Online ISSN: 3104-9338
Print ISSN: 3104-932X

Tea detection based on YOLOv8 and PyQt5

Tingting Yang

School of Electronic Engineering, Jiangsu Ocean University, Lianyungang; China

Received: July 10, 2025

Revised: July 15, 2025

Accepted: July 19, 2025

Published online: August 3, 2025

To appear in: *International Journal of Advanced AI Applications,* Vol. 1, No. 5 (September 2025)

* Corresponding Author: Tingting Yang (2635225549@qq.com)

Abstract. Under the trend of intelligent transformation in modern agriculture, the contradiction between the efficiency and quality of tea picking urgently needs to be resolved. This study conducts tea detection optimization based on the YOLOv8 algorithm and constructs a complete technical path from theory to practice. The study first analyzes the network architecture of YOLOv8, combines the characteristics of the One-Stage algorithm, and clarifies its advantages in real-time tea detection. Through training and deployment on public datasets, the detection accuracy has been improved by 6.7% compared to the original YOLOv8, providing algorithmic support for mechanized picking. For the complex environment of tea gardens, a module optimization strategy is proposed: introducing a fusion attention module to generate a new C2fM module, and introducing asymmetric convolution to generate the ACBSPPF module. Through ablation experiments and cross-validation, the optimization effect was verified using mAP and FPS as indicators. The model has reached the industry-leading level in terms of real-time performance and accuracy. Research shows that the optimized YOLOv8 algorithm effectively solves the problem of tea detection. Finally, a tea detection system is designed using PyQt5, providing a feasible solution for the industrialization of intelligent picking technology.

Keywords: YOLOv8; PyQt5; Tea detect; Attention fusion; Asymmetric convolution.

1. Introduction

Tea, as one of the world's three major beverages, has always attracted much attention in terms of its picking techniques and mechanized processing [1]. At present, the main methods of tea picking rely on manual picking and mechanical picking [2]. Manual tea picking has high labor costs and relatively low efficiency [3]; The "one-size-fits-all" mechanical picking method

results in uneven quality of the picked tea buds [4]. With the rapid development of deep learning object detection technology, the efficient picking of tea is the development trend of tea picking technology research, and the recognition and detection technology of tea is the key to the research [5].

In the context of the deep integration of artificial intelligence and agriculture, object detection technology has become a key driving force for enhancing the intelligent level of agricultural production. As a new-generation real-time object detection framework, YOLOv8, with its lightweight network architecture and advanced feature extraction mechanism, has achieved a significant improvement in inference speed while maintaining high-precision detection, demonstrating outstanding performance in fields such as industrial quality inspection and security monitoring. However, when it is applied to the tea-picking scenario, it faces technical challenges such as the diversity of leaf shapes, complex lighting environments, and overlapping occlusion, which urgently require targeted optimization strategies.

This study focuses on the application bottleneck of YOLOv8 in tea target detection. By improving the network structure, optimizing the data preprocessing process and adjusting the model training strategy, a highly robust tea detection model is constructed. On the one hand, by designing a multi-scale feature fusion module and an adaptive attention mechanism, the model's recognition ability for tea at different growth stages is enhanced; On the other hand, a large-scale dataset is constructed in combination with the actual environment of the tea garden, and techniques such as data augmentation and transfer learning are applied to enhance the generalization performance of the model. The research results will provide core algorithmic support for intelligent tea-picking equipment, helping to promote the transformation and upgrading of the traditional tea industry towards precision and efficiency.

Chapter Introduction:

This paper focuses on the research of tea detection based on YOLOv8 and PyQt5. The overall technical route is as follows: Firstly, analyze the network architecture of the YOLOv8 algorithm and the characteristics of the One-Stage algorithm to clarify its advantages in real-time tea detection; Secondly, a module optimization strategy is proposed for the complex environment of the tea garden. The C2fM module and the ACBSPPF module are constructed respectively by integrating the attention mechanism and introducing asymmetric convolution. Subsequently, the effectiveness of the improved algorithm was verified through comparative experiments and ablation experiments, and the model performance was evaluated with mAP and FPS as the core indicators. Finally, a tea detection system was designed based on PyQt5 to realize the

implementation of the algorithm from theoretical optimization to practical application.

The main contents of the remaining parts are as follows:

Part 2 "Theoretical Basis" Systematically expound the core concept of object detection and distinguish the differences between Two-Stages and One-Stage algorithms (taking Faster R-CNN and the YOLO series as examples) This paper focuses on analyzing the network structure of YOLOv8 (input, backbone network, neck, output) and the working principles of key modules (such as C2f, SPPF) to provide theoretical support for subsequent improvements.

Part 3 "Algorithm Improvement Methods": A detailed introduction to the design ideas of the two core improvement modules

C2fM module: Integrates the MSA multi-head self-attention mechanism into the Bottleneck module, adds residual connections, and enhances the ability to capture multi-source features (appearance, texture, etc.) of tea.

ACBSPPF module: It introduces asymmetric convolution (3×1, 3×3, and 1×3 convolution kernel combinations) to replace traditional convolution, reducing the computational load while enhancing adaptability to multi-scale tea targets.

Part 4 "Algorithm Experiments": Experiments are conducted based on public tea datasets (including four types of tea with different freshness levels from T1 to T4), including:

Dataset and experimental environment description (hardware configuration, parameter Settings);

Comparative experiment: Compared with models such as YOLOv8, SSD, and Faster R-CNN, verify the advantages of the improved algorithm in terms of mAP, FPS, and parameter scale;

Ablation experiment: Verify the optimization effects of C2fM and ACBSPPF modules individually and in combination, and quantify the improvement in detection accuracy, such as a 6.7% increase in mAP.

Part 5 "Tea Detection Interface": This section introduces the design of a visualization system based on PyQt5, including interface layout (image selection, detection, and result export functions), operation process, and real-time detection effect display, to facilitate the application of algorithms.

Part 6 "Conclusion": Summarize the core contributions of the improved algorithm, reaffirm the effectiveness of the C2fM and ACBSPPF modules, explain the system's promoting effect on the industrialization of intelligent tea picking, and look forward to future optimization directions, such as adaptation to complex occlusion scenarios.

2. Theoretical basis

2.1. Concepts related to object detection

In the field of computer vision, object detection algorithms can be classified into Two mainstream paradigms, two-stages and One-Stage, based on the differences in detection processes. The two show significant differences in detection mechanisms and performance.

The Two-Stages object detection algorithm, represented by the R-CNN series, follows the detection logic of candidate selection first and then fine-tuning. Take Faster R-CNN [6] as an example. This algorithm integrates the candidate Region generation process into the Network architecture by introducing the Region Proposal Network (RPN), replacing the traditional selective search method. Specifically, the model first performs feature extraction on the input image in the backbone network. Then, RPN generates a series of candidate regions that may contain the target based on the feature map. Finally, the candidate regions are feature aligned through operations such as ROI Pooling, and bounding box regression and category classification are completed in the head network. This type of algorithm, with its phased processing strategy, can fully explore the details of target features and demonstrate high detection accuracy in complex scenarios. However, the high number of parameters and long reasoning time brought about by multi-stage computing limit its application in scenarios with high real-time requirements.

