Tea Key Points Detection Based On Improved Yolov8

Lijun Zhang[1] and Chenpan Qiu[2]

[1] School of Automation, China University of Geosciences, Wuhan, No. 388, Lumo Road, Hongshan District, Wuhan 430074, China
lijunzh@cug.edu.cn

[2] School of Automation, China University of Geosciences, Wuhan, No. 388, Lumo Road, Hongshan District, Wuhan 430074, China
chenpanqiu@cug.edu.cn

Abstract. This paper introduces a novel method for detecting key points in tea leaves, utilizing an enhanced version of YOLOv8. To improve the accuracy of detecting tea leaf images under varying weather conditions, extensive data collection was conducted, and Gamma correction technology was applied. This approach aimed at optimizing the performance of images captured in different lighting environments. A comprehensive dataset comprising high-quality images was constructed. For the model design, YOLOv8 served as the foundational architecture, with the integration of the CSP (Cross Stage Partial) module and DepGraph pruning method to optimize model parameters and computational efficiency. Experimental results indicate significant enhancements in detection accuracy. The improved model exhibits robust performance, particularly under varying lighting conditions. Furthermore, after fine-tuning, the pruned model maintains high detection performance while significantly reducing computational complexity, demonstrating the effectiveness of the optimization strategies. These findings suggest that the integration of advanced techniques in YOLOv8 can significantly enhance key point detection in tea leaves, offering valuable benefits for agricultural applications.

Keywords: Gamma Correction,Yolov8, Cross Stage Partial, DepGraph.

1 Introduction

Tea harvesting, a time-sensitive endeavor reliant on manual labor [1], faces challenges due to the imprecision of traditional mechanical methods in selecting optimal tea parts [2], often leading to inadvertent leaf damage and economic losses [3]. The accurate grading of tea buds is essential for determining market value, emphasizing the significance of precise picking points during processing . Leveraging deep learning technologies has revolutionized agricultural practices, notably improving the accuracy of tasks such as tea bud detection [4]. In this dynamic landscape, integrating advanced technologies promises to redefine tea harvesting methodologies.

Recent advancements in tea leaf detection have been greatly influenced by the seamless integration of cutting-edge deep learning techniques. Sun et al. [5] introduced a pioneering deep learning-based algorithm tailored specifically for detecting tender tea buds amidst challenging backgrounds. Chen et al. [6] proposed an efficient methodology for discerning fresh tea sprouts, leveraging sophisticated image enhancement techniques alongside SSD methodologies. Li et al. [7] delved into compressed deep learning models, unveiling a novel approach for achieving high-efficiency tea shoot detection. Yang et al. [8] augmented the YOLO-V3 model, providing support for robotic tea picking systems. Wang et al. [9] extended the capabilities of the YOLOv3 model for heightened efficacy in tea leaf detection. Additionally, Li et al. [10] harnessed the power of deep learning in tea leaf detection and classification, enriching the process of assessing tea quality. Liu et al. [11] proposed an innovative YOLOv4-tiny model for real-time tea bud detection, enhancing operational efficiency. Xu et al. [12] modified the Mask-RCNN model for real-time tea bud detection, revolutionizing tea production methodologies. Wu et al. [13] leveraged YOLOv4 with self-attention mechanisms, ensuring robust tea leaf detection in complex settings. Meanwhile, Zhang et al. [14] fused deep learning methodologies with an enhanced SSD model, promising heightened detection accuracy. Li et al. [15] developed a novel tea leaf detection approach anchored upon the advanced YOLOv5 model, pushing boundaries of detection precision further. Additionally, Li et al. [16] devised a tea leaf detection algorithm based on an enhanced Faster R-CNN model, enriching the repertoire of detection techniques. Lastly, Xu et al. [17] made invaluable contributions to real-time tea bud detection through modifications to the Mask-RCNN model, underscoring the pursuit of efficiency and innovation in tea leaf detection methodologies. These seminal studies underscore remarkable progress in tea leaf detection, facilitated by deep learning technologies, paving the way for intelligent and automated systems in the tea industry landscape.

