

Tea Bud Detection and Recognition Based on an Improved YOLOv8 Algorithm

Qin Huang, Gang Jiang*, Wei Tang, Junmin Li

School of Mechanical Engineering, Xihua University, Chengdu 610039, Sichuan, China

*Corresponding author: Gang Jiang

Abstract: In natural tea gardens, tea buds are small, mutually occluded, and surrounded by background texture that closely resembles the target, which causes both missed and false detections. To raise detection accuracy without increasing model complexity, this paper proposes YOLOv8n-HQ, an improved YOLOv8n-based detector for tea buds. The algorithm introduces a P2 high-resolution detection layer to build a four-scale feature pyramid for stronger small-object representation, embeds a convolutional block attention module (CBAM) to suppress background interference, designs a lightweight re-parameterized convolution (LRPC) module to compress the model, and adopts a Wise-IoU loss to improve bounding-box regression accuracy. On a self-built dataset of 15 000 images, the proposed algorithm attains 89.9% mAP@0.5 and 67.6% mAP@0.5:0.95, which are 4.3 and 4.1 percentage points higher than the baseline; the parameter count and computational cost are reduced to 2.05 M and 6.0 GFLOPs, decreases of 35.9% and 31.0%, so that higher accuracy and a smaller model are achieved together. On both detection accuracy and model complexity the algorithm outperforms the compared models.

Keywords: Tea bud detection; YOLOv8; attention mechanism; structural re-parameterization; Wise-IoU loss.

1. Introduction

Famous teas are graded and priced by the tenderness of the fresh leaf, and the price of single buds, one bud with one leaf and one bud with two leaves differs substantially. At present this grading is still performed almost entirely by eye, and the picking cost has reached more than 60% of the total production cost of famous tea. Automatic detection of tea buds together with tenderness-level recognition is therefore a prerequisite for the practical deployment of tea-picking robots.

Object detection based on convolutional neural networks provides an effective route to this task. The YOLO family in particular strikes a good balance between accuracy and efficiency through its single-stage regression paradigm, and has already been applied to tea bud detection [1-2]. Two-stage detectors are represented by Faster R-CNN [3], while single-stage detectors are represented by SSD and the YOLO family [4-5]. Tea gardens nevertheless impose three constraints on a detector. The first is a scale constraint: the equivalent edge length of a tea bud is concentrated between 20 and 40 pixels, while the smallest detection scale of YOLOv8n, P3, has a downsampling factor of 8, so a single bud occupies only 2 to 5 pixels on the P3 feature map. The second is a background constraint: branches, soil and dead leaves closely resemble tea buds in both colour and texture, and buds and leaves overlap and occlude one another extensively, so false and missed detections occur simultaneously. The third is a complexity constraint: the conventional route to higher accuracy is paid for by a multiplied parameter count and computational cost, which conflicts with the computing budget of an embedded platform. The three constraints are coupled, and have long made detection accuracy and model complexity difficult to optimise together.

Taking YOLOv8n as the baseline, this paper addresses the

three constraints with a corresponding set of improvements: a P2 high-resolution detection layer is introduced to push the smallest detection scale down from P3 to P2 and to build a P2–P5 four-scale feature pyramid; a CBAM attention module [6] is embedded to re-calibrate the feature response along both the channel and the spatial dimension; an LRPC re-parameterized convolution module is proposed, which folds the multi-branch convolution used during training into a single convolution at inference time following the principle of structural re-parameterization [7], and which is combined with depthwise separable convolution and channel pruning; and a Wise-IoU loss [8] is adopted to suppress the gradient contribution of low-quality samples through a dynamic non-monotonic focusing coefficient. Acting together within a single framework, the four improvements allow YOLOv8n-HQ to reach 89.9% mAP@0.5 while reducing the parameter count to 2.05 M and the computational cost to 6.0 GFLOPs, so that detection accuracy and model complexity are optimised jointly.