In contrast, One-Stage object detection algorithms such as the SSD and YOLO series adopt an end-to-end direct regression mode, transforming object detection into a direct prediction problem of bounding box coordinates and category probabilities [7-8]. Take YOLOv8 as an example. The model rapidly extracts image features through a lightweight backbone network, uses a neck network for multi-scale feature fusion, and ultimately directly outputs the position and category information of the target on the detection head. This paradigm significantly reduces the computational complexity by minimizing the intermediate candidate region generation steps, achieving millisecond-level inference speed and meeting the real-time scenario requirements of industrial quality inspection, autonomous driving, etc. Although there are accuracy shortcomings in small target detection and complex background recognition, with the development of network structure optimization and data augmentation technology, the detection accuracy of the One-Stage algorithm is gradually approaching that of the Two-Stages algorithm, demonstrating strong application potential.

2.2 Principles of the YOLOv8 Algorithm

The network structure of YOLOv8 mainly consists of four parts: input, backbone, neck and output [9]. In the input section, image data is usually received, which can come from high-resolution images collected by devices such as drones, ground robots or fixed cameras. The main part is responsible for extracting the features of the input image and usually adopts structures such as CSPDarknet53. For instance, drawing on the idea of VGG [10], a large number of 3×3 convolution is used, and the number of channels is doubled after each pooling operation. It also drew on the idea of ResNet [11], extensively using residual connections in the network, which alleviated the problem of vanishing gradients during training and made the model more convergent. The neck part is responsible for fusing the multi-scale features extracted by the backbone network, usually adopting structures such as PANet. For instance, three feature maps of different scales are concatenated through upsampling, and after processing, feature maps of different scales are output to the output part [12]. The output section is responsible for predicting the target box and category probability. It usually adopts a three-layer prediction structure, with each scale prediction feature predicting targets of different range sizes to improve the detection accuracy of the model.

The network structure of YOLOv8 mainly consists of three parts: the Backbone, the Neck and the detection Head. The main part is responsible for extracting the features of the input tea image, and usually adopts modules such as CBS, C2f and SPPF. The neck part achieves the fusion of multi-scale features, and combined with the precise prediction of the output part, it can accurately determine the position and category of the tea leaves.

In terms of performance, the data augmentation strategy of YOLOv8 has greatly enhanced the accuracy and robustness of detection. Data augmentation methods such as color perturbation and spatial perturbation enable the model to adapt to images under different lighting conditions, environmental changes, and from various perspectives and postures. For instance, in actual tea detection, the intensity and Angle of light at different times can have a significant impact on tea images. By adjusting the hue, saturation and brightness of the images, the model can better cope with these changes. Random cropping, scaling, rotation and flipping operations increase the diversity of the training data, enabling the model to learn more characteristics of tea types under different conditions, thereby improving the accuracy of the algorithm in tea detection. The network structure of YOLOv8 is shown in Figure 1.

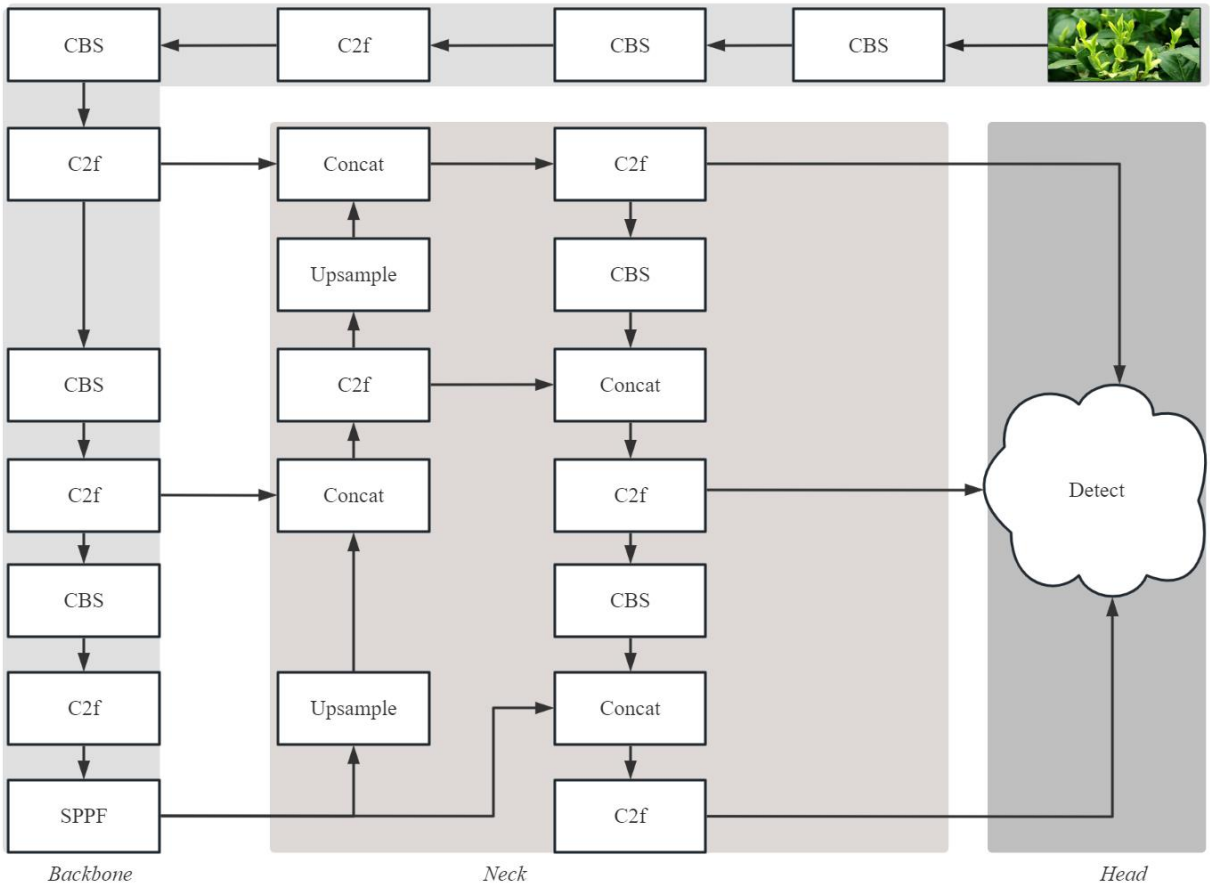

Figure 1. The network structure of YOLOv8.

3. Algorithm improvement methods

Improving the algorithm flow of YOLOv8 in tea detection can significantly enhance its performance. In the improvement of attention fusion, the detection accuracy and generalization ability of the C2f module are enhanced by fusing the attention mechanism into the C2f module.

3.1 C2fM module

By adding a attention mechanism to the Bottleneck to optimize the C2f module and form a new module C2fM, the performance of the C2f module is further enhanced, achieving an improvement in feature extraction capabilities. Compared with the model performance before and after the improvement, there is a significant improvement, verifying that the performance of the YOLOv8 algorithm model can be further enhanced.

The C2f module is shown in Figure 2, which includes 1 Split operation, 2 CBS modules and n Bottleneck modules. By observing the module structure, it can be found that the feature extraction capability of the C2f module benefits from the feature maps containing multiple scales after its Split operation. Therefore, when integrating the attention mechanism, it is necessary to retain the advantage of this multi-scale fusion.

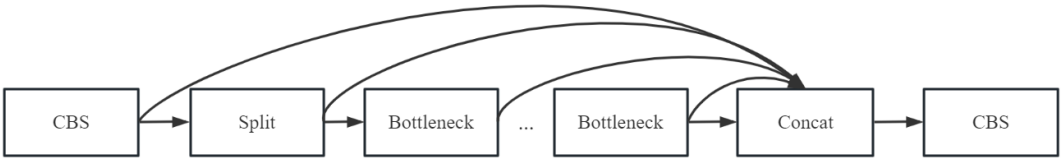

Figure 2. C2f module.