The primary objective of this study was to establish a comprehensive dataset tailored specifically for tea leaf detection, encompassing a wide array of weather conditions. Utilizing advanced techniques such as gamma correction [18], diverse lighting scenarios were meticulously simulated to ensure the robustness of the dataset. Additionally, efforts were made to enhance the performance of the YOLOv8-Pose [19] model by introducing the more efficient CSP [20] module in place of the conventional C2f module. To address the complex architecture of the YOLOv8 model, the sophisticated DepGraph [21] method was adopted for precise model pruning. This refinement process aimed to optimize the model for seamless deployment, effectively bridging the gap between theoretical research and practical implementation in the domain of tea harvesting. The overarching goal extends beyond technological advancement, focusing on streamlining and revolutionizing the tea harvesting process to enhance its efficiency and productivity. Through these efforts, the study seeks to leverage advanced technology to usher in a new era of efficiency and productivity in tea harvesting, ensuring sustainability and prosperity for future generations.Materials and Methods

1.1 Data acquisition and construction of datasets

Spring, spanning from March to May, marks the prime season for tea harvesting. During this period, tea buds are tender and the leaves are lush, ensuring top-notch quality. Tea picked shortly after the Qingming Festival is especially valued for its unique flavor and superior quality, making this time particularly ideal for harvesting.

The recent data collection took place immediately after the Qingming Festival, using a smartphone camera with a resolution of 4096 x 3072 to capture detailed images of tea leaves within the same tea plantation. The study focused on documenting tea leaf conditions under various weather scenarios, including sunny, cloudy, and rainy days, with sessions conducted both in the morning and afternoon. To ensure high data integrity and relevance, photos were meticulously filtered, excluding any that were blurry or lacked a clear subject. The resulting images provide comprehensive visual documentation of tea leaf growth under different weather conditions, offering valuable insights for analyzing tea quality.

However, capturing images during rainy weather posed significant challenges. The tea bushes, covered with rainwater, resulted in pronounced reflections on the tea leaves' surfaces, coupled with subdued overall lighting. Consequently, the quality of the images taken on rainy days was compromised, with many failing to clearly depict the intricate details of the tea leaves. Consequently, a substantial portion of the rainy-day images were excluded during the subsequent screening process.

Following meticulous filtering, we discarded photos that were blurry, excessively reflective, or suffered from inadequate lighting. Ultimately, only 886 high-quality images were retained. These visuals not only document the optimal conditions of the tea leaves during sunny and cloudy weather but also include a select few from rainy days that met quality standards. This dataset provides a rich visual resource for further research endeavors.Table 1 outlines the distribution of retained images according to different weather conditions, underscoring the comprehensive nature of the dataset.

Table 1. Statistics of data under different weather conditions.

Weather	Morning Period	Afternoon Period
Sunny	162	166
Overcast	150	161
Rainy	119	128

During the annotation process of the captured tea leaf photos, stringent standards were followed to ensure high quality and accuracy of the data. Special protocols were implemented for instances where tea leaf targets were partially obstructed by other objects. If the obstruction resulted in less than 60% visibility of the tea leaf target, annotation was refrained from to prevent inaccurate data from affecting the overall annotation quality. For the tea leaf targets, annotations included not only bounding boxes but also six key points: bud_top, bud_bottom, leaf_top, leaf_bottom, stem_top, and stem_bottom. Among the valid images, the quantity of tea leaf targets was

recorded, with some images containing annotations for up to 13 targets. These densely annotated images provided ample samples for model training.

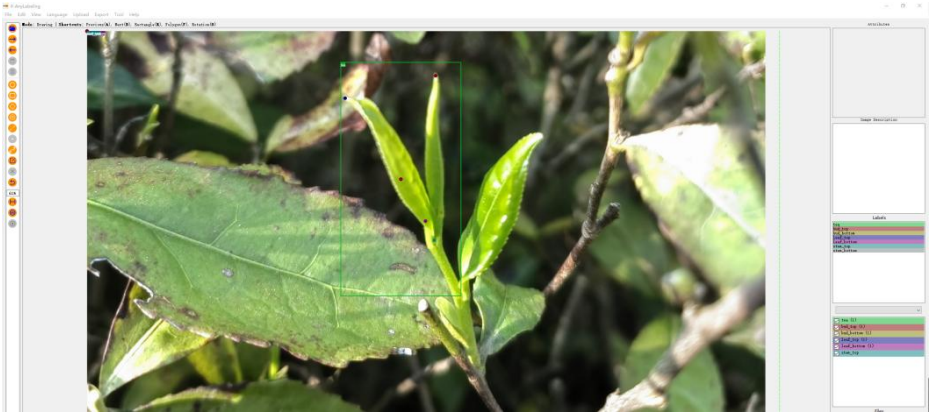

Fig. 1. Annotation process of the tea leaf by X-anylabeling

Upon completion of the annotation process, the annotated images were partitioned into training, testing, and prediction sets according to a 6:2:2 ratio. The training set, comprising 60% of the data, was used for model learning and refinement, while the testing set, accounting for 20%, served to evaluate model performance. The remaining 20% formed the prediction set, used to assess the model's predictive capabilities on unseen data. This meticulous approach to annotation and partitioning ensured the robustness and efficacy of the dataset, facilitating enhanced model training and evaluation processes.

