block. To fit the output distribution of typical fine-grained regression, Softmax–Sigmoid combinations are fused in channel-only and spatial-only attention branches to improve the model's attention to small targets.
3. For the problem of uneven distribution of categories in the dataset, considering the influence of the shape and scale of the bounding box regression samples themselves on the bounding box regression and using auxiliary borders of different sizes for different sizes of IoU samples, we design IS-IoU as a loss function by combining Inner-IoU [7] and Shape-IoU [8] as the bounding box regression loss.

2. Related Work

2.1. Object Detection

The YOLO family of models is widely used in academia and industry for its speed, accuracy, and ease of deployment. Since the release of the YOLO model, many improvements have occurred in the YOLO family of models over the years. YOLOv1–v3 [9–11] constructed the initial YOLO, which consists of a three-part detection structure consisting of a torso, a neck, and a head. YOLOv1–v3 is a single-stage object detection model designed to detect objects of different sizes. It becomes a representative single-stage object detection model by predicting objects of different sizes through multiscale branching. YOLOv4 [12] proposes the Mish activation function on the activation function and PANet on the feature fusion part. Not only that, YOLOv4 also proposes a series of data enhancement methods to improve the DrakNet structure, which provides a template for the subsequent YOLO models. YOLOv5 [13] improved the data enhancement strategies, such as the mosaic algorithm, and proposed various model variants based on YOLOv4. YOLOv5 is still active in academia and industry until now due to its excellent structure and easy modification. YOLOX [14] sets a good example for the YOLO model design by incorporating the multi-positive, unanchored, and decoupled headers into the model structure. YOLOv6 [15,16] proposed the EfficientRep Backbone and Rep-PAN Neck based on previous models and applied the reparameterization method. Wang et al. presented YOLOv7 [17]. E-ELAN in the YOLOv7 backbone network is an efficient layer aggregation network. It can also learn more diverse features by directing the computational blocks of different feature groups. Meanwhile, a new scaling strategy based on the tandem model is proposed, in which the same factor scales the depth and width of the blocks to maintain the optimal structure of the model. YOLOv8 [5] is the latest YOLO series algorithm at present. It integrates the points of the previous YOLO series and is currently the most powerful YOLO detector.

2.2. Attention

Recently, attention mechanisms have been introduced to many visual tasks [18–22]. Self-attention [18,23,24] has become a standard component for capturing remote interactions after success in sequential and generative modeling tasks. For object detection, the Squeeze-and-Excitation Network (SENet) [25] and Convolution Block Attention Module (CBAM) [26] show the most traditional performance.

The advantage of CBAM is that it can implement both channel-wise and spatially attentional techniques, which enhance the networks by making them more attentive to critical features and suppressing non-critical features, thus enhancing the network's repre-

sentation and generalization capabilities. However, CBAM also has significant drawbacks, one of which is the need for more spatial information at all levels, which prevents it from fully utilizing the properties of the attention space and only considers data within a local region, thus failing to identify long-range dependencies.

2.3. Multiscale Object Detection

In object detection, feature maps in different layers of the backbone network have semantic information about targets of different sizes. Initial object detection networks tend to use only one of the feature maps or directly use several of them, thus often leading to loss of information about the detected target. Lin et al. [27] proposed a feature pyramid network for object detection. The proposed FPN improves the network's performance and reduces semantic loss by fusing feature maps of different sizes.

Although FPN uses deep semantic features to construct each layer of the pyramid, the shallow features are lost as the network deepens, resulting in a lack of spatial information in the feature layer at the top of the pyramid, which is not conducive to accurate target localization. Assigning a target to a layer of the pyramid for prediction based on the scale of the target is not optimal because it does not take into account the valuable information that may be present in other feature layers. This strategy is not optimal because it needs to consider the valuable information that may exist in other feature layers. PANet [28] starts from the bottom of the feature pyramid already constructed by FPN and adds a bottom-up feature re-fusion side path to reconstruct a new pyramid with enhanced spatial information. Guo et al. proposed a solution to upgrade FPN to AugFPN [29] in response to the defects of FPN: the semantic information of different feature layers is inconsistent, and FPN will directly add them through 1×1 convolution, so AugFPN will be trained according to the fusion of the features before the detection of the loss is calculated, and then weighted and summed with the loss of the network to provide a supervised signal. Therefore, AugFPN is trained to detect and compute the loss directly based on the fused features, which are then weighted and summed with the network's loss to provide a supervised signal. However, the 1×1 convolution process will lead to losing the highest-level feature information. Therefore, AugFPN performs spatial pyramid pooling operation on the original highest-level feature maps and then fuses them with the highest-level feature maps after 1×1 dimensionality reduction after adaptive feature fusion of all the branches to make up for the loss of information; each instance of FPN is based on the scale heuristic. Each instance in FPN selects the feature layer heuristically according to the scale, but other layers may also have important feature information. Therefore, AugFPN extracts the features of each instance on each layer of the pyramid, lets the network learn the weight parameters, sums up these features, and finally predicts the target. Tan et al. proposed EfficientDet [30], which makes several changes based on the feature pyramid architecture of PANet. They removed nodes with only one input because they were not fused with features and contributed less to the multiscale features. They added a link between the input and output feature maps at the same scale to fuse richer features. They stacked the entire feature pyramid architecture multiple times to make it more potent in feature representation. Finally, the BiFPN architecture is obtained.

On the other hand, Ghaisi et al. [31], used a neural network architecture search to find a better construction scheme: a recurrent neural network was used as a controller, trained with reinforcement learning, and allowed the authors to decide which two feature layers to choose each time and in what way to fuse them and output them at a specific resolution until each layer of the feature pyramid was filled. The algorithm ultimately searches the resulting NAS-FPN.

3. Method

YOLO series models are the most mainstream application detection models in one-stage detection algorithms, and they are mainly optimized to balance detection accuracy. YOLOv8 is one of the best YOLO series models at present. The backbone network and neck network

refer to the YOLOv7 ELAN design idea, replacing the C3 structure of YOLOv5 with the C2f structure, which is richer in gradient flow, and adjusts the different number of channels for different scale models. The head network has been changed to the current mainstream decoupled-head compared to the previous YOLO series models, which separates the classification and detection heads. The head network has also been changed from anchor-based to anchor-free. We have made some improvements to YOLOv8 based on this, as shown in Figure 1.

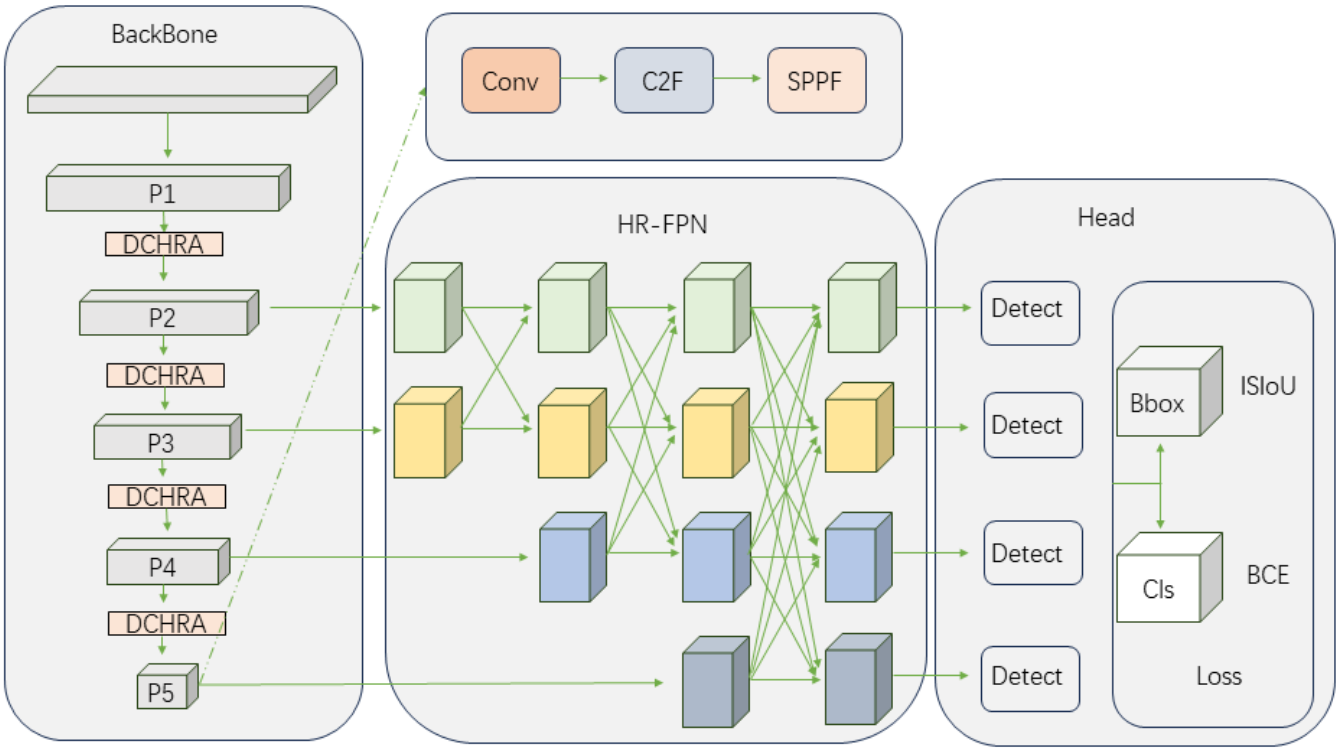

Figure 1. The architecture of the proposed HR-YOLOv8. It includes backcone, DCHRA, HR-FPN, and detection head.

