Operational Research in Engineering Sciences: Theory and Applications
Vol. 7, Issue 2, 2024, pp. 315-336
ISSN: 2620-1607
eISSN: 2620-1747
DOI: https://doi.org/10.31181/oresta/070215

AUTOMATED IDENTIFICATION OF TEA LEAF DISEASES AND PESTS USING DEEP LEARNING METHODS

Xianghong Deng[1], Tao Chen[2], Chonlatee Photong[1*]

[1]Faculty of Engineering, Mahasarakham University, Maha Sarakham, 44150, Thailand.
[2]Hunan Mechanical & Electrical Polytechnic, Changsha 410119, China.

Received: 04 February 2024
Accepted: 20 June 2024
First Online: 30 June 2024

Research Paper

Abstract: *Tea is a significant crop and deeply loved by individuals. In earlier times, the identification of tea leaf diseases and pests was manual and inefficient. With the increasing application of AI (artificial intelligence), deep learning and image recognition technology in the field of agriculture, this paper introduces a method with improved efficiency and precision for intelligent identification in tea leaf diseases and pests. We applied the deep learning target detection model, which is the recent version of YOLO (You Only Look Once), specifically YOLOv10s, for automated recognition of tea leaf diseases and pests. This research primarily involves three models: YOLOv8s, YOLOv9s, and YOLOv10s. After training and validation, we conducted a comprehensive performance evaluation and comparative analysis of these models. The comparison of performance metrics indicated that the model based on YOLOv10s performed the best. As shown by the test evaluation results, precision, recall, mAP50 (mean of Average Precision), F1-Score, these values are all higher than those achieved by YOLOv8s and YOLOv9s. Using the optimal YOLOv10s model, combined with the PyQt5 library, a tea leaf diseases and pests target detection recognition interface was developed. Based on this proposed model with YOLOv10s, the identification of tea leaf diseases and pests will be significantly improved for all the terms of higher efficiency, less costs, as well as enhanced quality and sustainability of tea production.*

Keywords: *Tea leaf diseases and pests, Deep learning, Image recognition technology, YOLOv10s, PyQt5*

1. Introduction

Tea is both a beverage and an important link between history, culture, economy and health. Tea diseases and pests can have various impacts on tea production and

*Corresponding Author: chonlatee.p@msu.ac.th, (C. Photong), 65010363007@msu.ac.th, (X. Deng), 583834529@qq.com, (T. Chen)

quality, including reduced yields, compromised quality, and economic losses. Effective disease control measures are crucial to ensuring tea yield and quality. Over the past several years, due to the swift advancement of intelligent and precise agricultural practices, the utilization of computer-based image recognition technology has progressively expanded, addressing numerous challenging issues within agricultural science (Chen et al., 2019). As deep learning has rapidly evolved and computational capabilities have seen substantial enhancements, object detection algorithms have shifted from relying on conventional, manually crafted features to adopting deep learning-driven detection methods. These methods leverage Convolutional Neural Networks (CNN) for feature extraction within context of deep learning-based target detection (Liu, 2024). Deep learning-based models have outperformed previous artificial intelligence systems in a number of domains, including speech recognition, language processing, and computer vision(C.-Y. Wang et al., 2024).

Deep learning technology offers robust instruments and approaches for artificial intelligence, allowing it to attain notable accomplishments in object detection endeavors. The integration of feature extraction and classifier training enables deep learning models to autonomously acquire task-relevant features without the need for manual formulation, thereby aiding in the identification of superior feature representations. Object detection in images is an important task in deep learning, focusing on identifying multiple objects and their locations within an image, and is applicable for detecting multiple objects.

In order to increase the quality of tea produced, this study presents a more effective and precise automated technique for identifying pests and diseases related to tea utilizing the most recent version of YOLOv10. Yolov10 adopts advanced model design methodologies and optimization strategies, enabling it to identify tea diseases with greater accuracy. Its comprehensive model design approach optimizes various components to enhance both efficiency and precision, which makes it particularly well-suited for environments with constrained resources. This study's primary contributions are:

(1) A fresh approach to the tea leaf disease and pest identification was presented that can cut down on labor intensity, time spent, and errors caused by visual inspection techniques.