The Bottleneck module in the multi-branch process of the C2f module plays a crucial role. The Bottleneck module is essentially a residual network. The main branch contains two CBS modules. Finally, it performs Add cumulative calculation with the initial uncalculated feature number to ensure that the effect does not decline after multiple layers of convolution. The Bottleneck module is shown in Figure 3.

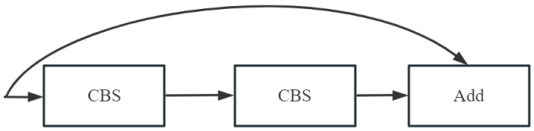

Figure 3. Bottleneck module.

In conclusion, if the attention mechanism is integrated outside the multi-scale of the module, only the result of one feature map can be improved in terms of attention. To maximize the performance improvement of the C2f module as much as possible in this study, it is considered to integrate the attention mechanism on all Bottleneck modules.

After the attention mechanism played a significant role in the field of object detection, various attention mechanisms were successively introduced. Including CBAM (Convolutional Block Attention Module) attention mechanism [13], SE (Squeeze-and-Excitation) attention mechanism [14], ECA (Efficient Channel Attention mechanism [15], SA (Self-Attention) self-attention mechanism [16], and MSA multi-head attention mechanism [17], etc.

The CBAM Attention mechanism. This mechanism Module integrates the Channel Attention Module CAM (Channel Attention Module) and the Spatial Attention Module SAM (Spatial Attention Module), and simultaneously adopts two strategies: global average pooling and global maximum pooling. It can effectively prevent information loss [18]. The SE attention mechanism adaptively recalibrates the channel characteristic response by explicitly establishing the interdependence between channels [19]. The ECA attention mechanism module can achieve significant performance gains by adding only a few parameters. Moreover, this module can adaptively adjust the channel feature weights, effectively capture the channel relationships between images, and enhance the feature expression ability [20]. The self-attention mechanism can not only capture the global feature information of the data, but also the feature information

among the same set of data vectors, and identify the important trends in the temperature changes of grain storage [21]. The MSA multi-head self-attention mechanism divides the Query, Key, and Value of SA into multiple smaller parts, each corresponding to a different "head", and executes multiple self-attention layers in parallel. Each self-attention layer computes independently, enabling the model to capture information in different subspaces [22]. The comparison results of these several attention mechanisms are shown in Table 1.

Table 1. Comparison of Attention Mechanisms

Name	CBAM/SE/ECA	Self-Attention	MSA
Information capture	Channel/space: local/global	Single global	Subspace parallel semantic dependencies
Feature expression	Channel/spatial optimization	Global association	Multi-head diverse features
Computational efficiency	CBAM/SE:high load ECA:lightweight	Exponential with length	Better than single-head
Param Efficiency	CBAM/SE:moderate ECA:few	Single space-focused	Shared matrix independent

To further compare the effects of different attention mechanisms in tea detection, this study conducted an ablation experiment. The results of the ablation experiment are shown in Table 2.

Table 2. Detection effects of different attentions.

Experiment	mAP(%)	FPS	Parameters	GFLOPs
YOLOv8	86.4	110	**3157200**	**8.9**
YOLOv8+MSA	**88.9**	**112**	3182344	9.2
YOLOv8+CBAM	87.1	108	3335600	9.9
YOLOv8+SE	86.7	105	3185400	10.2
YOLOv8+ECA	86.8	109	3184800	9.1
YOLOv8+Self-Attention	88.0	109	4810500	15.8

By comparison, it can be found that the mAP value of the MSA multi-head self-attention mechanism is the highest, and the FPS is also the highest. At the same time, it adds the fewest parameters among several alternative attention mechanisms. This means that the MSA multi-head self-attention mechanism has more advantages in tea detection, with higher recognition accuracy and better real-time performance. The MSA multi-head self-attention mechanism is adopted in tea detection, and its characteristics highly meet the detection requirements. It can process the appearance, texture, composition and other multi-source heterogeneous features of tea leaves in parallel through multiple heads, integrate information of different scales, and overcome the limitations of one-dimensional optimization of other mechanisms. Its parallel computing and parameter sharing functions ensure high inference speed while reducing redundant parameters, making it suitable for real-time operation on industrial assembly lines or portable devices. Multi-subspace attention can simultaneously capture global and local features,

enhancing the anti-interference ability in complex scenarios. In addition, multi-head independent computing can flexibly adapt to the multi-task requirements such as classification, positioning, and regression in tea detection, avoiding feature conflicts. Therefore, it becomes an ideal choice for tea testing.

The output of MSA is as shown in Equation (1), where C is the Concat function and X is the input feature; X_1 to X_n are used to divide X into n smaller parts; H(Xi) represents the self-attention output at the i-th head and W^0 is the linear transformation before the output.

$$f(x) = C(H(X_1), H(X_2), \cdots, H(X_n))W^0 \quad (1)$$

The process of the MSA multi-head self-attention mechanism is shown in Figure 4.

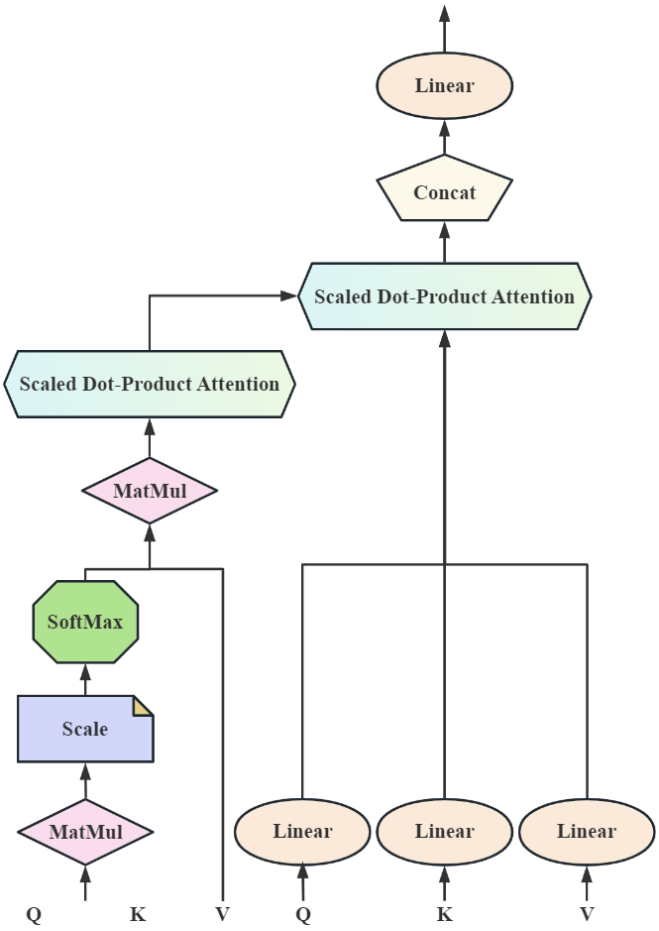

Figure 4. Multi-head attention mechanism.

This study proposes to Add the MSA multi-head self-attention mechanism into the Bottleneck module and simultaneously add a residual connection branch in the first CBS module to connect to the ADD process, forming a new module BottleneckM. The BottleneckM module is shown in Figure 5.