Images after preprocessing: the image size is scaled to a size not exceeding 640 × 640 input to the backbone part of the feature extraction. In this paper, we use YOLOv8's backbone, as shown in Figure 2, which consists of a C2F convolution module, Conv module, and SPPF composed of the backbone network. The backbone network is an essential part of the object detector, and the primary role is feature extraction. The C2F block of the backbone network is composed of a convolutional composition layer, a batch normalization layer and a SiLU activation function. YOLOv8 is designed with a structure that enhances the learning ability and robustness of the network. SPPF block utilizes spatial pyramid pooling and CSP structure for more efficient gradient combination.

3.1. HR-FPN

We propose a feature pyramid network called HR-FPN, similar to other object detection methods, where we extract feature maps of different sizes in different layers of the backbone network for fusion. In addition, our approach is similar to the YOLOv8 framework in that the last layer of features is extracted from each feature layer of the backbone network, resulting in a set of features with different scales, denoted as C2, C3, C4, C5. We first input the larger-scale features C2 and C3 into the feature pyramid network. C2 is downsampled and added to C3 to obtain it, and C3 is upsampled and added to C2, followed by a convolution layer, and is kept constant in size. Subsequently, C4 is added, the same operation is performed as in the previous network layer, and C5 is added.

After the feature fusion step, a set of multiscale features P2, P3, P4, P5 will be generated. Conventional convolution neural networks (e.g., low-resolution networks) work in series and thus recover high resolution from low resolution. The parallel approach of HR-FPN allows the maintenance of high resolution through the entire neural network and, thus, a more accurate representation.

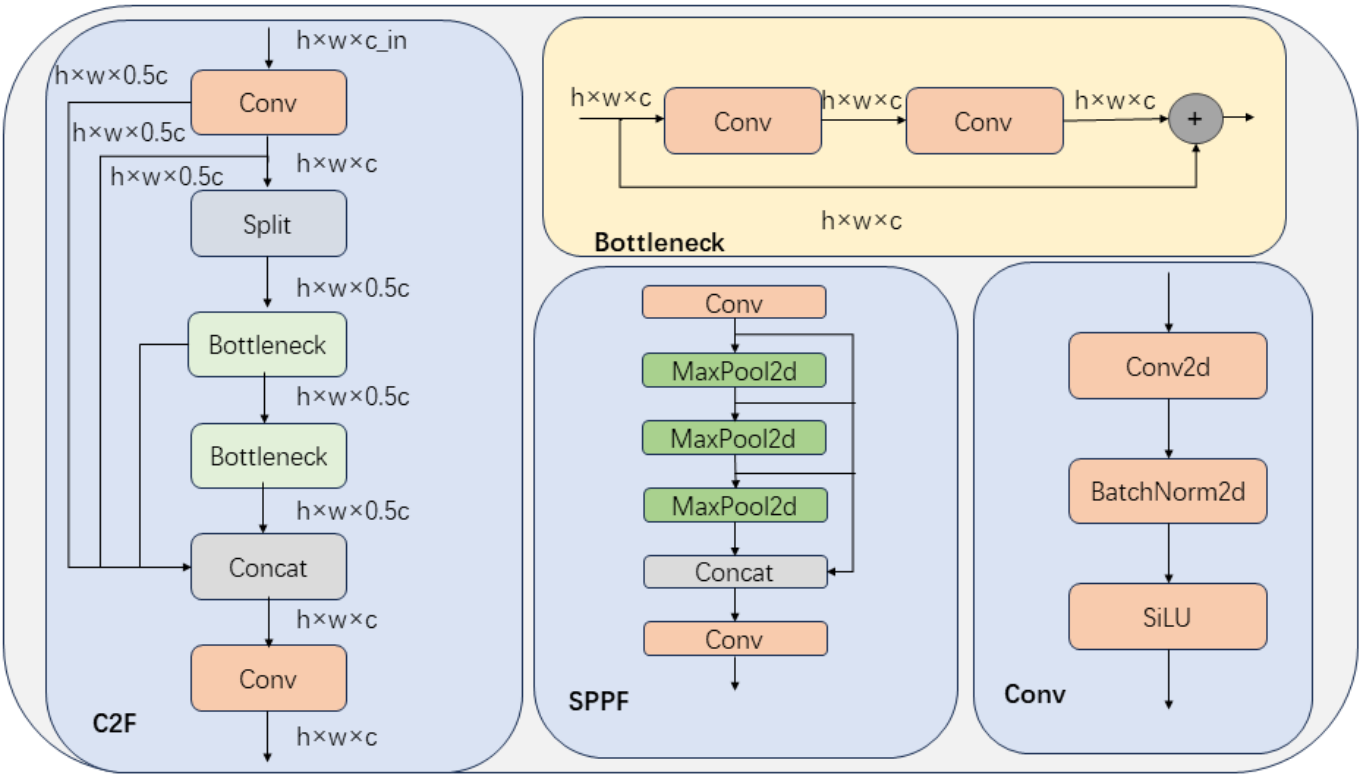

Figure 2. Specific details of C2F, bottleneck, SPPF, and Conv blocks in the proposed network.

The architecture of our proposed HR-FPN is shown in Figure 3. Since the semantic gap between non-adjacent-level feature maps is larger than the gap between neighboring feature maps, especially for the top- and bottom-level feature maps, i.e., C2 and C5, direct fusion will lead to poor fusion due to the sizeable semantic gap. Therefore, to avoid this problem, we designed the structure of HR-FPN to be asymptotic, allowing neighboring feature maps to be fused before using the fused feature maps with the feature maps of other layers. Using this structure can make the fusion of semantic features from different layers closer.

To align the dimensions and prepare the features for fusion, we upsample the features using 1×1 convolution and bilinear interpolation methods. We perform downsampling using a 3×3 convolution with a step size of 2, depending on the desired downsampling rate. For example, we apply one 3×3 convolution with a step size of 2 for the feature fusion of C2 and C3, while for C5 and C2, we need to apply three 3×3 convolutions with a step size of 2 for downsampling.

3.2. DCHRA

We designed a new dual self-attention mechanism called DCHRA, as shown in Figure 4, including a spatial attention branch and a channel attention branch, to highlight or suppress features and maintain high resolution inside the module by pooling/downsampling to reduce the potential loss of high-resolution information in deep neural networks. We design DCHRA to completely collapse features in one dimension while maintaining high resolution in the direction of the other dimension, e.g., in the channel attention mechanism, the number of channels in one of the branches is compressed to 1. At the same time, the

width and height remain unchanged. Furthermore, the dynamic range of the attention is increased by Softmax normalization on the feature tensor, which is then mapped using Sigmoid functions.