While diversity of weather conditions was considered during tea leaf image capture, the concentration of shooting times led to an uneven distribution of image data under different lighting conditions. Given the relatively weak robustness of deep learning models to outdoor lighting variations, it became necessary to enhance the dataset to improve the model's adaptability to varying lighting conditions. To address this, the dataset was supplemented with tea leaf images under different lighting conditions by adjusting image brightness using Gamma Correction technology. This method applies non-linear transformations to pixel values, allowing precise control over image brightness and contrast by adjusting the Gamma Value. A Gamma Value greater than 1 brightens the image, while a value less than 1 darkens it.This approach enabled random enhancement of the brightness and contrast of images in the dataset, covering a broader range of lighting scenarios. As a result, the sample size of tea leaves under different lighting conditions was expanded, strengthening the model's generalization capability.

Fig. 2. Utilizing gamma correction to simulate various lighting conditions

1.2 Basic model

The backbone network of YOLOv8 Pose is fundamental for extracting multi-scale features from input images, forming the foundation of the entire model. It primarily consists of the Convolutional module (Conv), the C2f module, and the SPPF module. The Conv module employs standard convolution operations and typically comprises convolutional layers, Batch Normalization (BN), and the SiLU (Swish) activation function. Compared to the traditional ReLU activation function, SiLU offers improved gradient flow and training stability. The C2f module is a crucial component of YOLOv8's backbone network, designed to simultaneously process and fuse multi-scale features through multiple convolutional layers of different scales. In comparison to the ELAN module in YOLOv7[22], the C2f module has been optimized to more effectively integrate deep information. The SPPF module (Spatial Pyramid Pooling - Fast) achieves multi-scale feature extraction through stacked 3x3 pooling operations that approximate larger pooling windows, thus reducing computational complexity. SPPF captures context information of various scales, enhancing the model's ability to detect targets.

The neck network of YOLOv8 Pose enhances the detection capability for targets of different scales by further processing and fusing the feature maps outputted by the backbone network. This network employs advanced feature fusion techniques, including Feature Pyramid Network (FPN) and Path Aggregation Network (PAN). FPN is utilized to generate multi-scale feature maps by integrating features from different hierarchical levels in a top-down manner. This process enhances the spatial resolution of high-level feature maps and enriches the semantic information of low-level feature maps, thereby improving the detection capability for small targets. PAN, an extension of FPN, facilitates bidirectional feature information flow by not only transmitting high-level features to low-level ones but also feeding back low-level features to high-level ones. This design further strengthens the fusion effect of feature maps and enhances the model's capability to handle small targets and complex scenes by increasing the expressive power of low-level features.

The head network of YOLOv8 Pose transforms feature maps into detection results using an innovative anchor-free design and specifically predicts human keypoints through dedicated modules. The Pose module serves as the core component of the head network, focusing on pose estimation. This module predicts the coordinates of each target's keypoints through specific convolutional layers, outputting two-dimensional coordinates (x, y) along with confidence scores. This design enables the model to accurately identify keypoints of the human body. In terms of loss function design, YOLOv8 Pose combines boundary box regression and pose estimation. It typically employs the CIoU[23] (Complete Intersection over Union) loss to optimize boundary box prediction, uses cross-entropy loss for target category prediction, and adopts MSE (Mean Squared Error) to measure the gap between predicted keypoints and real keypoints, while also considering the confidence of keypoints. This multitask loss function design enhances the accuracy and stability of the model in pose estimation tasks.