2. Improved YOLOv8n-HQ Algorithm

YOLOv8 consists of a backbone, a neck and a decoupled detection head [9]. The backbone is a modified CSPDarknet containing five downsampling stages, in which the C2f module enhances gradient reuse through cross-stage partial connections; the neck adopts a path aggregation feature pyramid network (PAFPN) that fuses deep semantic and shallow positional information along top-down and bottom-up paths and outputs three scales, P3, P4 and P5; and the head decouples the classification and regression branches and regresses the distances from the object centre to the four sides of the bounding box in an anchor-free manner. Four modifications are made on this framework, and the resulting network is shown in Fig. 1.

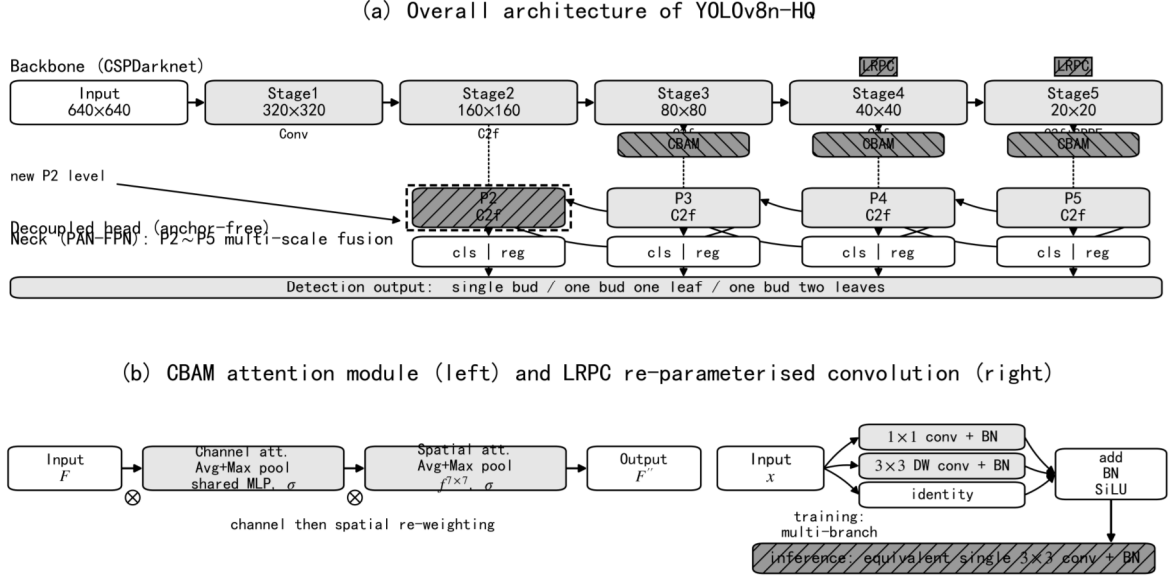


Figure 1. Overall architecture of the proposed YOLOv8n-HQ with its three structural modifications

2.1. P2 High-Resolution Detection Layer

Lowering the detection scale is the most direct way to reduce the loss of small-object features. The $160 \times 160 \times 64$ feature map produced by stage 2 of the backbone is reduced to 32 channels by a 1×1 convolution and concatenated along the channel dimension with the P3 feature map returned by the bottom-up path; after C2f fusion it yields a $160 \times 160 \times 32$ feature map for detection. The smallest detection scale is thereby pushed from P3 down to P2, the downsampling factor falls from 8 to 4, and the effective response area of a single bud is enlarged by roughly a factor of four. The output

channels of P3, P4 and P5 are adjusted to 64, 128 and 256 so that the channel ratio between the four scales stays balanced. The P2 branch has the highest resolution and the finest localisation, and together with P3–P5 it forms a complete coverage from detail to semantics.

2.2. CBAM Attention Module

To address the close similarity between background and target features, a CBAM module [6] is embedded after the C2f module of stages 3, 4 and 5 of the backbone, re-calibrating the feature map along both the channel and the spatial dimension. The computation is

$$F' = M_c(F) \otimes F, \quad F'' = M_s(F') \otimes F' \quad (1)$$

$$M_c(F) = \sigma(\text{MLP}(\text{AvgPool}(F)) + \text{MLP}(\text{MaxPool}(F))) \quad (2)$$

$$M_s(F') = \sigma(f^{7 \times 7}([\text{AvgPool}(F'); \text{MaxPool}(F')])) \quad (3)$$

where F is the input feature map; $M_c(\cdot)$ and $M_s(\cdot)$ are the channel and spatial attention; $\otimes$ denotes element-wise multiplication; σ is the Sigmoid function; $f^{7 \times 7}$ is a 7×7 convolution; r is the channel compression ratio, taken as 16; and $[\cdot; \cdot]$ denotes concatenation along the channel dimension.