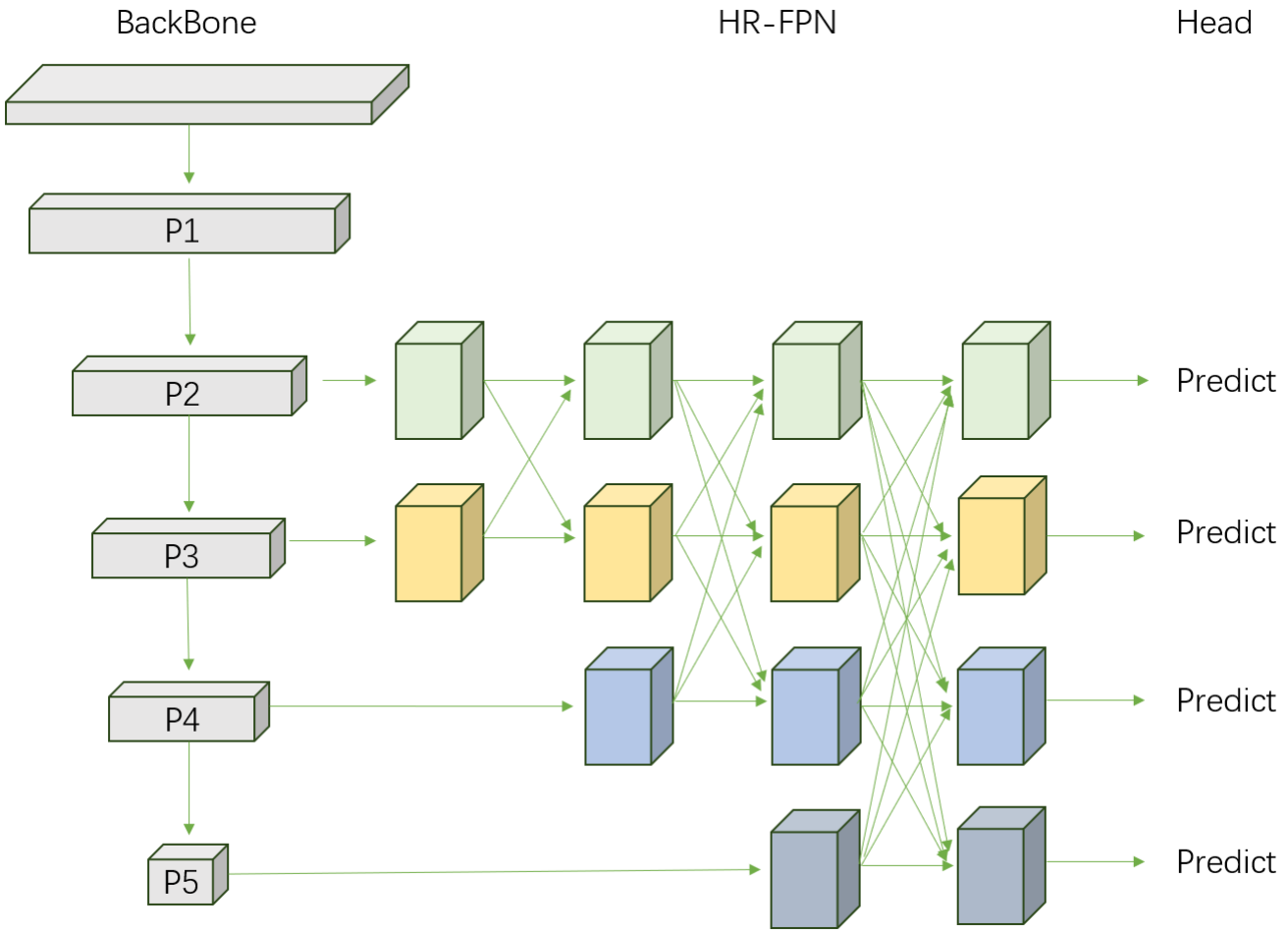

Figure 3. The architecture of the proposed HR-FPN. HR-FPN fuses two low-level features in the initial stage. The subsequent stage fuses higher-level features, while the final stage adds top-level features to the feature fusion process.

Compared to other attention methods, the self-attention mechanism proposed in this paper is not compressed to a great extent in both the spatial and channel dimensions, making the information loss relatively small. For the channel attention mechanism, the input features are first converted into two parts, q and v, using a 1×1 convolution, where the channel of q is completely compressed, and the channel dimension of v remains relatively high. The information of q was augmented using Softmax. The two parts were then matrix-multiplied and followed by a 1×1 convolution, LN, and the dimensions on the channels were raised to their original dimensions. Finally, a Sigmoid function was used so that all the parameters were kept between 0 and 1, and then the output was dot-multiplied with the initial input to obtain the result. For the spatial attention mechanism, similar to the channel attention mechanism, the input features are first converted into two parts, q and v, using a 1×1 convolution, where the spatial dimensionality compression for q is converted to a size of 1×1.

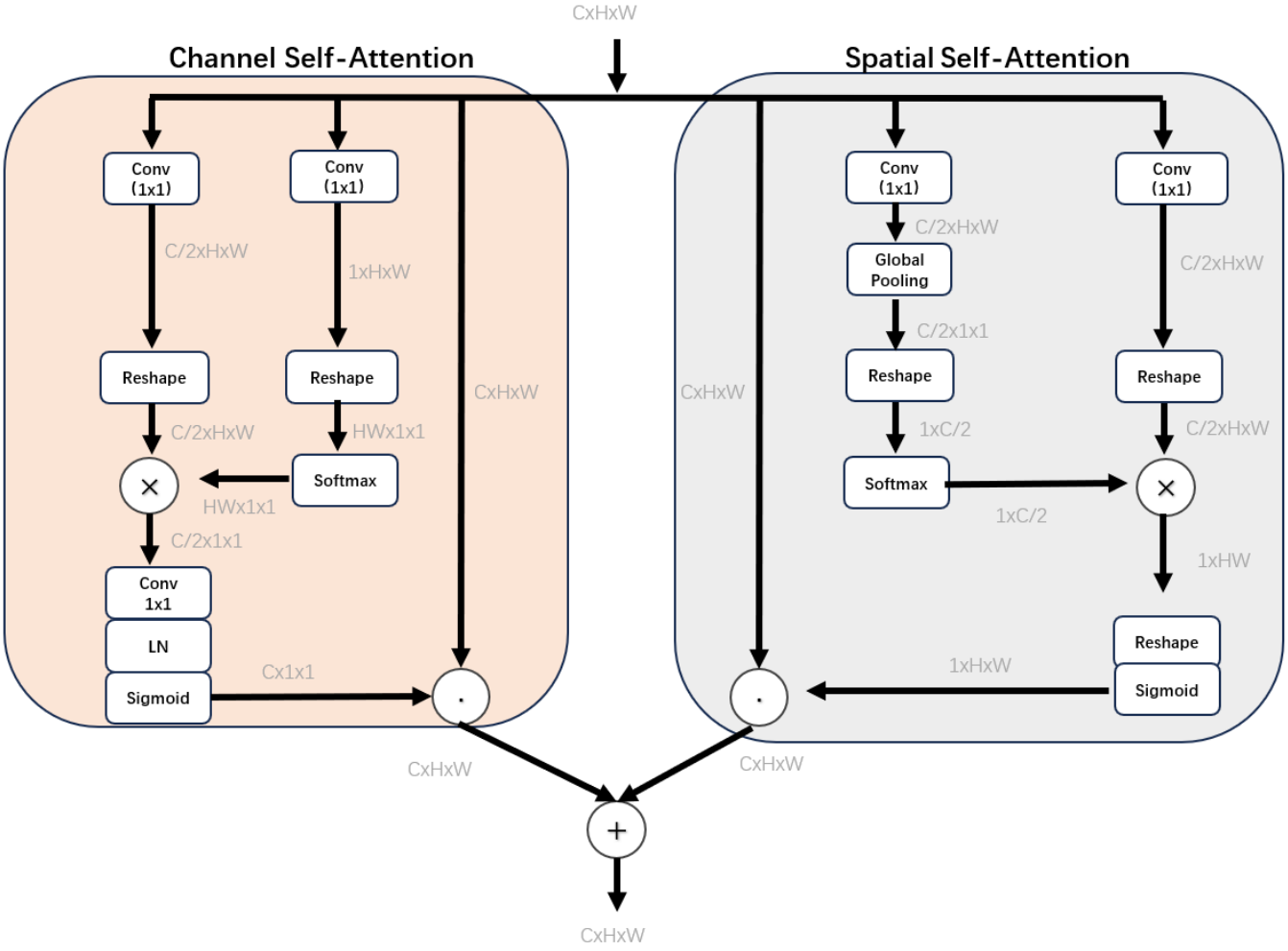

Figure 4. Structure of Dual Channel High-Resolution Attention (DCHRA). We adopt a parallel structure to combine the output of spatial and channel attention mechanisms.