1.3 Architecture of Yolov8 with CSP

The Partial Dense Block, a fundamental element of the CSP module, is inspired by DenseNet's dense connectivity concept. Unlike traditional Dense Blocks, it adopts a partial connectivity strategy to alleviate parameter explosion and computational overhead. Comprising multiple dense blocks, each block contains convolutional layers where inputs are not only received from the preceding layer but also from all preceding layers within the same block. However, unlike conventional Dense Blocks, Partial Dense Blocks establish connections solely among specific dense blocks, strategically limiting excessive parameter growth. This partial connectivity approach effectively mitigates computational complexity while maintaining robust feature expression capabilities.

Partial Transition Layers, integral to the CSP module, play a crucial role in adjusting feature map dimensions similar to the transition layers found in ResNet. However, they employ a partial connectivity strategy. Composed of convolutional and pooling layers, Partial Transition Layers facilitate the resizing of input feature maps to align with the dimensions of subsequent layers. Unlike traditional transition layers, which resize all feature maps uniformly, Partial Transition Layers selectively resize specific feature maps. This targeted approach minimizes computational burden while preserving the diversity and richness of feature maps. By employing this partial connectivity method, feature map resizing operations remain efficient and effective, thereby enhancing the overall performance and efficiency of the CSPNet architecture.

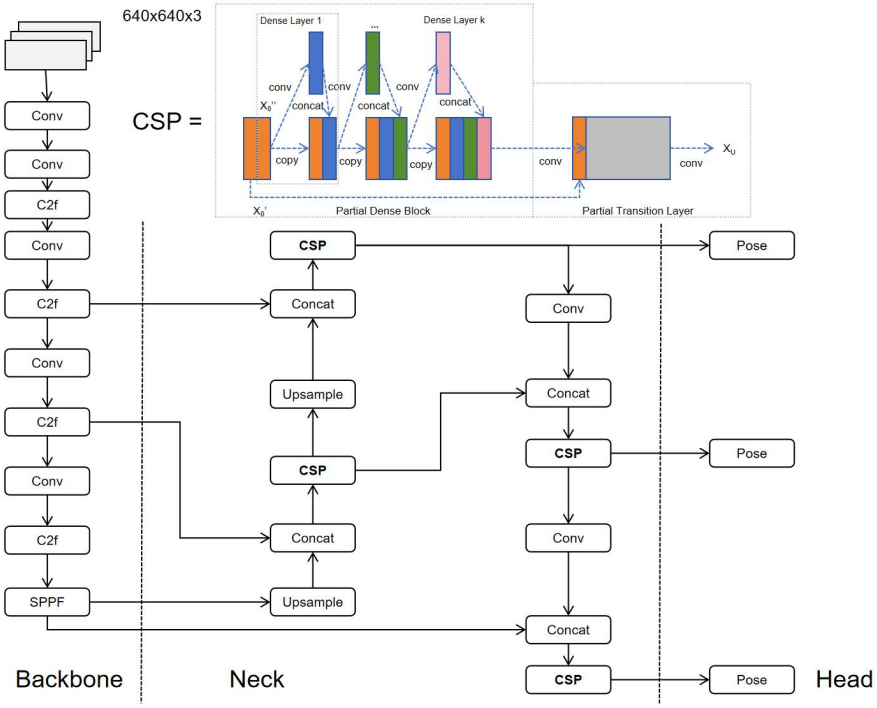

Fig. 3. The architecture of Yolov8-Pose with CSP

Replacing the C2F portion of the neck structure in YOLOv8 with CSP brings several benefits. Firstly, the CSP module introduces a partial connectivity scheme, optimizing parameter usage and computational efficiency. By selectively fusing features across stages, CSP reduces redundant computations, enhancing overall performance. Secondly, CSP enhances feature propagation by strategically merging information from previous layers, leading to richer representations and improved model understanding. Additionally, CSP offers better parameter efficiency compared to C2F, as it controls parameter growth while maintaining or even improving performance. Moreover, the versatility of CSP has been demonstrated across various tasks, ensuring robustness and adaptability in different scenarios. Ultimately, this replacement aligns with the pursuit of improved efficiency and performance in YOLOv8, contributing to its effectiveness in real-world applications.

1.4 DepGraph Prune

In neural networks, layers often have complex parameter dependencies. For example, in YOLOv8's convolutional modules, pruning a convolutional layer requires simultaneous adjustments to its associated batch normalization (BN) layers and residual connections to avoid disrupting the network's architecture and functionality. These dependencies vary across different modules, appearing as residual, connection,

and reduction dependencies. Effective structural pruning must account for these inter-layer dependencies. To this end, the dependency graph (DepGraph) is used to explicitly model these relationships, representing layers as nodes and their dependencies as edges. Constructing a dependency graph involves accurately identifying direct dependencies between layers, enabling precise modeling of the network's structural dependency patterns.