The channel attention compresses the spatial dimension into a channel descriptor, strengthening the response of the channels important for tea buds and suppressing background noise channels; the spatial attention pools along the channel dimension and then encodes positional information with a 7×7 convolution, highlighting the region in which the bud lies. The two sub-modules are connected in series and their outputs are added to the outputs of the three stages with weights of 1.0, 0.8 and 0.8, so that the network suppresses the background at several levels simultaneously. The additional parameter cost is 0.06 M.

2.3. LRPC Lightweight Re-parameterized Convolution

Lightweight design and accuracy optimisation are carried out together rather than in opposition. Following the principle of structural re-parameterization [7], the C2f module is rebuilt (L in LRPC stands for lightweight): during training the standard convolution is replaced by three parallel branches consisting of a 1×1 convolution, a 3×3 depthwise separable convolution and an identity mapping, whose outputs are summed and then passed through batch normalisation and a SiLU activation; at inference time the batch-normalisation parameters are folded into the convolution kernel and the three branches are merged into a single 3×3 convolution. Writing W for the convolution kernel weight and γ , β , μ and σ^2 for the scale, shift, mean and variance of the batch normalisation, the folding is

$$W' = \frac{\gamma}{\sqrt{\sigma^2 + \varepsilon}} \odot W, \quad b' = \beta - \frac{\gamma\mu}{\sqrt{\sigma^2 + \varepsilon}} \quad (4)$$

where W' and b' are the fused kernel weight and bias; ε is a small constant preventing division by zero, taken as 1×10^{-5} ; and $\odot$ denotes channel-wise multiplication.

The key point is the decoupling of the parameters between the training and inference phases: the multi-branch structure of the training phase provides richer gradient propagation paths, which favours convergence and accuracy, whereas at inference time the branches are merged equivalently, removing only the structural overhead introduced by the training-time branches and leaving no branch redundancy. On this basis the standard convolutions of stages 4 and 5 are replaced by depthwise separable convolutions, and channel pruning is applied to some convolution layers at a ratio of 0.2 according to the sparsity of the batch-normalisation scale factor γ . Compared with the configuration without LRPC the parameter count falls from 3.31 M to 2.05 M and the

computational cost from 9.6 GFLOPs to 6.0 GFLOPs; relative to the YOLOv8n baseline the parameter count and computational cost are reduced by 35.9% and 31.0%, with no loss of detection accuracy.

2.4. Wise-IoU Loss Function

The bounding boxes of overlapping and occluded bud samples are annotated with comparatively low positional quality, and assigning them the same regression weight as high-quality samples would interfere with the learning of clear samples. The Wise-IoU v3 loss [8] is therefore adopted in place of CIoU. It first weights the IoU loss by a distance-attention coefficient R_{WIoU} to obtain WIoU v1, and then scales the result by a dynamic non-monotonic focusing coefficient r derived from the outlier degree; the definitions are

$$L_{\text{IoU}} = 1 - \text{IoU}, \quad L_{\text{WIoUv1}} = R_{\text{WIoU}} \cdot L_{\text{IoU}},$$

$$R_{\text{WIoU}} = \exp\left(\frac{(x - x_{\text{gt}})^2 + (y - y_{\text{gt}})^2}{(W_g^2 + H_g^2)^*}\right) \quad (5)$$

$$\beta = \frac{L_{\text{IoU}}}{L_{\text{IoU}}}, \quad r = \frac{\beta}{\delta \cdot \alpha^{\beta-\delta}}, \quad L_{\text{WIoUv3}} = r \cdot L_{\text{WIoUv1}} \quad (6)$$

where (x, y) and $(x_{\text{gt}}, y_{\text{gt}})$ are the centres of the predicted and ground-truth boxes; W_g and H_g are the width and height of the smallest enclosing box; the superscript $*$ indicates that the quantity does not participate in gradient back-propagation; the mean of L_{IoU} is estimated by an exponential moving average; β is the outlier degree, expressing how far the predicted box deviates from the current average regression quality; and α and δ are hyper-parameters controlling the curvature and the peak position of the focusing curve, taken as 1.9 and 3.0 respectively following reference [8], so that r attains its maximum value of 1 when $\beta = \delta$.