In contrast, the spatial dimension of v is kept at a relatively large level. q information is expanded using Softmax. Multiplying the two parts by the matrix and picking up the reshaping brings the spatial dimension to the original dimension. Sigmoid was then used to keep all parameters between 0 and 1. Formally, we instantiate the DCHRA mechanism as the following DCHRA block:

Channel-only branch $A^{ch}(X) \in \mathcal{R}^{C\times 1\times 1}$:

$$A^{ch}(X) = F_{SG}[W_{z|\theta_1}((\sigma_1(W_v(X))) \times F_{SM}(\sigma_2(W_q(X))))] \quad (1)$$

where $A^{ch}(X) \in \mathcal{R}^{C\times 1\times 1}$ is a feature tensor of one sample; C, 1, and 1 are the number of elements along X's height, width, and channel dimensions, respectively. W_q, W_v, and W_z are 1×1 convolution layers, respectively; σ_1 σ_2 are two tensor reshape operators; $F_{SM}(\bullet)$ is a Softmax operator; and "$\times$" is the matrix dot product operation $F_{SM}(X) = \sum_{j=1}^{N_p} \frac{e^{x_j}}{\sum_{m=1}^{N_p} e^{x_m}} x_j$. The internal number of channels between W_q, W_v, and W_z, is C/2. The output of the channel-only branch is $Z^{ch} = A^{ch}(X) \odot^{ch} X \in \mathcal{R}^{C\times H\times W}$, where $\odot^{ch}$ is a channel-wise multiplication operator.

Spatial-only branch $A^{sp}(X) \in \mathcal{R}^{1\times H\times W}$:

$$A^{sp}(X) = F_{SG}[\sigma_3(F_{SM}(\sigma_1(F_{GP}(W_q(X)))) \times \sigma_2(W_v(X)))] \quad (2)$$

where $F_{GP}(\bullet)$ is a global pooling operator $F_{GP}(X) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} X(:, i, j)$. The output of the spatial-only branch is $Z^{sp} = A^{sp}(X) \odot^{sp} X \in \mathcal{R}^{C \times H \times W}$.

We add the above two branches point by point to obtain the following result:

$$DCHRA(X) = Z^{ch} + Z^{sp} = A^{ch}(X) \bigodot^{ch} X + A^{sp}(X) \bigodot^{sp} X \tag{3}$$

We will analyze the internal resolution and complexity of DCHRA. DCHRA maintains the highest resolution within the module compared to other attention mechanisms in the channel $(C/2)^3$ and spatial ([W, H]) dimensions. In addition, in our channel attention mechanism, Softmax is re-weighted and fused, utilizing Softmax as a nonlinear activation at the feature tensor of size $C/2 \times W \times H$. Our design performs better resolution-squeezing and excitation while having comparable computational complexity for GC blocks. Not only do we retain the complete [W, H] spatial resolution in our spatial attention, but we also internally retain the $2 \times C \times C/2$ learnable parameters in W_q and W_v for nonlinear Softmax re-weighting, which is a much more robust structure than existing blocks.

3.3. IS-IOU

In IS-IoU, we use auxiliary bounding boxes with scales of different sizes. In the regression process, the trend of the IoU values of the auxiliary and actual bounding boxes is the same. The quality of the regression results of the actual bounding box can be obtained with the help of the auxiliary bounding box due to the difference in scale between the auxiliary bounding box and the actual bounding box. According to the conclusion of Inner-IoU, for different IoU samples, the absolute value of the IoU gradient of the auxiliary bounding box with different sizes is different from the absolute value of the IoU gradient of the actual bounding box. For high IoU samples, the absolute value of the IoU gradient of the smaller-scale auxiliary bounding box is larger than the absolute value of the IoU gradient of the actual bounding box. For low IoU samples, the absolute value of the IoU gradient of the IoU gradient of the smaller scale auxiliary bounding box is larger than the absolute value of the IoU gradient of the actual bounding box.

Based on the above analysis, IS-IoU can accelerate convergence by choosing a minor scale auxiliary bounding box when calculating the loss of high IoU samples, and by choosing a larger scale auxiliary bounding box when calculating the loss of low IoU samples. In short, using auxiliary bounding boxes of different sizes can accelerate convergence. In addition, existing IoU losses suffer from weak generalization and slow convergence in different detection tasks, and we use auxiliary bounding boxes to compute losses and accelerate the bounding box regression process. In IS-IoU, we introduce a scaling factor to control the scaling size of the auxiliary bounding box. Our method can overcome the problems of weak generalization and slow convergence in existing methods by using auxiliary bounding boxes and detectors with different scales.

Ground truth (GT) box and anchor are denoted as B^{gt} and B, respectively. The centroids of the GT box and the inner GT box are denoted by (x_c^{gt}, y_c^{gt}), while (x_c, y_c) denotes the centroids of the anchor and the inner anchor. The width and height of the GT box are denoted by w^{gt} and h^{gt}, respectively, and the anchor's width and height are denoted as w and h, respectively. The variable "ratio" corresponds to the scale factor, usually in the range [0.5, 1.5].

$$\begin{cases} b_l^{gt} = x_c^{gt} - \frac{w^{gt} * ratio}{2} \\ b_r^{gt} = x_c^{gt} + \frac{w^{gt} * ratio}{2} \end{cases} \tag{4}$$

$$\begin{cases} b_t^{gt} = y_c^{gt} - \frac{h^{gt} * ratio}{2} \\ b_b^{gt} = y_c^{gt} + \frac{h^{gt} * ratio}{2} \end{cases} \tag{5}$$

$$\begin{cases} b_l = x_c - \frac{w * ratio}{2} \\ b_r = x_c + \frac{w * ratio}{2} \end{cases} \tag{6}$$

$$\begin{cases} b_t = y_c - \frac{h * ratio}{2} \\ b_b = y_c + \frac{h * ratio}{2} \end{cases} \tag{7}$$

$$inter = (min(b_r^{gt}, b_r) - max(b_l^{gt}, b_l)) * (min(b_b^{gt}, b_b) - max(b_t^{gt}, b_t)) \tag{8}$$

$$shape = \left(w^{gt} * h^{gt}\right) * (ratio)^2 + (w * h) * (ratio)^2 - inter \tag{9}$$

$$IOU^{IS} = \frac{inter}{shape} \tag{10}$$

4. Experiments

4.1. Oilpalmuav Dataset

4.1.1. Datasets and Assessment Indicators

We conducted quantitative experiments on the oilpalmuav image dataset to evaluate our proposed methodology. Oilpalmuav is a large UAV image dataset containing 363,877 palms oils labeled in five categories:healthy palms, dead palms, mismanaged palms, smallish palms, and yellowish palms. The dataset's images were obtained from Site 1 for South Kalimantan and Papua and Site 2 for Indonesia and we used UAVs to acquire images. The study area has various land cover types. All 363,877 oil palms in both sites were manually annotated by hand. Some examples of these five types of oil palm are shown in Figure 5. We evaluated the performance of the test images of oilpalmuav and used precision, recall, AP50, mAP, number of parameters, and FLOPs to evaluate the results. Precision and recall are computed using the following equation:

$$Precision = \frac{TP}{TP + FP} \tag{11}$$

$$Recall = \frac{TP}{TP + FN} \tag{12}$$

In object detection, TP represents the number of correctly identified target boxes, while FP represents the number of incorrectly identified target boxes. FN refers to the number of missed target boxes, calculated by subtracting TP from the total number of target boxes. A P–R curve is plotted to assess precision and recall for each class. The area enclosed by this curve determines the mAP of the category. The default mAP value is set at mAP@.5:.95, representing the average mAP over the range of 0.5 to 0.95 with step increments of 0.05. The calculation formulae are shown as follows:

$$AP = \int_0^1 \mathrm{p(r)dr} \tag{13}$$

$$mAP = \frac{\sum_{i=1}^{k} AP_i}{k} \tag{14}$$

where p(r) is the P–R curve and m is the number of target classes. In deep learning, assessing the scale of a model is crucial for designing it effectively. The number of parameters is a reliable indicator of a model's scale, while the amount of computation describes its execution efficiency. In the case of the same or similar indicators, such as accuracy and average precision, the smaller the number of parameters and calculations, the lower the cost of occupied storage and computing power, and the better the model.