In the dependency graph, we focus on the direct dependency relationships between layers of the neural network and translate these relationships into edges in the graph. This process includes dependency matrix construction, dependency graph generation, and recursive dependency expansion.The dependency matrix G is used to record the dependency relationships between layers in the neural network.In this matrix, $G_{ij}=1$ indicates that layer i depends on layer j meaning that during pruning operations, layer j influences the results of layer i. In this matrix, $G_{ij} = 1$ indicates that layer i depends on layer j, meaning that during pruning operations, layer j affects the output of layer i. The dependency matrix comprehensively records the direct dependencies between layers in the network.

$$D = \{(i, j), (i, k)\} \tag{1}$$

Where the edges (i, j) and (i, k) indicate in the dependency graph that there is an edge from layer i to both layers j and k, indicating that layer i depends on both layers j and k.To comprehensively cover the dependency relationships of all layers in the network, it is necessary to recursively expand the dependencies of each layer. This means starting from each layer and gradually expanding the dependency relationships of its dependent layers until all direct and indirect dependencies are included. This ensures that the dependency graph accurately reflects the mutual dependency patterns of all layers in the network.

When performing Group-level pruning on Yolov8, a criterion for assessing the importance of parameter groups within layers is needed. Traditional single-layer importance criteria perform poorly when dealing with complex dependency relationships because they fail to capture the interplay between layers. To address this issue, a dependency-graph-based group-level importance criterion is utilized.

For each layer i , the importance score I_i of its parameters W_i can be calculated using different criteria, such as:

$$I_i = \| W_i \| p \tag{2}$$

Where $\|W_i\|p$ denotes the p-norm (L2 norm) of the parameters Wi.The group-level importance score I_{group} is the average or weighted average of the importance scores of all layers within the group.

$$I_{group} = \frac{1}{N_k} \sum_{i=1}^{N_k} I_{ki} \tag{3}$$

Where N_k is the number of layers in the k-th group, and I_{ki} is the importance score of the parameters of the i-th layer in the k-th group.

After pruning based on the group-level importance score I_{group} and a predefined prune-rate, the network structure may change. Hence, adjustments are necessary to

maintain its functionality: For pruned layers, adjust their input and output connections to ensure uninterrupted forward propagation paths in the network.Fine-tune the pruned network to restore or enhance its performance compared to before pruning.These steps ensure that the pruned network remains functional and effective in its tasks.

2 Experiments

2.1 Comparative experiment

The software and hardware environment and parametersused in this experiment are shown in table 2.

Table 2. Software and hardware parameters

Accessories	Specification
Operating system	Ubuntu-18.0.4
CPU	12th Gen Intel(R)core(TM)i7-12700F
GPU	NVIDA RTX3080
Development environments	Python3.9, Pytorch2.2.1, CUDA10.8

The YOLOv8 codebase (available at 'Ultralytics/Yolov8') served as the foundational framework, extended and customized according to the specifications outlined in section 2.3. During training, the Stochastic Gradient Descent (SGD) optimizer was employed, initialized with a learning rate of 1E-3, which decayed to a final rate of 1E-5, while incorporating a weight decay of 5E-3 to mitigate overfitting. Momentum was set to 0.8 for the initial three warm-up phases before adjusting to 0.937 for subsequent iterations. The training regimen spanned 300 epochs with a batch size of 8, and horizontal flipping augmentation was disabled.