(2) The use of YOLOv10s, the most recent version, in identification of diseases and pests in tea plants was investigated. It illustrates how cutting-edge deep learning methods can be used in agricultural settings. Assessing the capability of the suggested model with metrics including Precision, Recall, mAP50 and F1-Score. The indicators provide vital knowledge of the model's function in actual use; especially, in accurately identifying pests and diseases that affect tea.

(3) Based on the best-performing YOLOv10s model obtained after training, a software system with a graphical interface was developed using Python and PyQt5. It effectively supports detection of data such as images, videos, and cameras, and also supports saving the detection results. It is capable of real-time and off-site detection of tea diseases.

2. Literature Review

In this work, three distinct forms of tea illnesses were detected automatically using Support Vector Machine (SVM) and a comparatively modest number of variables (Hossain et al., 2018). There is no dedicated detection system. The apply of convolutional neural networks for the detection of tea diseases was suggested by researchers in the study. The application of serial CNNs; specifically, Inception-ResNet-v2, Xception, and GoogleNet, was the researchers' main focus (Krisnandi et al., 2019). In this work, over a thousand pictures of tea leaves function as the foundation for training the model, which is built on a faster region-based convolutional neural network (Faster R-CNN)(Lee et al., 2018). The data augmentation in this research was accomplished by using a limited number of initial training samples made up of pictures of sick tea leaves, and then creating fresh training samples using an unconditional generative model called SinGAN (Hu & Fang, 2022). The paper presents an augmented target detection and recognition model, called AX-RetinaNet, which is a better version of RetinaNet, with the goal of automatically detecting and recognizing tea diseases photos (Hu & Fang, 2022). The paper utilizes a deep learning platform called "Region-based Faster Convolutional Neural Network" to detect Tea Leaf Blight (TLB) leaves (Hu et al., 2021). Adaptive Spatial Feature Fusion (ASFF) technology and the BiFPN feature fusion network have enhanced the multi-scale feature fusion of tea illnesses, improving the model's resistance to interference from complicated backdrops (Lin et al., 2023). They discovered that the YOLOv8 model, significantly improves classification results through its C2f module (Xiao et al., 2023). The YOLO (You Only Look Once) system distinguishes itself through its exceptional equilibrium of speed and precision, facilitating rapid and dependable object recognition in photographs (Terven et al., 2023). This research utilizes the SIoU-optimized loss function to improve the model's capacity for learning from pest samples, building upon the foundation of the original YOLOv8 network architecture(Z. Wang et al., 2024). Xue et al. (2023) incorporated self-attention and convolution (ACmix), and the convolutional block attention module (CBAM), into YOLOv5. Wang et al. (2022) proposed TIA YOLOv5, which adds a transformer encoder block to the backbone network to enhance the model's sensitivity to weeds. The YOLOv5 model is enhanced and used to detect thick and microscopic tea shoots (Lawal et al., 2023).

In commercial orchards, this study conducted a comprehensive assess of the performance of all arrangements of the YOLOv8, YOLOv9, and YOLOv10 object recognition algorithms for detecting green fruit (fruitlets)(Sapkota et al., 2024). Utilizing YOLOv8s as a foundation, this research introduces an enhanced model termed YOLOv8s-CFB. The backbone network of this model integrates partial convolution (PConv), upgrades the C2f module, and incorporates a novel CSPPC structure, designed to decrease processing overhead and boost performance speed (Zhao et al., 2024). A method for multi-level feature fusion is incorporated, paired with the Ghost model to ensure adaptability while minimizing network size. Additionally, a SERes detection head is incorporated to augment the network's capacity to identify similar objects (Zhu et al., 2024). The paper presented the better lightweight target detection model GAS-YOLOv8 (Wang & Wang, 2024). The paper demonstrates that employing the AW-YOLOv8 model enhances early warning effectiveness by identifying species of insect pests during the cotton boll-forming stage (Chen, 2024). The study introduced an enhanced model Yolo-GSG based on Yolox for detecting tea buds (Gui et al., 2024). Based on YOLOv8, this work suggests YOLOv8-HD, a new wheat seed detection network (Ma et al., 2024). The purpose of the paper is to present a novel model, built upon a modified YOLOv8 framework, for enhancing the effectiveness of recognizing tea diseases and pests (Ye et al., 2024). YOLOv9 distinguishes itself in object identification through its strategic architectural modifications that tackle the critical issue of information loss in deep neural networks, which is essential for preserving detection reliability and processing speed (Sapkota et al., 2024). Two of the key advancements in YOLOv9 are the Generalized Efficient Layer Aggregation Network (GELAN) and the Programmable Gradient Information (PGI) structure. Compared to YOLOv8, the deep model's superior design enables it to cut both the number of specifications and computations by 49% and 43% (C.-Y. Wang et al., 2024). A more precise and efficient way to automate the monitoring process is presented in this study. Utilizing YOLOv9, the most recent version, makes it possible to categorize tomato maturity stages and makes tomato counting easier (Vo et al., 2024). Comprehensive tests show that, in comparison to other cutting-edge detectors, the YOLOv10 achieves cutting-edge performance while balancing efficiency (A. Wang et al., 2024).