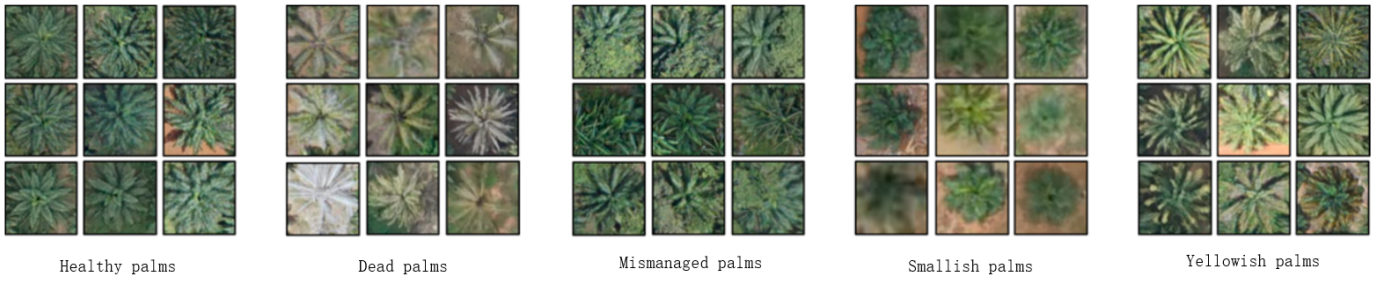

Figure 5. Dataset example.

4.1.2. Experimental Details

We use YOLOv8 and train from scratch for our HR-YOLOv8 on the oilpalmuav dataset without using pre-trained models. We use the NVIDIA A40 for model training and testing. The input image size is uniformly resized to 640 × 640 pixels, and we use YOLOv8's configuration to set the hyperparameters, with our batch_size set to 4 due to training on only one graphics card. We use YOLOv8's default configuration, mosaic enhancement, with the parameter set to 1 for image enhancement. For the optimizer, we used SGD and performed 300 epochs of iterative training.

4.1.3. Ablation Experiment

In order to validate the effectiveness of our approach, we performed an ablation study on oilpalmuav, where we investigated and analyzed the role of each component and compared it with the baseline model of oilpalmuav. Accuracy, recall, average precision, number of parameters, and computation were used as the evaluation indexes of the modules. We performed experiments with YOLOv8l, YOLOv8l + HRFPN, YOLOv8l + DCHRA, and YOLOv8l + IS-IoU to evaluate their impact on performance. We then used HR-YOLOv8 and compared it to the baseline model. The specific ablation experiments are shown in Table 1. In addition, we also compared different loss functions, including ShapeIoU, InnerIoU, and CIoU. The specific results are shown in Table 2. It can be observed that our method significantly improved the performance of each.

Table 1. Table of ablation experiments with different methods.

Methods	P	Q	AP50	mAP	Para (m)	GFLOPs
YOLOv8	87.3	64.1	74.8	62.5	43.63	165.4
YOLOv8 + HR-FPN	89.3	67.2	77.2	66.4	28.27	122.3
YOLOv8 + DCHRA	90.5	66.3	76.8	65.6	44.84	169.5
YOLOv8 + IS-IoU	83.7	67.8	75.6	64.7	43.63	165.4
HR-YOLOv8 (ours)	92.9	67.6	79.4	67.7	29.48	126.4

Table 2. Comparision with other loss functions.

Methods	P	Q	AP50	mAP
Shape-IoU	80.2	62.5	71.1	58.9
Inner-IoU	82.3	61.8	72.6	61.8
CIoU [32]	87.3	64.1	74.8	62.5
ISIoU (ours)	83.7	67.8	75.6	64.7

We first made changes to the FPN component on the baseline model. Under the same conditions, the AP50 using HR-FPN is significantly higher than the baseline model. Since HR-FPN enhances the fusion between features, especially between scales with large differences, it improves the metric mAP by 3.1% for small targets. In addition, the structure of HR-FPN is simple and effective, and the number of parameters and operations is greatly

reduced compared to the baseline model. In the attention mechanism part, we only added four DCHRA blocks in the backbone network of the model, resulting in less feature loss due to downsampling in the backbone network. Thus, with a slight increase in the number of parameters, our model improves the small target metric mAP by 3.1%. These results demonstrate that our proposed module performs well on the dataset, especially for small object detection. Then, we experimented for the loss function and improved the mAP by 2.2% with the addition of ISIoU only. Finally, as shown in Table 2 and Figure 6, we compared it with the existing popular loss functions, and ISIoU is also shown to be faster and more stable during the training process.

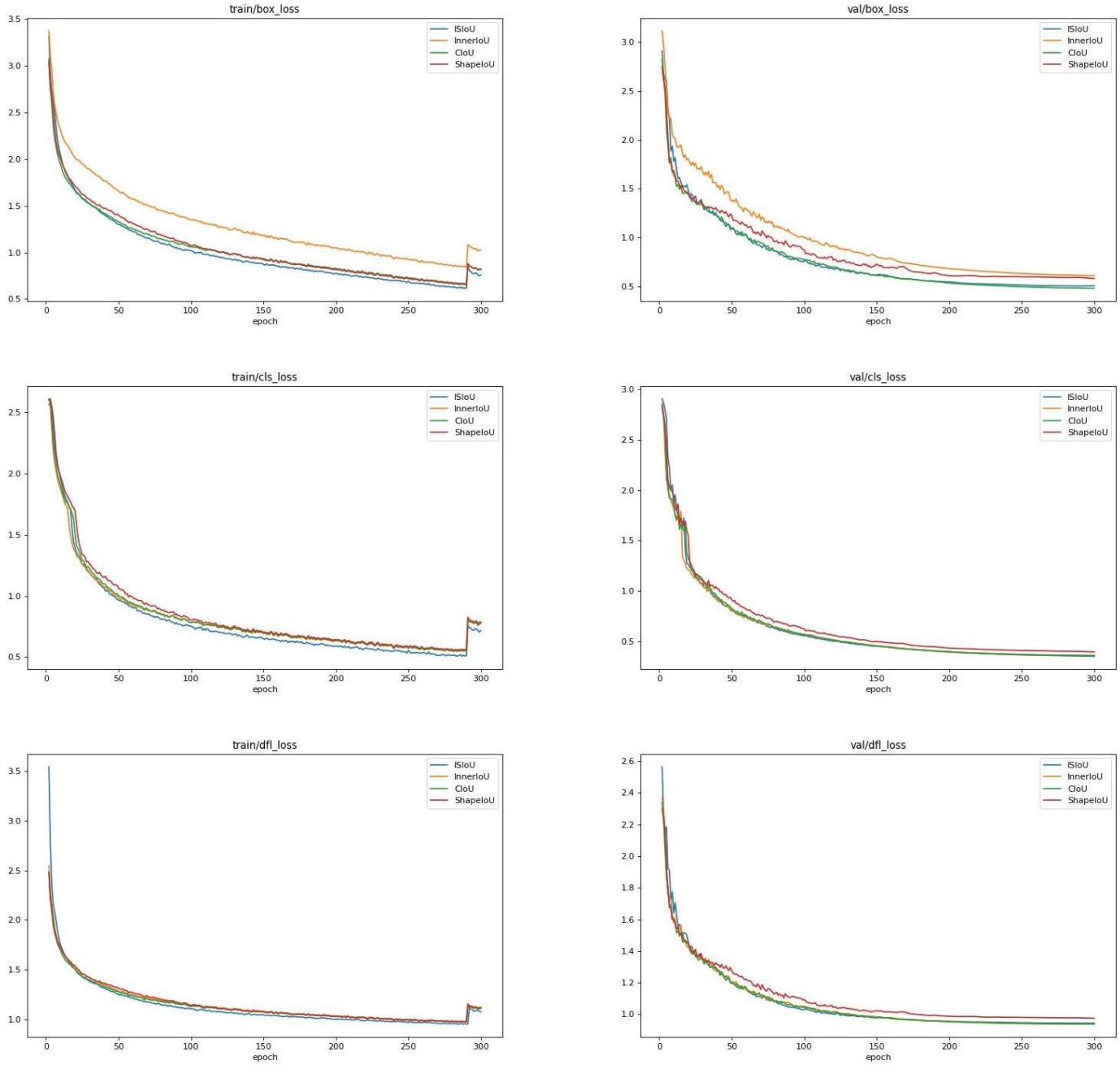

Figure 6. Loss curves for training and validation with different loss functions.

4.1.4. Comparison with Other Methods

Our focus is on evaluating the speed and performance of our models. Precisely, we measure the model's FLOPs and number of parameters. We compare HR-YOLOv8 to other state-of-the-art detectors in the YOLO family, YOLOv8, YOLOv7, and RetinaNet. Table 2 presents a comparison of each target detection algorithm on the oilpalmuav validation set for accuracy metrics. HR-YOLOv8 shows significant improvement, with 5.2%/14.6% improvement in mAP compared to YOLOv8-l and YOLOv7, respectively, while providing comparable or better performance regarding FLOPs and number of parameters. The number provides comparable or better performance. HR-YOLOv8 has a significant 4.8%/12.2% increase in AP compared to RetinaNet and FCOS.