As Eqs. (5) and (6) show, r reaches its maximum of 1 at $\beta = \delta$ and decreases whenever β deviates from δ in either direction, so the focusing mechanism is non-monotonic: for low-quality samples with a large β the decay of r effectively suppresses their harmful gradients, while for well-localised samples with a small β the reduction of r avoids redundant optimisation. The denominator of R_{WIoU} uses the size of the smallest enclosing box without its gradient, so that the coefficient acts only as a weighting factor and does not introduce a gradient that hinders convergence; this is favourable for tea buds, which are predominantly small to medium-sized. The regression objective is a weighted sum of the classification loss, the Wise-IoU regression loss and the distribution focal loss, with weights of 0.5, 7.5 and 1.5.

3. Experimental Results and Analysis

3.1. Dataset Construction and Experimental Setup

The images were collected from three standardised tea gardens (cultivars Fuding Dabaicha and Longjing 43) in two tea seasons, spring and autumn, covering four lighting conditions: clear sky with front light, clear sky with back light, overcast diffuse light and high humidity after rain. The acquisition devices included a mobile phone, a UAV and an industrial camera, and 15 000 images were obtained and uniformly resized to 640×640 pixels. Following the picking standard for famous tea, the samples were annotated into three tenderness grades: single bud, one bud with one leaf, and one bud with two leaves. Two technicians annotated the images independently and cross-checked the results, giving 132 600 valid annotated instances, which were divided into training, validation and test sets in a ratio of 7:2:1. Statistics on the annotated boxes show that the equivalent edge length of the targets is concentrated between 15 and 90 pixels and that 71.3% are smaller than 32 pixels; this small-object-dominated characteristic determines the design direction of the detection scales. The class composition and the target-size distribution are shown in Fig. 2.

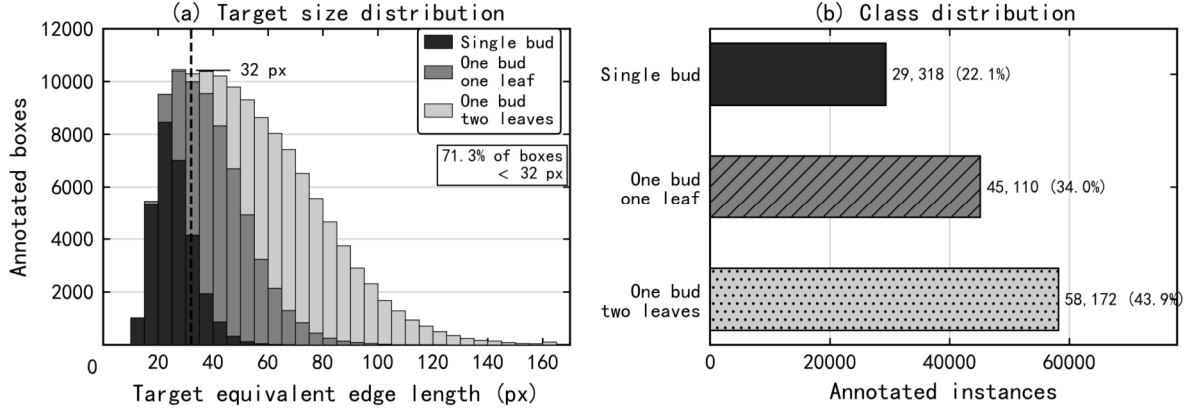


Figure 2. Class composition and target-size distribution of the self-built tea bud dataset

The experiments were run on an NVIDIA RTX 4090 with Ubuntu 22.04, PyTorch 2.0.1 and Ultralytics YOLOv8 [9]. The input resolution was 640×640 pixels; the optimiser was stochastic gradient descent with an initial learning rate of 0.01, a momentum of 0.937 and a weight decay of 0.0005; the learning rate followed a cosine annealing schedule with three warm-up epochs; the batch size was 16 and the total number of training epochs was 300, the weights giving the highest validation $mAP@0.5$ being taken as the final model. Online augmentation during training comprised Mosaic stitching, random rotation and translation within $\pm 15^\circ$, colour jitter and random scaling and flipping, with Copy-Paste augmentation added for small objects. To ensure a fair comparison, all compared models were retrained to convergence on the same data split and with the same training parameters.