Deep learning models have achieved a certain improvement in accuracy of pests and diseases classification and recognition in tea leaf. However, there remains potential for additional enhancement in both model parameters and overall performance. This paper proposes an improved performance and accuracy method for automated monitoring of tea pests and diseases. By utilizing the recent version of YOLO, particularly YOLOv10, it enables the detection of tea diseases and accurately locates the positions of diseases and pests. The adoption of an NMS-free training methodology, coupled with the incorporation of large-kernel convolutions and partial self-attention modules in YOLOv10, signifies a major advancement in model architecture. The novel features not only elevate detection accuracy but also mitigate computational demands, thereby rendering YOLOv10 highly appropriate suitable for resource-constrained environments.

3. Materials and Methodology

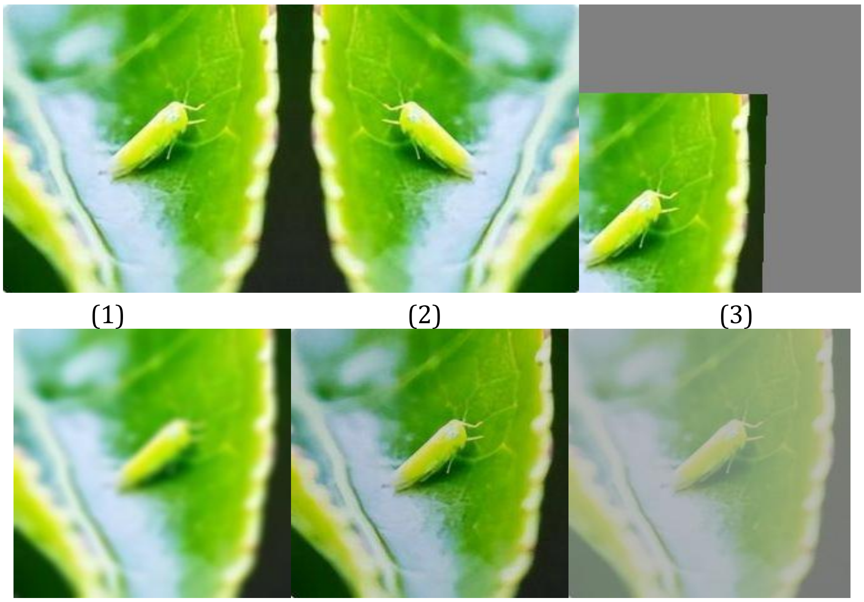
3.1 Approach to Data Acquisition and Preparation

The insect pests and tea diseases that were the subject of this study's photos almost were captured in the open air. We mainly collected three common tea diseases and pests in tea gardens namely brown_blight, gray_blight and tea leafhopper.Tea brown_blight and gray_blight disease can feature brown and gray lesions that affect the color and appearance of tea leaves. In severe cases, they can cause leaf abscission, leading to reduced tea production. The tea leafhopper insect is one of the primary pests in tea gardens. It feeds on tea leaf sap, leading to leaf yellowing and reduced yields. We were able to photograph tea leaves with few targets as well as those with a lot of targets due to the fact that our images were photographed in a natural setting. The photographs, which show a lot of foliage, include shadows and reflections on the leaves as well as foliage obscuration and overlap. We have collected three common tea diseases and pests in tea gardens namely brown_blight, gray_blight and tea leafhopper approaching more than 1000 images. A few typical samples from our data archives are displayed in Figure 1.