As shown in Tables 3 and 4 and Figure 7, regarding model FLOPs and several parameters, our method reduces the number of parameters by 14.15 m and 39 GFLOPs compared to YOLOv8-l. Compared to YOLOv7, it reduces the number of parameters by 7.02 m and increases by 23.2 GFLOPs. Compared to RetinaNet and FCOS, HR-YOLOv8 reduces the number of parameters by 6.93/2.64 m in the number of parameters and increases by 44.13/48.4 GFLOPs.

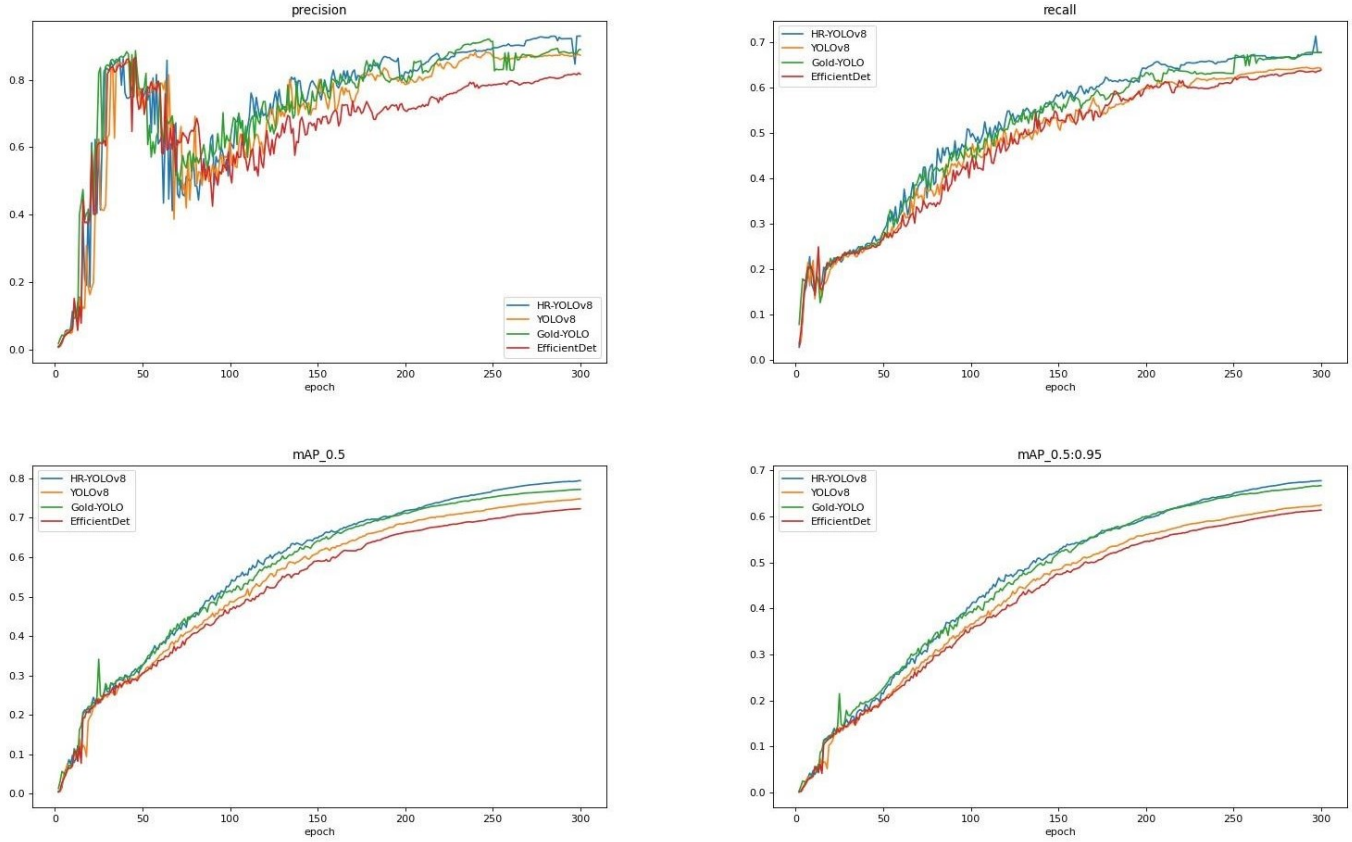

Figure 7. Precision curves, recall curves, and mAP curves for training with different methods.

We list the mAP of our method and other methods on specific categories in Table 5 and increased mAP by at least 0.2% for each category.

Table 3. Comparison experiment of accuracy metrics between HR-YOLOv8 algorithm model and other models on oilpalmuav dataset.

Methods	P	Q	AP50	mAP
YOLOv5	81.5	62.1	74.1	59.6
YOLOv7	76.7	68.3	73.5	53.1
YOLOv8	87.3	64.1	74.8	62.5
RetinaNet [33]	-	-	66.6	62.9
Fcos [34]	-	-	61.1	55.5
Faster RCNN [35]	-	-	60.2	52.3
EfficientDet [30]	81.6	63.8	72.2	61.3
Gold-YOLO [36]	88.9	67.8	77.1	66.6
HR-YOLOv8 (ours)	92.9	67.6	79.4	67.7

Table 4. Experimental table comparing the size of the HR-YOLOv8 model with other models on the oilpalmuav dataset.

Methods	Size	Para (m)	GFLOPs
YOLOv5	640 × 640	53.13	134.7
YOLOv7	640 × 640	36.50	103.2
YOLOv8	640 × 640	43.63	165.4
RetinaNet	640 × 640	36.41	82.27
Fcos	640 × 640	32.12	78.0
Faster RCNN	640 × 640	42.50	167.9
EfficientDet	640 × 640	21.74	55.0
Gold-YOLO	640 × 640	41.56	156.6
HR-YOLOv8 (ours)	640 × 640	29.48	126.4

Table 5. Comparative experiments on map of specific categories between the HR-YOLOv8 model and other models on oilpalmuav dataset.

Methods	Healthy Palm	Dead Palm	Mismanaged Palm	Smallish Palm	Yellowish Palm
EfficientDet	88.5	40.1	42.7	60.3	66.6
YOLOv8	90.0	37.7	46.2	63.0	68.3
Gold-YOLO	89.2	51.9	50.1	61.5	69.2
HR-YOLOv8 (ours)	90.2	54.7	51.4	65.5	71.4

4.1.5. Confusion Matrix

The confusion matrix is a visualization tool for supervised learning that allows for the visualization of how well a model classifies across categories. With the confusion matrix, we can see the percentage of the model that categorizes correctly and incorrectly on each category and the categories that the model predicts incorrectly. To better evaluate the model's performance, we show the confusion matrix for the baseline model and our model, as shown in Figure 8, with different improvements for each category.

4.1.6. Visualization of Results

In order to verify the detection effect of the HR-YOLOv8 algorithm in real scenarios, aerial UAV images from several complex scenarios in the oilpalmuav validation set were selected for testing. The detection results are shown in Figure 9, with the baseline model on the left and the improved model on the right, with oil palm instances included in the boxes and land and other tree species not included. In each row of comparison, it can be found that the confidence of the target box, especially in small areas, has been improved. For example, the confidence of the critical part of the figure and the part of the overlapping box in the lower left corner has been significantly improved compared to the baseline model. In addition, it can be seen from the figure that the improved model has improved the detection accuracy, reducing missed detection, such as the bottom of the second column.

The upper left corner of the first column contains three examples of small oil palm, and it can be found that the left side of the baseline model missed one.