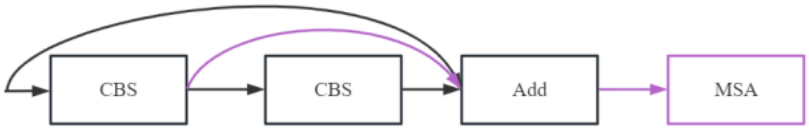

Figure 5. BottleneckM module.

The BottleneckM module is then integrated with the C2f module to form the new C2fM module, as shown in Figure 6.

The C2fM module, based on the C2f module, integrates the MSA multi-head self-attention mechanism with the Bottleneck module and adds a residual branch to the Add process in the first CBS module. Its advantage lies in achieving the capture of semantic information in different subspaces through the parallel processing of multi-source heterogeneous features of tea by MSA multi-heads, and avoiding feature loss by combining residual connections. It not only retains the multi-scale fusion advantages of the original module, but also enhances the expression ability of complex features such as the appearance and texture of tea.

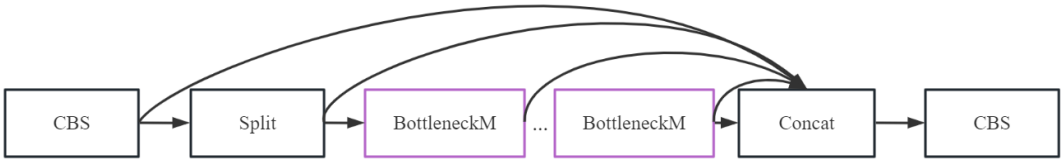

Figure 6. C2fM module.

3.2 ACBSPPF module

The SPPF (spatial pyramid pooling fusion) structure adopted in YOLOv8 is improved by introducing a feature fusion module on the basis of the SPP structure, enhancing the perception ability and detection performance of the model [23]. The SPPF module is shown in Figure 7. There is one convolution operation after input and one before output. The intermediate process includes three Max pooling operations and concatenates feature maps of different sizes together through Concat. Similar to the C2f structure, the SPPF module also has the advantage of multi-scale fusion. Meanwhile, compared with the traditional spatial pyramid pooling, SPPF has a faster feature processing speed while maintaining similar performance. It reduces the computational load and processing time by performing a faster pooling operation on the feature map.

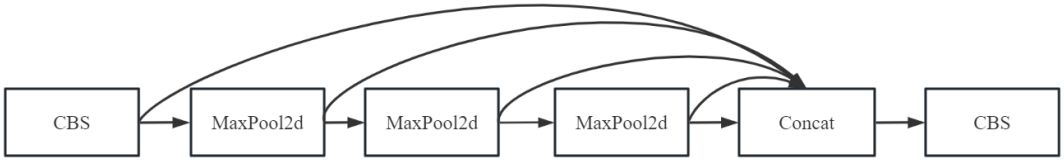

Figure 7. SPPF module.

The significant advantage of the SPPF structure lies in its ability to adaptively fuse multi-scale feature information, thereby endowing it with outstanding feature extraction capabilities. However, experiments show that when the SPPF module deals with occlusion scenes or small object detection tasks, such as tea detection tasks, it overly focuses on obtaining local image information, thereby causing partial loss of global information and ultimately adversely affecting the accuracy of model detection. Further optimization is urgently needed.

The Simplified Spatial Pyramid Pooling-Fast (SimSPPF) module, compared with the traditional Spatial Pyramid Pooling-Fast (SPPF) module, can achieve efficient utilization of computing resources in object detection tasks, especially when dealing with high-resolution images [24]. After testing and verification, a single CBR is 18% faster than CBS. By optimizing the activation function, SimSPPF reduces the computational burden of the model. This is not only extremely beneficial for deploying the model on resource-constrained devices, but also significantly improves the computing efficiency in server or cloud environments. This structural optimization enables the model to maintain high performance while also being more suitable for various computing environments. Its structure is shown in the following figure.

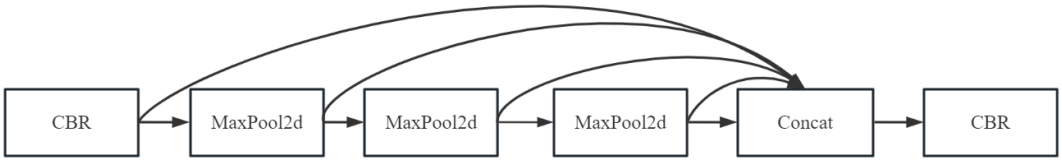

Figure 8. SimSPPF module.

Compared with SPPF, SimSPPF reduces the computational burden of the model, but it is still based on Conv convolution operations and still requires a large amount of computational space. Therefore, it is considered to replace the CBR module with a more lightweight module.

In the current era of continuous evolution of computer vision technology, asymmetric convolution, as a key technology for enhancing the efficiency and accuracy of models, is playing an increasingly important role. The front-end structure of Asymmetric Convolution includes the Asymmetric Convolution Block (ACB), the Fully Connected Layer (FC), and the activation function GELU.

The sampling structure of asymmetric convolutional layers includes three different Conv convolution types, namely, convolution kernel size of 3×1, convolution kernel size of 3×3, and convolution kernel size of 1×3. Compared with traditional symmetrical convolution, this method can enhance the influence of local salient features and has achieved success in many computer vision tasks [25].

The asymmetric convolution is shown in Figure 9. The front-end structure flow on the left

clearly presents the complete path of data from input to passing through the asymmetric convolution layer, the fully connected layer, and finally being output through the activation function. The asymmetric convolution sampling structure on the right visually presents the combination methods of three different convolution kernels.

After the SPPF module is combined with asymmetric convolution to replace the conventional convolution module, a new module ACBSPPF is generated, as shown in Figure 10. The CBS modules at both ends are replaced with ACB modules.

This structure, with almost no increase in computational burden, replaces and extends the single-branch convolution with the main branch and multiple secondary branches, and uses lightweight convolution for convolution replacement, effectively enhancing the ability to extract multi-scale features of images. Meanwhile, through the combination design of expansion rates, it ensures the continuity of the receptive field of the concatenated feature map and multi-scale feature extraction of modules can be achieved by introducing lightweight convolution, adjusting the number of branches, and other methods.

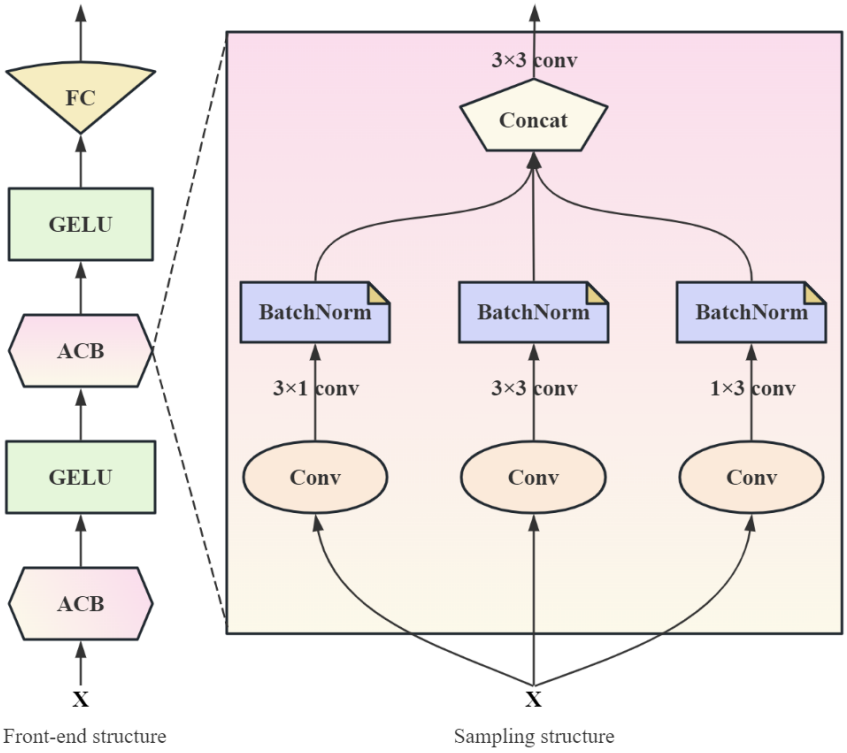

Figure 9. Asymmetric convolution.