3.2. Evaluation Metrics

The evaluation metrics are precision P , recall R , average precision AP , mean average precision mAP and frame rate (frames per second) FPS , where precision and recall are defined as

$$P = \frac{TP}{TP + FP}, \quad R = \frac{TP}{TP + FN} \quad (7)$$

where TP , FP and FN are the numbers of true positives, false positives and false negatives.

The average precision of a class is calculated as the area under its precision–recall curve, and the mean average

precision is the arithmetic mean over all classes:

$$AP = \int_0^1 P(R) dR \quad (8)$$

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

where $P(R)$ is the precision as a function of recall and N is the number of classes, taken as $N = 3$. $mAP@0.5$ is the mean average precision at an IoU threshold of 0.5, while $mAP@0.5:0.95$ is the average of the mAP values obtained as the IoU threshold increases from 0.5 to 0.95 in steps of 0.05, and therefore imposes a stricter requirement on bounding-box localisation.

The frame rate is calculated as $FPS = N_{img}/t_{total}$, where N_{img} is the total number of test images and t_{total} is the total inference time including pre-processing and post-processing; all speed metrics were measured five times on the same platform and averaged.

3.3. Ablation Study

To verify the individual contribution of each module, CBAM, the P2 detection layer, LRPC and the Wise-IoU loss were added to the YOLOv8n baseline in turn, giving five ablation configurations. The results are listed in Table 1.

Table 1. Ablation results of each improvement module

ID	CBAM	P2 layer	LRPC	WIoU	Params/M	GFLOPs	$mAP@0.5/\%$	$mAP@0.5:0.95/\%$
A (baseline)	×	×	×	×	3.20	8.7	85.6	63.5
B	✓	×	×	×	3.25	8.9	86.7	64.3
C	✓	✓	×	×	3.31	9.6	87.8	65.0
D	✓	✓	✓	×	2.05	6.0	88.9	65.9
E (ours)	✓	✓	✓	✓	2.05	6.0	89.9	67.6

Note: ✓ indicates the module is enabled; × indicates disabled.