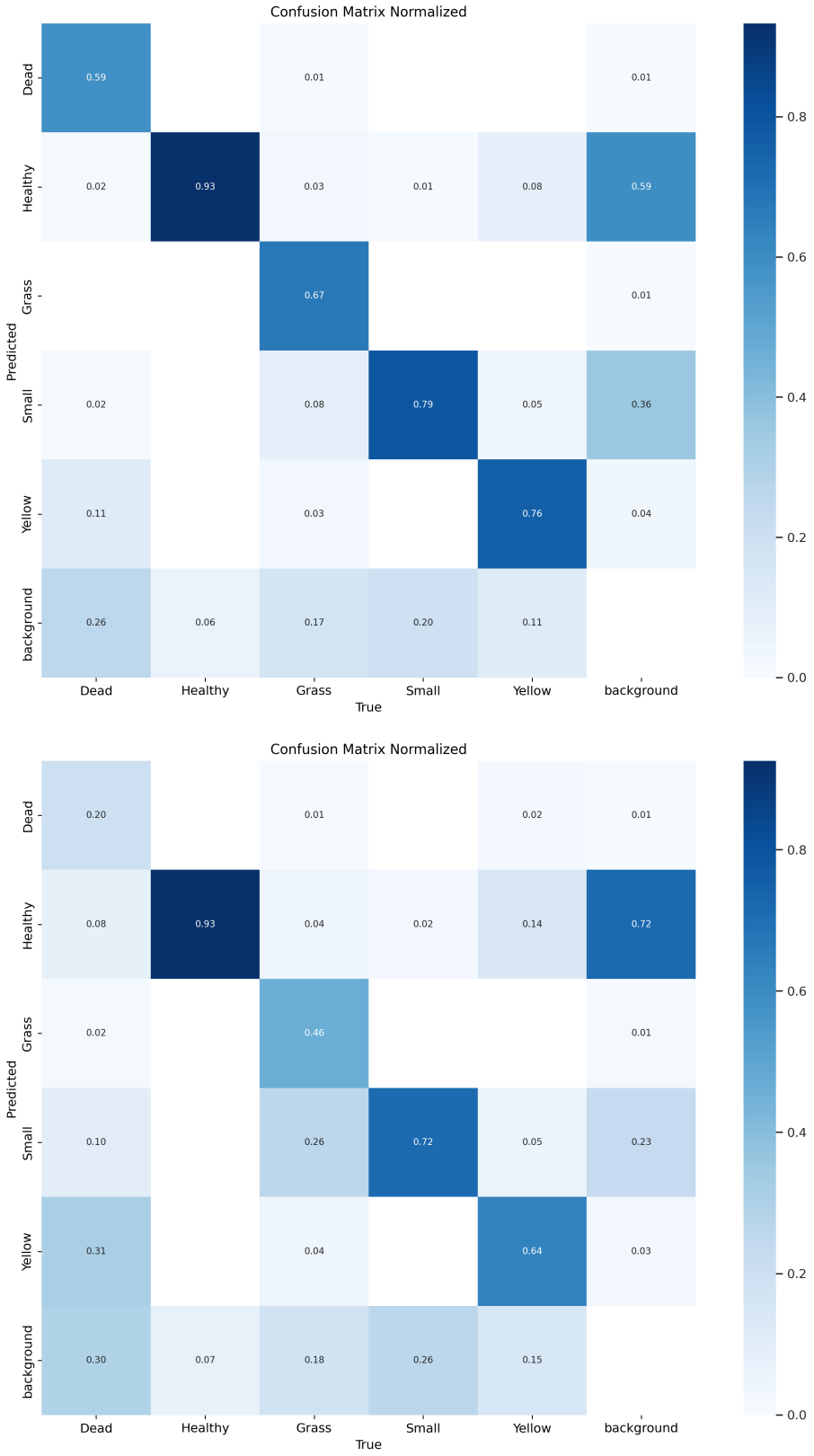

Figure 8. The confusion matrix of our model and the baseline model, with HR-YOLOv8 on the **top** and YOLOv8 on the **bottom**. The diagonal is the correctness of the model predictions (in decimal).

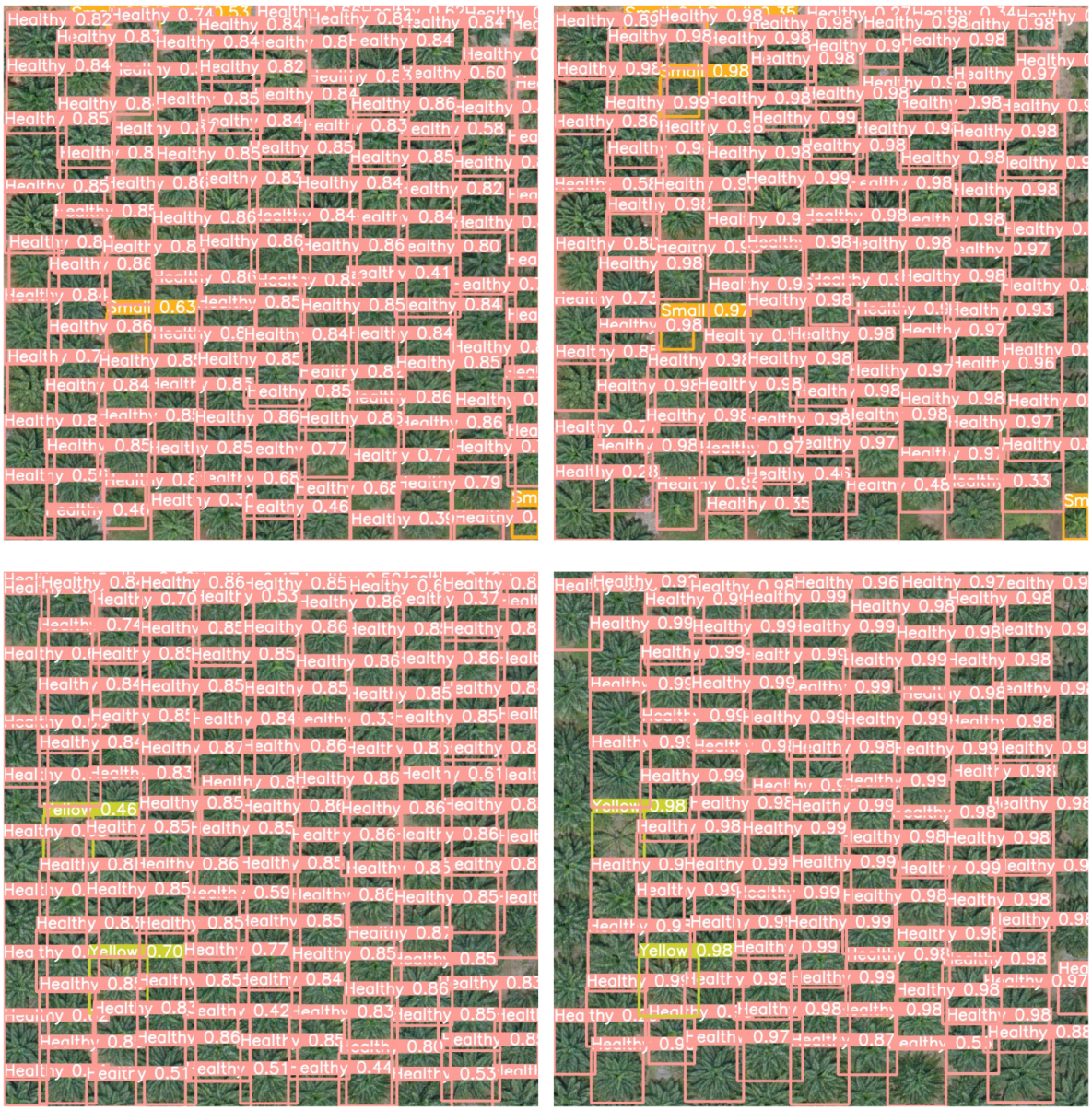

Figure 9. Visualization of results. The two pictures on the (**left**) are HR-YOLOv8, and the two pictures on the (**right**) are YOLOv8. The pink boxes in the picture are healthy palm, the orange boxes are small palm, and the light green boxes are yellowish palm.

4.2. Strawberry Ripeness Dataset

We also validated HR-YOLOv8 on the strawberry ripeness dataset. The strawberry ripeness dataset is obtained from the public domain of Baidu PaddlePaddle AI Studio and the download URL is https://aistudio.baidu.com/datasetdetail/147119 (accessed on 17 December 2023) . We use the labeling tools to check and correct the labeling information of the strawberry fruits. The strawberry ripeness dataset was taken in a greenhouse and contains 3100 pictures. The dataset is divided into a training set (2480 pictures), a verification set (310 pictures), and a test set (310 pictures) according to the ratio of 8:1:1. The strawberry ripeness dataset contains three types of samples: half_ripened, green, and fully_ripened strawberries, with 10,880, 2408, and 2835 samples, respectively. The basic information of the dataset is shown in Table 6.

Table 6. Strawberry ripeness dataset basic information.

Dataset	Images	Half_Ripened	Green	Fully_Ripened
Training Set	2480	1916	8806	2246
Validation Set	310	241	1017	294
Test Set	310	251	1057	295
Total	3100	2408	10,880	2835

As shown in Table 7, the mAP of our method is improved by 0.6% compared to YOLOv8, which proves the superiority and generalization ability of our method.

Table 7. Experimental table comparing the size of the HR-YOLOv8 model with other models on the strawberry ripeness dataset.