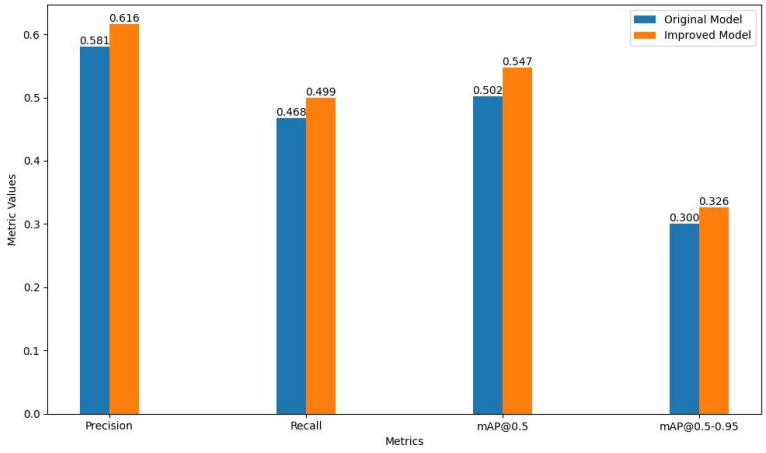

Fig. 4. Model metrics comparison

The evaluation metrics include precision, recall, and mean Average Precision (mAP) at IoU thresholds of 0.5 and 0.5-0.95, serving as performance benchmarks. Comparative analysis reveals a significant improvement across all evaluation metrics in the improved model. Notable enhancements include a 3.5% increase in precision, a 4.5% boost in mAP50, indicating superior detection capability and robustness in the improved model. The observed performance enhancement underscores the effectiveness of optimization strategies in the improved model. The increased precision implies a reduction in false positives, crucial for applications requiring precise target localization.

2.2 Prunning with DepGraph

When using the DepGraph method to prune the YOLOv8 model, it's essential to consider the importance of the initial convolutional modules in feature extraction. Therefore, we have decided not to prune this part to avoid significant accuracy loss. Instead, our focus will be on pruning the remaining parts of the backbone and the neck through convolutional operations. We adopt an iterative approach to gradually prune the model to the specified pruning rate. In each iteration, we fine-tune the model for 50 epochs before proceeding to the next pruning round.

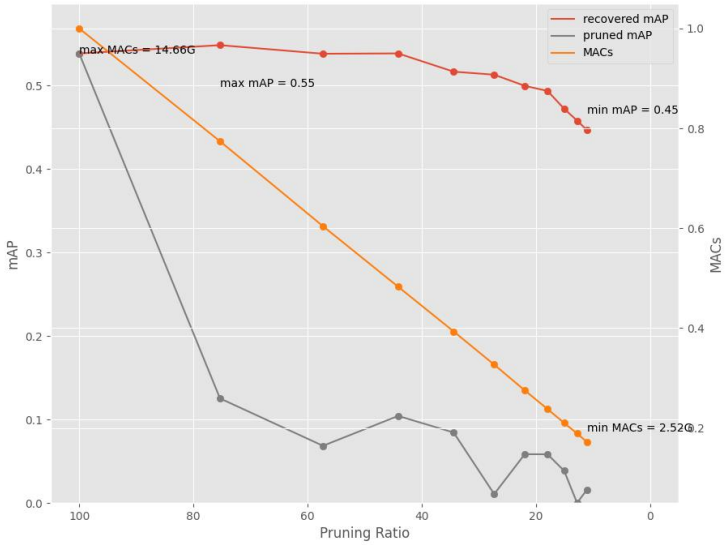

Fig. 5. Model performance with difference pruning ratio

Using MACS (Model Arithmetic Computations) and mAP (mAP50) as metrics to evaluate model performance, it's evident from the graph that after model pruning, there is a noticeable decrease in mAP, signifying a discernible level of performance degradation due to pruning. However, through fine-tuning, the model's performance is effectively restored, underscoring the pivotal role of fine-tuning in optimizing pruned models.On the other hand, the corresponding MACS after pruning exhibits an exponential decrease, indicative of a successful reduction in model computational

complexity, thereby facilitating more efficient inference. Despite the decline in MACS, the model's mAP remains at an acceptable level. Notably, when model pruning reaches approximately 55%, the accuracy of the refined model remains on par with the original model. Furthermore, with pruning reaching around 80%, the model's MAP continues to exceed 0.5, nearing the performance level of the unoptimized model.

Fig.6. The results of Yolov8 with 50% pruning ratio

From the images, it is evident that even after pruning 50% of the model, it still retains commendable recognition capabilities. This indicates that despite reducing the model's complexity significantly, its performance has not noticeably degraded and it continues to accurately recognize target objects. This further validates the positive impact of pruning strategies on optimizing the model while maintaining its performance.

3 Conclusion

Our model design incorporated YOLOv8 as the core architecture, augmented by the CSP module and DepGraph pruning method to refine model parameters and improve computational efficiency. The experimental results highlighted substantial improvements in detection accuracy. The enhanced model demonstrated strong performance, especially under fluctuating lighting conditions. Moreover, fine-tuning allowed the pruned model to sustain high detection performance while markedly reducing computational complexity.These outcomes indicate that advanced techniques integrated into YOLOv8 can significantly bolster key point detection in tea leaves, offering a valuable tool for agricultural applications. The model's capability to handle diverse lighting scenarios and maintain efficiency underscores its practical potential for real-world deployment.