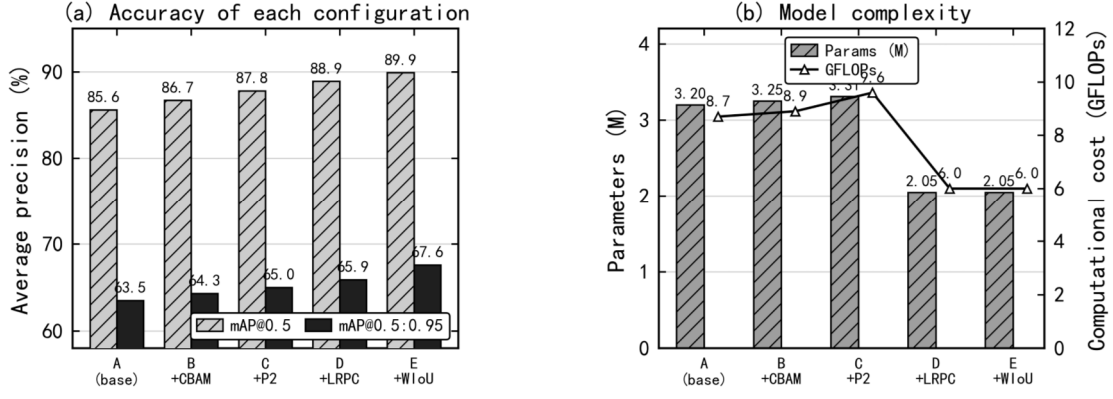


Figure 3. Ablation results of the individual improvement modules

Table 1 and Fig. 3 show that the four improvements contribute independently and to a comparable degree. CBAM raises mAP@0.5 by 1.1 percentage points at a cost of 0.05 M parameters, indicating that channel and spatial attention together suppress the interference of the tea-garden background in a stable manner. The P2 detection layer also raises mAP@0.5 by 1.1 percentage points, matching the addition of a small-object detection scale, at a cost of 0.06 M parameters and 0.7 GFLOPs. LRPC reduces the parameter count from 3.31 M to 2.05 M (a decrease of 38.1%) and the computational cost from 9.6 GFLOPs to 6.0 GFLOPs (a decrease of 37.5%) while still raising mAP@0.5 by 1.1 percentage points, confirming that re-parameterization combined with depthwise separable convolution can compress the model while preserving its feature-extraction capability. The Wise-IoU loss contributes a further 1.0 percentage point and raises mAP@0.5:0.95 from 65.9% to 67.6%. With the four improvements combined, mAP@0.5 rises from 85.6% to 89.9% (an increase of 4.3 percentage

points) and mAP@0.5:0.95 from 63.5% to 67.6% (an increase of 4.1 percentage points), while the parameter count falls from 3.20 M to 2.05 M (a decrease of 35.9%) and the computational cost from 8.7 GFLOPs to 6.0 GFLOPs (a decrease of 31.0%), so that detection accuracy and model complexity are optimised simultaneously. It should be noted that the accuracy gain of the P2 detection layer is obtained at the cost of additional computation, and that the LRPC module recovers this overhead; together they support the joint optimisation of accuracy and complexity.

3.4. Comparison with State-of-the-Art Detectors

To establish the performance level of the proposed algorithm it is compared with YOLOv5s, YOLOv7-tiny and the YOLOv8n baseline under identical conditions. The results are listed in Table 2.

Table 2. Performance comparison with mainstream detectors on the self-built tea bud dataset

Model	Params/M	GFLOPs	mAP@0.5/%	mAP@0.5:0.95/%	FPS
YOLOv5s	7.20	16.5	86.8	63.9	71
YOLOv7-tiny	6.03	13.2	84.5	61.7	112
YOLOv8n (baseline)	3.20	8.7	85.6	63.5	128
YOLOv8n-HQ (ours)	2.05	6.0	89.9	67.6	143

Note: FPS is measured on an NVIDIA RTX 4090 with FP16 precision and batch size 1.

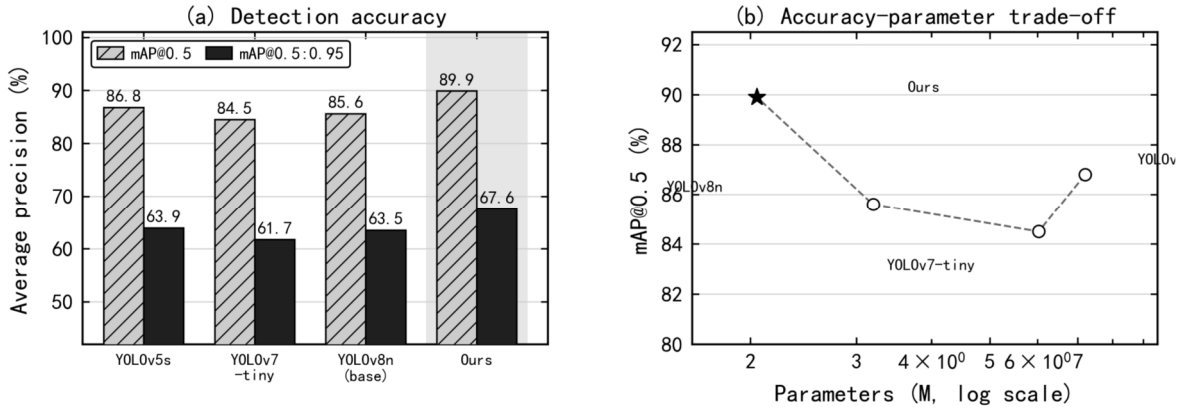


Figure 4. Detection accuracy and parameter count of the compared models

Table 2 and Fig. 4 show that the proposed algorithm outperforms the compared models, which have larger