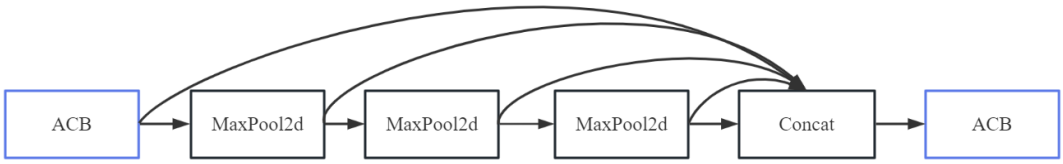

Figure 10. ACBSPPF module.

3.3 Improved algorithm structure

This study focuses on improving two new modules: The MSA multi-head self-attention mechanism is added to the Bottleneck module, and a residual connection branch is added to the first CBS module to connect to the Add process, forming a new module BottleneckM. Further, the BottleneckM module is replaced by the C2f module to fuse into the new module C2fM. In another key improvement, asymmetric convolution technology was introduced to upgrade the SPPF module. By using heterogeneous combinations of 3×1, 3×3, and 1×3 convolution kernels and integrating them with the spatial pyramid pooling mechanism, the ACBSPPF module was generated, significantly reducing the computational load while enhancing adaptability to tea targets of different scales. The above improvements effectively reduced the number of model parameters, simultaneously enhanced the detection speed and accuracy, and significantly improved the model's positioning accuracy and recognition ability for tea.

The new YOLOv8 model after combining the C2fM module and the ACBSPPF module is shown in Figure 11.

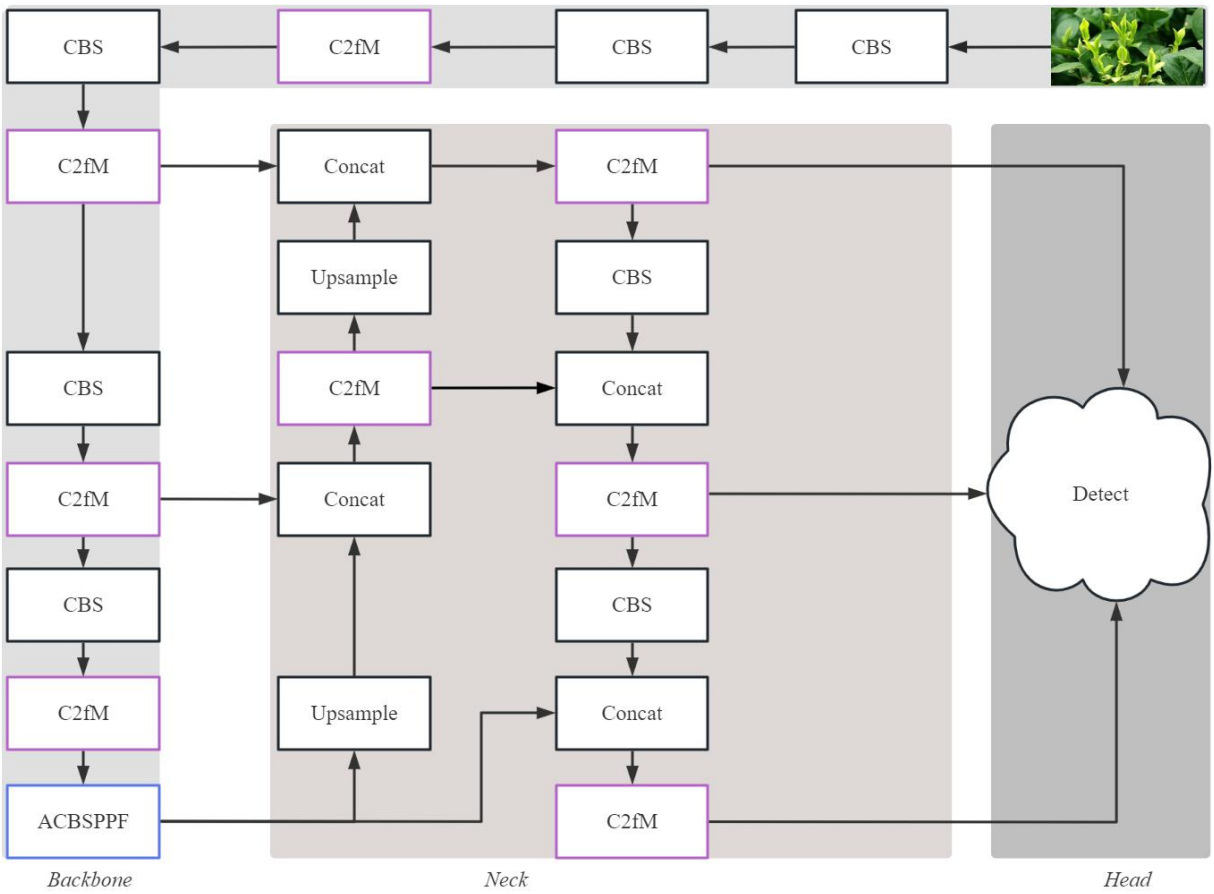

Figure 11. The improved YOLOv8 model.

4. Algorithm experiment

4.1 Dataset

The tea dataset is a publicly available dataset published by Kabir M and [26] et al. The dataset contains 2,208 original images. The dataset is systematically divided into four different categories (T1:1-2 days, T2:3-4 days, T3:5-7 days, T4: more than 7 days).

The category description of this dataset is shown in Table 3.

Table 3. Description of Tea Types.

Type	Days	Description	Quantity
T1	1-2	Tea processed within 48 hours after picking has the highest freshness and aroma quality	562
T2	3-4	Tea picked within 72 to 96 hours is of excellent quality and retains its flavor well	615
T3	5-7	Tea made within 5 to 7 days after picking will have a moderate decline in flavor and aroma	508
T4	7+	Tea picked for more than 7 days will have a significantly reduced essential oil content and is not recommended for consumption	523

The images corresponding to different types are shown in Figure 12.

Figure 12. Images corresponding to different types

Some of the images in the dataset are shown in Figure 13. It can be seen that there are various tea images of different sizes and appearances in the dataset.

Figure 13. Images of tea in the dataset.

The algorithm operation environment of this study is based on the dynamic cloud network server. The detailed environment configuration selected is shown in Table 4.

Table 4. Environmental Configuration.

Name	Configuration
Operating system	Ubuntu18.04
GPU	NVIDIA RTX 4070 Ti
CUDA	11.7
Python	3.9.16
PyTorch	1.13.1

4.2 Parameter Settings

To ensure the efficient advancement of grid training, it is necessary to systematically configure the parameters of the network model. During the training phase, the SGD optimizer is adopted. The core hyperparameters are set as follows: The initial learning rate is set to 0.01, combined with a momentum value of 0.937 to accelerate convergence and reduce oscillations. At the same time, a weight decay coefficient of 0.005 is used to suppress overfitting. In the data preprocessing stage, the size of all images is unified to 640×640. In terms of training strategy, set 100 epochs to complete the full data iteration, and adjust the batch size to 16 to balance computational efficiency and memory consumption.