(1) (2) (3)

Figure 1: The dataset's representative samples. (1) brown_blight (2) gray_blight (3) tea leafhopper

Techniques for augmenting image data primarily enhance the diversity of training data through the application of diverse transformations to pictures. Common image data augmentation techniques include: rotation, flipping, and cropping, which alter the orientation and position of images. Scaling and deformation change the size and shape of images. Color transformation alters the brightness, contrast, saturation, and other aspects of images. To boost the model's generalization capability, we adopt the data enhancement technique, which directly uses the transforms module of the torchvision library of PyTorch, a deep learning framework, to realize random cropping, rotation, Gaussian blurring, brightness and contrast change of the image, etc., PyTorch provides extensive data augmentation tools, and we can directly use the transforms module in torchvision to achieve this, which increases the diversity of the training data. As shown in Figure 2.



(1) (2) (3)

(4) (5) (6)

Figure 2: The representative samples of image data enhancement (1) original (2) flipping (3)random affine (4) gaussian blur (5) brightness (6) contrast

To annotate the images of diseases and pests in tea leaves (brown_blight ,gray_blight and tea leafhopper), Manual labeling of rectangular regions was performed with the LabelImg annotation tool. We employed the LabelImg tool to annotate the tea diseases and pests target area in the images (Liu, 2024).

As shown in Figure 3, the label format was a .xml file to accommodate the YOLO algorithm. Annotation boxes were added, and corresponding label files were generated for the diseased areas in the images. Label information encompassed the image filename, image size, and specific details, including the labels and the coordinates of the top-left and bottom-right pixels. We have labeled the three common diseases with brown, gray, and leafhopper as shorthand names. Our authors conducted a manual review of the annotation results, conducting detailed checks on the accuracy, completeness, and consistency of each annotation to ensure the quality of the annotations.

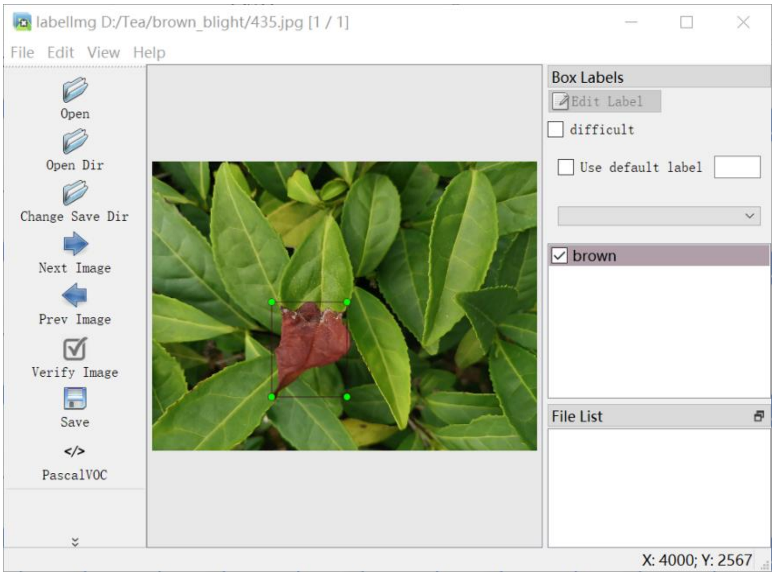


Figure 3: Image labeling fabrication

3.2 Overall Methodology

Research in computer vision has centered on real-time target detection, with the goal of precisely predicting the class and position of objects in a photo while minimizing latency. Among them, YOLO is becoming increasingly popular due to its balance between performance and efficiency. The following elements are typically included in the training framework for YOLO models, which are used to recognize diseases and pests in the tea leaf: data collection, annotation, preprocessing, model selection, training, evaluation, and optimization. The main concept of YOLO is to simultaneously forecast multiple bounding boxes and class probabilities within each grid cell after dividing the entire photo into a predetermined number of grid cells.