parameter counts, in both detection accuracy and model complexity. Against YOLOv5s the parameter count falls from 7.20 M to 2.05 M, a decrease of 71.5%, and the computational cost from 16.5 GFLOPs to 6.0 GFLOPs, a decrease of 63.6%, while $mAP@0.5$ and $mAP@0.5:0.95$ rise by 3.1 and 3.7 percentage points respectively. Against YOLOv7-tiny the parameter count and the computational cost are reduced by 66.0% and 54.5%, while $mAP@0.5$ and $mAP@0.5:0.95$ rise by 5.4 and 5.9 percentage points respectively. Against the YOLOv8n baseline the parameter count falls from 3.20 M to 2.05 M (a decrease of 35.9%) and the computational cost from 8.7 GFLOPs to 6.0 GFLOPs (a decrease of 31.0%), while $mAP@0.5$ and $mAP@0.5:0.95$ rise by 4.3 and 4.1 percentage points respectively. These results show that the proposed algorithm attains higher detection accuracy with a smaller model, outperforming the compared models in both accuracy and complexity; in particular, the improvement of 3.7 percentage points in the localisation-quality metric $mAP@0.5:0.95$ over YOLOv5s shows that the predicted boxes fit the true outlines of the targets more closely and can therefore provide a more reliable geometric basis for tenderness grading based on bounding-box size.

4. Conclusions

Aiming at the scale, background and complexity constraints of tea bud detection in natural tea gardens, this paper proposes the improved algorithm YOLOv8n-HQ: a P2 high-resolution detection layer completes the small-object detection scale, a CBAM attention module suppresses background interference, an LRPC re-parameterized convolution module compresses the model, and a Wise-IoU loss improves the bounding-box regression quality. Together the four improvements form a joint optimisation scheme for detection accuracy and model complexity. On the self-built dataset of 15 000 images the algorithm attains 89.9% $mAP@0.5$ and 67.6% $mAP@0.5:0.95$ with 2.05 M parameters, outperforming the compared models in both detection accuracy and model complexity. The ablation study shows that the four improvements contribute independently: the P2 detection layer increases the ability to detect small objects while adding computation, whereas CBAM, LRPC and Wise-IoU raise accuracy while reducing or maintaining the model complexity, so that the four together achieve the

joint optimisation of detection accuracy and model complexity. Future work will extend the dataset to more cultivars and regions to improve generalisation, and will study the precise localisation of picking points.

References

- [1] Xie, S., & Sun, H. W. (2023). Tea-YOLOv8s: A tea bud detection model based on deep learning and computer vision. *Sensors*, 23(14), 6576. <https://doi.org/10.3390/s23146576>.
- [2] Zhang, K., Yuan, B. H., Cui, J. Y., et al. (2024). Lightweight tea bud detection method based on improved YOLOv5. *Scientific Reports*, 14, 30449. <https://doi.org/10.1038/s41598-024-81472-1>
- [3] Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6), 1137–1149. <https://doi.org/10.1109/TPAMI.2016.2577487>
- [4] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., & Reed, S. (2016). SSD: Single shot multibox detector. In *Proceedings of the European Conference on Computer Vision (ECCV)* (pp. 21–37). Springer.
- [5] Redmon, J., & Farhadi, A. (2017). YOLO9000: Better, faster, stronger. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 7263–7271). IEEE.
- [6] Woo, S., Park, J., Lee, J. Y., & Kweon, I. S. (2018). CBAM: Convolutional block attention module. In *Proceedings of the European Conference on Computer Vision (ECCV)* (pp. 3–19). Springer.
- [7] Ding, X., Zhang, X., Ma, N., Han, J., Ding, G., & Sun, J. (2021). RepVGG: Making VGG-style ConvNets great again. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 13733–13742). IEEE.
- [8] Tong, Z., Chen, Y., Xu, Z., & Yu, R. (2023). Wise-IoU: Bounding box regression loss with dynamic focusing mechanism. *arXiv preprint arXiv:2301.10051*. <https://doi.org/10.48550/arXiv.2301.10051>
- [9] Ultralytics. (2023). Ultralytics YOLOv8: A state-of-the-art real-time object detection model. <https://github.com/ultralytics/ultralytics>