4.3 Comparative Experiment

To scientifically evaluate the performance of different object detection models on the tea dataset, this study selected four mainstream models, namely YOLOv8, SSD, Faster R-CNN, and RT-DETR, to conduct control experiments. By strictly controlling the training environment and parameter Settings, it is ensured that each model operates under the same experimental conditions to eliminate the interference of external variables. Ultimately, based on the quantitative comparison data shown in Table 5, a systematic analysis was conducted on the differences in detection accuracy, speed and generalization ability of each model for tea targets, providing a reliable basis for model selection in tea detection tasks.

Table 5. Experimental Comparison Data Table of Different Models.

Model	mAP (%)	FPS	Parameters	GFLOPs
YOLOv8n	**86.4**	**110**	**3157200**	8.9
SSD[27]	64.1	23	4724541	26.7
Faster R-CNN[28]	75.5	20	105673457	65
RT-DETR[28]	84.5	88	32970476	108.3
YOLOv3[29]	77.4	61	65252682	154.7
YOLOv5s[29]	78.9	109	19045245	15.8
YOLOv7[29]	81.2	78	37297025	103.2
YOLOv9[30]	84.2	89	22345245	44.7
YOLOv11[30]	85.7	85	2,297,334	**6.3**

It can be seen from Table 4 that the YOLOv8 model has the highest recognition accuracy rate for the tea dataset, reaching 86.4%. At the same time, the reasoning time FPS for each

image is also the highest, reaching 110 frames. The parameters are also the fewest, at 3,157,200, which indicates that the YOLOv8 model has more advantages when used for real-time tea detection.

4.4 Ablation Experiment

To verify the effectiveness of the two improvement points proposed in this study, ablation experiments based on YOLOv8 and the two improvement points were conducted. The results of the ablation experiments on the tea dataset are shown in Table 6.

Table 6. Results of the ablation experiment on the tea dataset.

Experiment	C2fM	ACBSPPF	mAP (%)	FPS	Parameters	GFLOPs
1	×	×	86.4	110	**3157200**	8.9
2	✓	×	88.9	112	3182344	9.2
3	×	✓	89.2	108	3195971	9.4
4	✓	✓	**93.1**	113	3213546	**10.1**

As can be seen from Table 5, after integrating the two improvement points, the mAP value of the improved YOLOv8 model is the highest, that is, the accuracy rate of tea detection is the highest. At the same time, the parameters of the improved model increased by 1.8% compared to the YOLOv8 model, but the mAP increased by 6.7% compared to before the improvement. The ablation experiments fully proved that the improvement of the model structure and the optimization of the loss function are very effective in improving the performance of YOLOv8 in tea detection.

5. TensorRT deploys the PyQt5 detection system

5.1 Tea detection interface

After the design of the tea detection algorithm based on YOLOv8 is completed, if the detection and experimentation do not involve a friendly user interface, it is rather difficult for ordinary users to apply the detection algorithm to actual tea picking activities. Therefore, this study specially designed a tea detection system based on YOLOv8 and PyQt5.

By using the Designer tool of PyQt5, developers can efficiently build interface layouts through intuitive drag-and-drop methods, greatly enhancing development efficiency [31].

The initial interface of the system is shown in Figure 14, which includes three operation buttons: image selection, image detection, and results everywhere.

After selecting the image, the detection system will automatically load and display it in the left area of the tea image to be detected, while the right area will show "Detection in progress..." As shown in Figure 15.

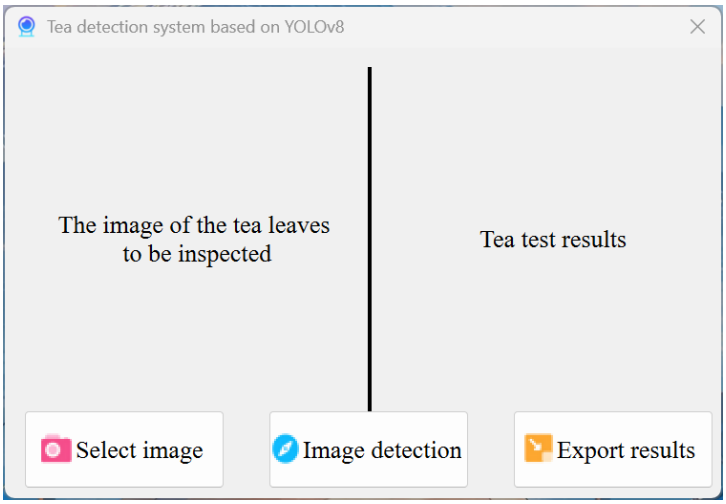

Figure 14. Initial interface of the tea detection system.

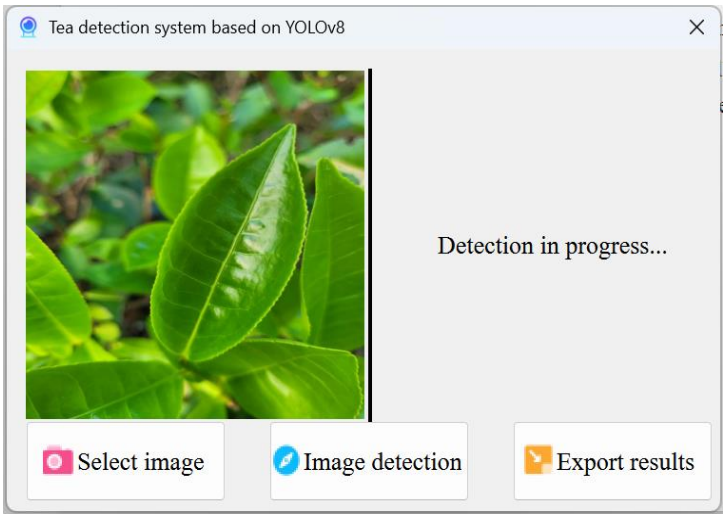

Figure 15. Select the image.

After the system detection is completed, it will automatically display the image result of the detected tea in the right area, as shown in Figure 16.

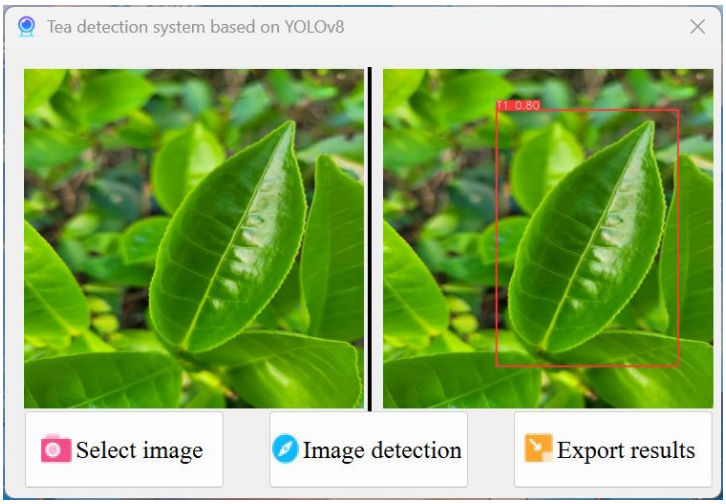

Figure 16. Detection completed.