References

1. Lu, Y.J. Debiao, Significance and realization of mechanized picking of famous green tea in China. Chin. Tea 2018, 40, 1–4.
2. Chen J, Chen Y, Jin X, et al. Research on a parallel robot for green tea flushes plucking. Proceedings of the 5th International Conference on Education, Management, Information and Medicine. 2015: 22–26.

3. Yang, H., Chen, L., Ma, Z., Chen, M., Zhong, Y., Deng, F., et al. (2021). Computer vision-based high-quality tea automatic plucking robot using Delta parallel manipulator. Comput. Electron. Agric. 181, 105946. doi: 10.1016/j.compag.2020.105946.
4. Xu, W.; Zhao, L.; Li, J.; Shang, S.; Ding, X.; Wang, T. Detection and classification of tea buds based on deep learning. Comput.Electron. Agric. 2022, 192, 106547.
5. Sun X, Mu S, Xu Y, Cao ZH, Su T. Detection algorithm of tea tender buds under complex background based on deep learning. J. Hebei Univ. 2019;39:211–216.
6. Chen B, Yan J, Wang K. Fresh Tea Sprouts Detection via Image Enhancement and Fusion SSD. J. Control Sci. Eng. 2021;26:13–24.
7. Li Y, He L, Jia J, Chen J, Lyu L, Wu C. High-efficiency tea shoot detection method via a compressed deep learning model. Int. J. Agric. Biol. Eng. 2022;3:159–166.
8. Yang H, Chen L, Chen M, Ma Z, Deng F, Li M. Tender Tea Shoots Recognition and Positioning for Picking Robot Using Improved YOLO-V3 Model. IEEE Access 2019;7:180998–181011.
9. Wang Y, Zhou H, Liu Z, Wang C. Research on tea leaf detection method based on improved YOLOv3 model. J. South China Agric. Univ. 2021;42:37–43.
10. Li Z, Li X, Wang Y, Li C. Deep learning-based tea leaf detection and classification method. J. Tea Sci. 2020;40:152–158.
11. Liu H, Huang X, Zhang L, Zhang Y, Wu Y. An improved YOLOv4-tiny model for real-time tea bud detection. J. Tea. 2022;48:25–32.
12. Xu J, Zhang Y, Wang L, Chen Z, Liu H. Real-time detection of tea bud based on modified Mask-RCNN. J. Tea. 2021;47:36–42.
13. Wu Y, Li H, Zhang W, Jiang Y, Chen L. Tea leaf detection based on YOLOv4 and self-attention mechanism. J. Plant Prot. 2022;49:875–882.
14. Zhang X, Wang T, Liu H, Li S, Chen L. Tea leaf detection based on deep learning and improved SSD model. J. Tea Sci. 2023;43:317–324.
15. Li X, Zhang Y, Wang H, Chen Q, Liu J. Tea Leaf Detection Algorithm Based on Improved Faster R-CNN. J. Tea Sci. 2022;42:103–110.
16. Chen W, Liu X, Zhang L, Li M, Wang Y. Research on Tea Leaf Detection Method Based on Improved YOLOv5 Model. J. Tea. 2023;49:15–21.
17. Xu J, Zhang Y, Wang L, Chen Z, Liu H. Real-time detection of tea bud based on modified Mask-RCNN. J. Tea. 2021;47:36–42.
18. Wikipedia: Gamma correction. https://en.wikipedia.org/wiki/Gamma_correction, last accessed 2024/06/12.
19. Wikipedia: YOLOv8-Pose. https://docs.ultralytics.com, last accessed 2024/03/12.
20. C.-Y. Wang, H.-Y. M. Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I.-H. Yeh, “CSPNet: A new backbone that can enhance learning capability of CNN,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), 2020, pp. 1571–1580.
21. Fang G, Ma X, Song M, et al. Depgraph: Towards any structural pruning[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023: 16091-16101.
22. 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[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2023: 7464-7475.
23. Zheng Z, Wang P, Ren D, et al. Enhancing geometric factors in model learning and inference for object detection and instance segmentation[J]. IEEE transactions on cybernetics, 2021, 52(8): 8574-8586.