Methods	Size	P	Q	AP50	mAP	Para (m)	GFLOPs
YOLOv5	640 × 640	90.2	91.5	96.0	87.1	53.13	134.7
YOLOv7	640 × 640	89.2	90.1	95.0	86.3	36.50	103.2
YOLOv8	640 × 640	90.5	92.3	96.4	87.6	43.63	165.4
Faster RCNN	640 × 640	73.9	93.2	94.9	85.7	42.50	167.9
HR-YOLOv8 (ours)	640 × 640	92.3	92.1	97.1	88.2	29.48	126.4

5. Conclusions

Our paper proposes an object detection method for recognizing and monitoring the growth status of crops in agriculture, aiming to solve the detection of more minor crops on UAV images—the improvement of YOLOv8 multiscale feature fusion with partially missing information. Considering the need for a model with good real-time performance and appropriate parameters in practical application, we propose HR-YOLOv8, which adds a new dual self-attention mechanism to the YOLOv8 backbone network in order to reduce the potential loss of high-resolution information in deep neural networks through pooling/downsampling, maintaining a high internal resolution in the attentional computation between existing attentional blocks. To fit the output distribution of a typical fine-grained regression, Softmax–Sigmoid combinations are fused in the channel- and spatial-only attention branches and improve the model's attention to small targets. Second, we propose a feature fusion module named HR-FPN concerning the network structure of HR-Net. As a result, our method can maintain high resolution rather than recovering high resolution from low resolution; thus, the learned representation may be more spatially accurate. We repeat the multiresolution fusion to improve the high-resolution representation with the help of the low-resolution representation. Finally, we use IS-IoU as the bounding box regression loss and propose shape–point distance and shape–normalized distance loss for the tiny-object-detection task, taking into account the influence of the shape and scale of the bounding box regression samples themselves on the bounding box regression. Our proposed HR-YOLOv8 model improves the detection performance on the crop growth state. We conducted a series of comparative experiments on the dataset using HR-YOLOv8, and the experimental results show that the method outperforms existing methods for detection.

Compared with various object detection methods, the method in this paper have better detection results and higher accuracy. In the follow-up research, the paper focuses on the precise identification of crop location without increasing the complexity of the network to improve the detection effect of the crop growth status. The ultimate goal is to deploy the model in the inspection of UAVs with a reasonable size, fast real-time, high detection accuracy, and broad applicability of the crop growth status detection system to meet the future development and requirements of intelligent agriculture. Additionally, we will explore the potential of our method for other precise agriculture applications in our future work.

Author Contributions: Conceptualization, J.Z.; methodology, J.Z. validation, J.Z.; writing—original draft preparation, J.Z., W.Y. and Z.L.; supervision, D.C.; funding acquisition, W.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key Research and Development Program of China (Grant No. 2022ZD0115802), the Key Research and Development Program of the Autonomous Region (Grant No. 2022B01008), the National Natural Science Foundation of China (Grant No. 62262065), the Tianshan Science and Technology Innovation Leading talent Project of the Autonomous Region (Grant No. 2022TSYCLJ0037).

Institutional Review Board Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

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

References

1. Zheng, J.; Fu, H.; Li, W.; Wu, W.; Yu, L.; Yuan, S.; Tao, W.Y.W.; Pang, T.K.; Kanniah, K.D. Growing status observation for oil palm trees using Unmanned Aerial Vehicle (UAV) images. *ISPRS J. Photogramm. Remote Sens.* **2021**, *173*, 95–121. [CrossRef]
2. Yu, Z.; Cao, Z.; Wu, X.; Bai, X.; Qin, Y.; Zhuo, W.; Xiao, Y.; Zhang, X.; Xue, H. Automatic image-based detection technology for two critical growth stages of maize: Emergence and three-leaf stage. *Agric. For. Meteorol.* **2013**, *174*, 65–84. [CrossRef]
3. Zhang, P.; Li, D. EPSA-YOLO-V5s: A novel method for detecting the survival rate of rapeseed in a plant factory based on multiple guarantee mechanisms. *Comput. Electron. Agric.* **2022**, *193*, 106714. [CrossRef]
4. Yang, Q.; Li, W.; Yang, X.; Yue, L.; Li, H. Improved YOLOv5 method for detecting growth status of apple flowers. *Comput. Eng. Appl.* **2022**, *58*, 237–246.
5. Wang, G.; Chen, Y.; An, P.; Hong, H.; Hu, J.; Huang, T. UAV-YOLOv8: A small-object-detection model based on improved YOLOv8 for UAV aerial photography scenarios. *Sensors* **2023**, *23*, 7190. [CrossRef] [PubMed]
6. Wang, J.; Sun, K.; Cheng, T.; Jiang, B.; Deng, C.; Zhao, Y.; Liu, D.; Mu, Y.; Tan, M.; Wang, X.; et al. Deep high-resolution representation learning for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *43*, 3349–3364. [CrossRef] [PubMed]
7. Zhang, H.; Xu, C.; Zhang, S. Inner-iou: more effective intersection over union loss with auxiliary bounding box. *arXiv* **2023**, arXiv:2311.02877.
8. Zhang, H.; Zhang, S. Shape-IoU: More Accurate Metric considering Bounding Box Shape and Scale. *arXiv* **2023**, arXiv:2312.17663.
9. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
10. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.
11. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
12. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
13. Wu, W.; Liu, H.; Li, L.; Long, Y.; Wang, X.; Wang, Z.; Li, J.; Chang, Y. Application of local fully Convolutional Neural Network combined with YOLO v5 algorithm in small target detection of remote sensing image. *PLoS ONE* **2021**, *16*, e0259283. [CrossRef]
14. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. Yolox: Exceeding yolo series in 2021. *arXiv* **2021**, arXiv:2107.08430.
15. Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W.; et al. YOLOv6: A single-stage object detection framework for industrial applications. *arXiv* **2022**, arXiv:2209.02976.
16. Li, C.; Li, L.; Geng, Y.; Jiang, H.; Cheng, M.; Zhang, B.; Ke, Z.; Xu, X.; Chu, X. Yolov6 v3. 0: A full-scale reloading. *arXiv* **2023**, arXiv:2301.05586.
17. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 7464–7475.
18. Shaw, P.; Uszkoreit, J.; Vaswani, A. Self-attention with relative position representations. *arXiv* **2018**, arXiv:1803.02155.
19. Bello, I.; Zoph, B.; Vaswani, A.; Shlens, J.; Le, Q.V. Attention augmented convolutional networks. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 3286–3295.
20. Andreoli, J.M. Convolution, attention and structure embedding. *arXiv* **2019**, arXiv:1905.01289.
21. Ramachandran, P.; Parmar, N.; Vaswani, A.; Bello, I.; Levskaya, A.; Shlens, J. Stand-alone self-attention in vision models. *Adv. Neural Inf. Process. Syst.* **2019**, *32* .
22. Cao, Y.; Xu, J.; Lin, S.; Wei, F.; Hu, H. Gcnet: Non-local networks meet squeeze-excitation networks and beyond. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, Seoul, Republic of Korea, 27 October–2 November 2019.
23. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*.
24. Cordonnier, J.B.; Loukas, A.; Jaggi, M. On the relationship between self-attention and convolutional layers. *arXiv* **2019**, arXiv:1911.03584.
25. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
26. 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.

27. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
28. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path aggregation network for instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8759–8768.
29. Guo, C.; Fan, B.; Zhang, Q.; Xiang, S.; Pan, C. Augfpn: Improving multi-scale feature learning for object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 12595–12604.
30. Tan, M.; Pang, R.; Le, Q.V. Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10781–10790.
31. Ghiasi, G.; Lin, T.Y.; Le, Q.V. Nas-fpn: Learning scalable feature pyramid architecture for object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 7036–7045.
32. Zheng, Z.; Wang, P.; Ren, D.; Liu, W.; Ye, R.; Hu, Q.; Zuo, W. Enhancing geometric factors in model learning and inference for object detection and instance segmentation. *IEEE Trans. Cybern.* **2021**, *52*, 8574–8586. [CrossRef]
33. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
34. Tian, Z.; Shen, C.; Chen, H.; He, T. FCOS: Fully convolutional one-stage object detection. arXiv 2019. *arXiv* **1904**, arXiv:1904.01355.
35. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*. [CrossRef] [PubMed]
36. Wang, C.; He, W.; Nie, Y.; Guo, J.; Liu, C.; Wang, Y.; Han, K. Gold-YOLO: Efficient object detector via gather-and-distribute mechanism. *Adv. Neural Inf. Process. Syst.* **2024**, *36*.