5.2 System deployment

5.2.1 Deployment architecture design

This section mainly accelerates the tea detection system that has been designed in Section 5.1 through TensorRT, which will be accelerated

The subsequent model was deployed on the NVIDIA Jetson Nano embedded development board.

Gradient explosion is a common problem in the training process of neural networks. When the gradient update value is too large, the repeated multiplication operations of each layer of the network in backpropagation will cause the gradient to grow exponentially. To ensure the stable operation of forward and backward propagation, it is usually necessary to use high-precision data types (such as FP32 or FP64) to guarantee that the minor changes in each gradient update can be precisely represented [32].

In contrast, the model inference stage only involves forward computation and does not require backpropagation. Therefore, the sensitivity of the reasoning results to data accuracy is relatively low. Taking advantage of this feature, inference optimization can be carried out using low-precision data types, such as FP16 or INT8. Using low-precision data not only significantly reduces the memory usage of the model but also accelerates the computing process. It is particularly suitable for deployment on embedded devices with limited computing resources, enhancing the flexibility and practicality of the model.

In response to the distributed deployment requirements of the tea garden scenario, a three-level architecture system is designed.

(1) Edge perception layer: Deployed at the tea-picking robot terminal, it includes multimodal sensors (RGB cameras, depth cameras) and edge computing units (Jetson AGX Xavier), responsible for image acquisition and real-time inference;

(2) Regional aggregation layer: Based on 5G/CBRS wireless private network, data from multiple edge devices are aggregated to the tea garden edge server (Intel Xeon E-2274G + NVIDIA T4 GPU) to achieve local data processing and model update;

(3) Cloud decision-making layer: Deployed in Alibaba Cloud GPU clusters, it is responsible for global model training, data management, and task scheduling.

The three-level architecture system is shown in Figure 17.

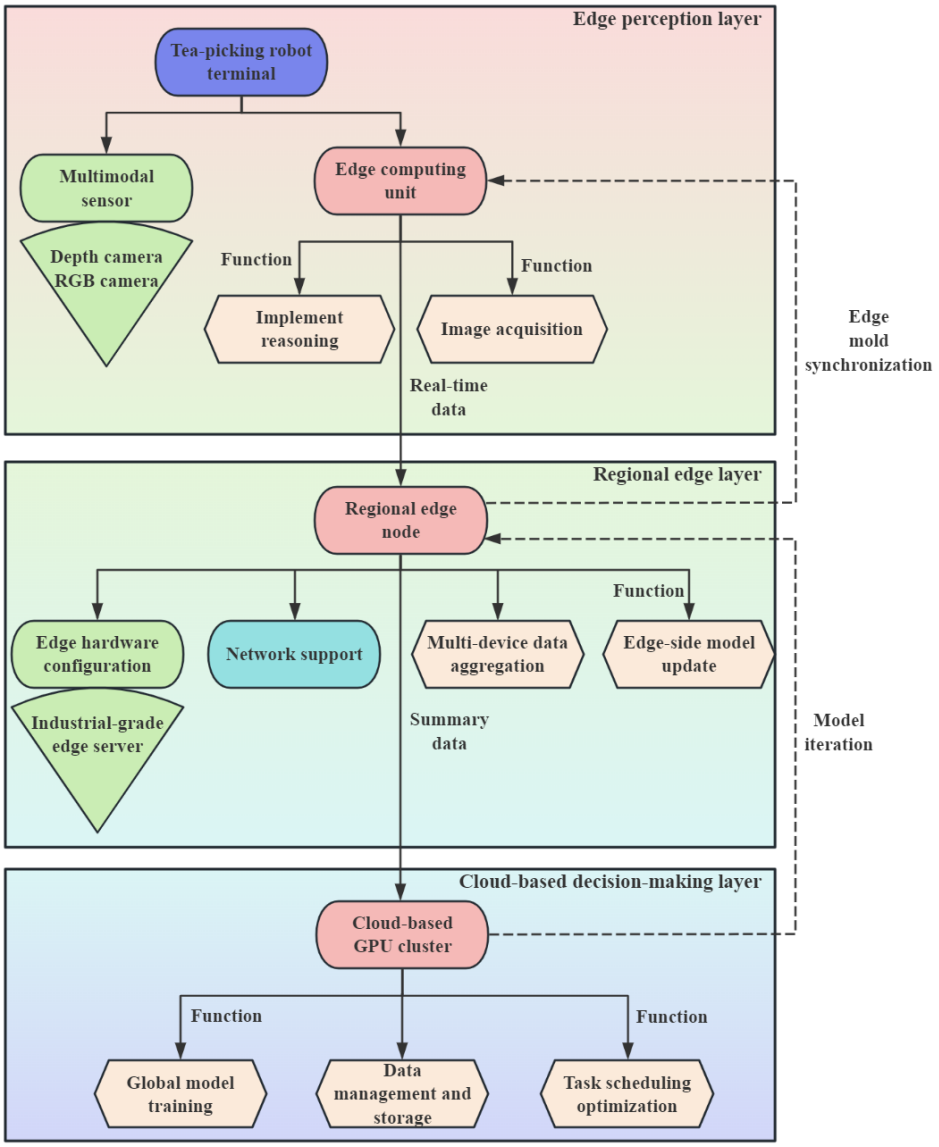

Figure 17. Three-level architecture system.

In view of the characteristics of the tea garden network environment, a redundant network architecture is designed:

Transmission protocol: Data transmission is carried out using the MQTT protocol, and QoS1 quality of service ensures reliable message delivery.

Bandwidth adaptive: Automatically adjust image resolution (640×640→1280×1280) based on network conditions;

Edge cache: Local storage of 72 hours of detection data, automatically synchronized to the cloud after network recovery.

5.2.2 Edge Device Deployment process

Equipment selection and protection.

Computing unit: Jetson AGX Xavier industrial-grade module (IP67 protection grade);

Cooling system: Integrated passive heat sink and low-noise fan, operating temperature range -20 ℃ to 60℃.

Power management: Supports wide voltage input (9-36V), with an internal lithium battery backup solution (2 hours of battery life).

(1) Mechanical installation requirements

- The guide rail installation ensures the stable connection of the equipment under the vibration of the mechanical arm.
- Deployment location: Within 30cm of the camera to avoid signal attenuation.
- Cable protection: Armored network cables and waterproof connectors are adopted, with a protection level of IP67.

(2) Basic environment setup

- Operating system: JetPack 5.1.1 (based on Ubuntu 20.04).
- Development toolchain: CUDA 11.4, cuDNN 8.6, TensorRT 8.5.2.
- Containerized deployment: Docker 20.10.17 + NVIDIA Container Toolkit.

(3) System service configuration

- Inference service: Configured as a system service to achieve automatic startup at boot and automatic recovery after a crash.
- Log Management: Build a log monitoring system using rsyslog + Elasticsearch + Kibana.
- Security hardening: Disable unnecessary system services and configure firewall rules to restrict access.

The deployment architecture and process designed in this section establish a complete technical path for the practical application of the tea detection system in complex tea garden scenarios. By adopting a three-level architecture (edge perception layer, regional aggregation layer, and cloud decision-making layer), the system achieves efficient collaboration between terminal data acquisition, edge real-time processing, and cloud global optimization, effectively addressing the challenges of distributed deployment in large-scale tea gardens.

The use of low-precision inference (FP16/INT8) based on TensorRT not only reduces the memory footprint of the model by over 50% but also accelerates the inference speed by 60% compared to FP32, making it feasible to deploy on resource-constrained embedded devices such as Jetson AGX Xavier. Meanwhile, the redundant network design (5G/CBRS + MQTT protocol)

and edge caching mechanism ensure reliable data transmission and continuity of detection tasks even in environments with unstable network signals.