After data processing, the discrepancy among the models' predicted labels and the actual labels is calculated. In Figure 4, in order to reduce the loss function and strengthen model behavior, the backpropagation technique is used to update the model's performance. A number of parameters must be changed during model training, including the loss function, which determines the ideal weight parameters for the neural network by using a particular measure. Optimizer: establishes how the model updates itself using its own loss function and the data it has seen, allowing for effective parameter space exploration. To guarantee uniformity and comparability, every model configuration was trained using the same computational system and identical hyperparameter values. End-to-end object detection is accomplished by the YOLO method, which makes use of a different CNN model. The chief concept is to apply the complete photo as the network's input and to regress the bounding box's position and classification in the output layer directly.

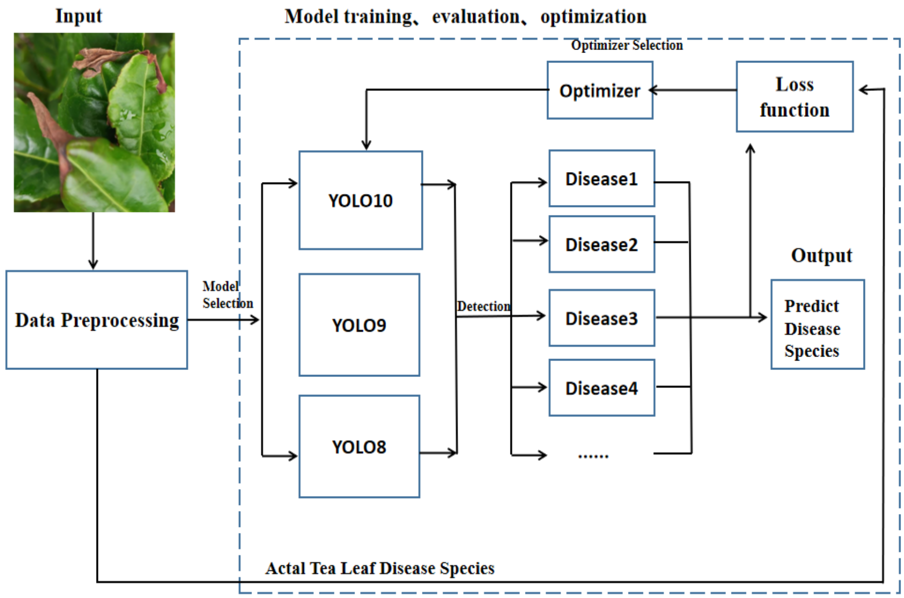


Figure 4: The training framework of YOLO models

Based on the compiled dataset, we train the object detection model using the cutting-edge YOLOv10s, YOLOv9s, and YOLOv8s object identification techniques to achieve real-time identification of the target objects.

With additional advances, YOLOv9s expands upon the YOLOv8s network and focuses mainly on resolving issues related to information loss in deep learning frameworks. The creative application of the data constraint concept and invertible functions, which guarantees YOLOv9's great accuracy and efficiency, is the basis of its design. In YOLOv9, a unique idea known as Programmable Gradient Information (PGI) was presented to solve the issue of data bottlenecks and guarantee the preservation of critical data throughout deep network levels. This improves detection performance overall by producing dependable gradients that enable precise model changes. The General Efficient Layer Aggregation Network (GELAN) is an important configuration breakthrough that makes it possible for YOLOv9 to attain higher computing efficacy and parameter efficiency. Because of its flexible integration of different computing blocks, Thanks to its architecture, YOLOv9 can accommodate a diverse array of applications without sacrificing accuracy or performance.

By removing Non-Maximum Suppression (NMS) and optimizing different model components, YOLOv10's design improves on the advantages of earlier YOLO models and achieves state-of-the-art performance while drastically lowering computational cost.