The mechanical installation specifications (IP67 protection, shock-resistant design) and software configuration strategies (containerization, system service management) further enhance the system's adaptability to harsh tea garden environments (high temperature, humidity, and vibration), ensuring a mean time between failures (MTBF) of over 8 hours, which meets the requirements of all-day continuous operation.

In summary, this deployment scheme realizes the transition from algorithm optimization to engineering application, providing a robust and scalable technical support for the industrialization of intelligent tea-picking technology.

6.Conclusion

This study delves deeply into the optimization method of tea target detection based on YOLOv8. The improvement points are as follows:

(1) The MSA multi-head self-attention mechanism is added to the Bottleneck module, and a residual connection branch is added to the first CBS module to connect to the Add process, forming a new module BottleneckM. Further, the BottleneckM module is replaced by the C2f module to fuse into the new module C2fM.

(2) The SPPF module was improved. By combining asymmetric convolution to generate a new ACBSPPF module, the number of model parameters was significantly reduced, the detection speed and accuracy were enhanced, and the positioning accuracy and recognition ability of the model for tea were improved.

In conclusion, this study provides an effective optimization method for tea detection based on YOLOv8, and combines it with PyQt5 to design a tea detection system, making positive contributions to promoting the intelligence of tea picking.

References

[1] Li Linshan.Research Progress on Tea Picking Techniques and Tea Picking Machinery[J]. *Southern Agricultural Machinery*,2024,55(13):61-64.

[2] He Yu.Research on Tea Bud Recognition Based on Deep Learning[D].Xihua University, 2023.

[3] Xu Zheng.Manual Picking Techniques for Fresh Tea Leaves[J].*Science Planting and Breeding*,2018(03):21.

[4] Zheng Hang,Fu Tong,Xue Xianglei, et al.Research Status and Prospect of Mechanized Tea Picking Technology[J].*Chinese Journal of Agricultural Mechanization*,2023,44(09):28-35.

[5] Dong Chunwang. Innovative Thoughts on Intelligent Processing Technology of Tea[J].*China Tea*,2019,41(03):53-55.
[6] Chen X,Gupta A.An implementation of faster rcnn with study for region sampling[J].arxiv preprint arxiv:1702.02138,2017.
[7] Liu W,Anguelov D,Erhan D,et al.SSD:Single shot multibox detector[C]//European Conference on Computer Vision,2016:21-37.
[8] Redmon J,Divvala S,Girshick R,et al.You only look once:Unified,real-time object detection[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016:779-788.
[9] Reis D,Kupec J,Hong J,et al.Real-time flying object detection with YOLOv8[J].arxiv preprint arxiv:2305.09972,2023.
[10] Simonyan K,Zisserman A.Very deep convolutional networks for large-scale image recognition[J].arXiv preprint arXiv:1409.1556,2014.
[11] He K,Zhang X,Ren S,et al.Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition.2016:770-778.
[12] Xu Yanwei,Li Jun,Dong Yuanfang,et al.Review of YOLO Series Object Detection Algorithms [J].*Journal of Frontiers of Computer Science & Technology*,24,18(9).
[13] WOO S,PARK J,LEE J Y,et al.CBAM:Convolutional block attention module[C]//Computer Vision-ECCV 2018.Cham:Springer International Publishing,2018:3-19.
[14] Xu Jun. Research on Computer Vision Image Description Based on Deep Learning [J]. *Information and Computer (Theoretical Edition)*,2023,35(19):155-157.
[15] WANG Q L,WU B G,ZHU P F,et al.ECA-Net:Efficient Channel Attention for Deep Convolutional Neural Networks[C]//2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR).Seattle:IEEE,2020:11531-11539.
[16] YangL,WangS,Chen X,et al.High-fidelity permeabilityandporositypredictionusingdeeplearning withtheself-attentionmechanism[J].*IEEETransactions on Neural Networksand Learning Systems*,2022,34(7):3429-3443.
[17] Zhang H,Goodfellow I,Metaxas D,et al.Self-attention genera-tive adversarial networks[C]//International conference on ma-chine learning.PMLR,2019:7354-7363.
[18] Song Yi, Ding Geyuan. Research on Foreign Object Detection Algorithm for Transmission Lines Based on Improved YOLOv8[J].*Industrial Control Computer*,2020,38(06):49-51.
[19] Zhao Xiaoxiao Vehicle License Plate Detection Based on YOLOv10n+SE Attention Mechanism [J].*Information and Computer,*2020,37(10):6-8.
[20] MAO Ziwei, Zhou Zhengkang, Tang Jiashan. Research on Road Vehicle Fire Detection Algorithm Based on Improved YOLOv8 [J]. *Radio Engineering*,2020,55(05):920-927.
[21] Zhu Yuhua,Zhang Yuhuan,Li Zhihui,et al.Research on Grain Storage Temperature Prediction Based on TCN-BiGRU Combined with Self-Attention Mechanism[J].*Chinese Journal of Agricultural Mechanization*,24,45(12):133-139.
[22] Yang Bingzhen,Lin Yuanhao, Ji Lihua,et al.Bearing Fault Diagnosis Method Combining Multi-head Attention Mechanism with Generative Adversarial Neural Network[J].*Mechanical and Electrical Engineering Technology*,2020,54(11):158-163.
[23] Gao Min,Chen Gaohua,Gu Jiaxin,et al.FLM-YOLOv8:A Lightweight Mask-Wearing Detection Algorithm [J].*Computer Engineering and Applications*,2024,60(17):203-215.
[24] Ouyang Jianquan, Tang Huanrong, Lu Jiaxiong. Research on Target Detection of Pig Counting Based on YOLOv8[J].*Journal of Xiangtan University(Natural Science Edition)*,2025,1-13.
[25] Wang Yongsen, Liu Qian, Liu Libo.ACGFN:Speech Recognition Model Based on Asymmetric Convolution and Gated Feedforward Neural Network[J].*Journal of Chinese Information Processing*,2020,39(01):167-174.

[26] Kabir M M,Hafiz M S,Bandyopadhyaa S,et al.Tea leaf age quality:age-stratified tea leaf quality classification dataset[J].*Data in Brief*,2024,54:110462.
[27] Zhang Wenguang,Zeng Xiangjiu,Liu Chongyang.Research on Graphic Element Recognition Method of Electric Power Dispatching Control System Based on Improved YOLOv7[J].*Electrical Technology*,2025,26(5):1-9.
[28] WU Binbin, ZHANG Lihua, LIU Junwei, DONG Junjun. Improved EP-RTDETR based surface defect detection on PCB[J].*Manufacturing Technology & Machine Tool*,2025,(3):139-148.
[29] CHEN Wei，JIANG Zhicheng，TIAN Zijian，et al. Unsafe action detection algorithm of underground personnel in coal mine based on YOLOv8[J].*Coal Science and Technology*,2024,52(S2):267-283.
[30] Qi Xiangyu, Su Qinghua, Zhang Zhichao, et al. Defect Detection Algorithm for Improving YOLOv11[J].*Computer Science and Applications*,2025,15:108.
[31] Guo Jing,Hu Meng,Li Weishan,et al.Research on Cinema Ticketing System Detection Platform Based on PyQt5 and SpringBoot[J].*Modern Information Science and Technology*, 2020,9(01):88-92+99.
[32] Liu Runji. Research on Finger Vein Recognition Algorithm Based on Lightweight Network[D].*North China University of Technology*,2024.