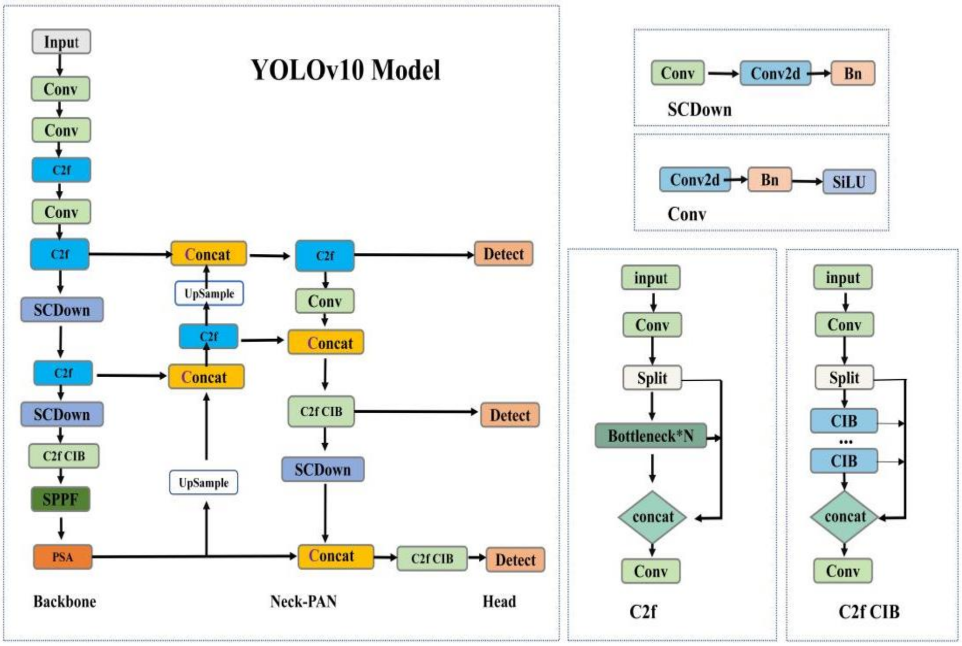


Figure 5: The Yolov10 model network structure

In Figure 5, the components of the Yolov10 model network structure are as follows: Backbone: In order to optimize gradient flow and lessen unnecessary calculation, the YOLOv10 backbone applies an improved version of CSPNet (Cross Stage Partial Network) for feature extraction. SCDown (Spatial-channel decoupled downsampling) decouples space and channels. The number of channels is first adjusted by 1×1 point-by-point convolution, and then spatial downsampling is done by 3×3 depth convolution, which minimizes the computational cost while maximizing the information retention.

Neck: The purpose of the Neck is to combine characteristics from several scales and transfer them to the head. For efficient multi-scale feature fusion, PAN (Path Aggregation Network) layers are included. PSA, or Point-wise Spatial Attention, is an efficient local self-attention mechanism that processes a portion of the convolved features through a multi-head self-attention network (MHSA), and a feed-forward network (FFN), and then connects the two sections and fuses them by convolution. Enhanced global modeling capability and reduced computational complexity.

During training, the Yolov10 model employs two prediction heads: one that uses one-to-many allocation and the other that uses one-to-one allocation. Efficient inference without NMS is made possible by the model's ability to use the enriched supervised signals from the one-to-many assignments during training and the predictions from the one-to-one assignments during inference. One-to-many Head:

During the training process, it has multiple predictions for each object, offering abundant supervisory signals and enhancing learning precision. One-to-One Head: During inference, it initiates the excellent forecasting for each object, discarding the need for NMS, thus decreasing processing delay and increasing performance.

YOLOv10 has further advanced by incorporating NMS-free training and inference techniques, leading to a substantial decrease in latency and a notable enhancement in real-time capabilities. By integrating spatial-channel separated down-sampling and a rank-assisted block configuration, YOLOv10 has strengthened the feature extraction procedure, leading to improved accuracy accompanied by a reduction in both parameters and computational demands. The evaluation metrics unequivocally demonstrate that YOLOv10 surpasses its earlier versions, especially in the context of smaller models, rendering it exceptionally well-suited for edge deployments where both precision and efficiency are paramount.

3.3 Experiment Setup

Based on a Windows operating system, this experiment loads the model using the PyTorch learning framework, writes the program in Python, and saves the model parameters at the optimal time for training. Table 1 displays the experimental platform's setup table. PyTorch uses Dynamic Computational Graph, which builds the computational graph at each forward propagation, which makes debugging and development more flexible and intuitive. PyTorch serves as a robust and adaptable deep learning framework, catering to a variety of scenarios ranging from research endeavors to practical, real-world applications.

Table 1: Configuration table of experiment platform

Experimental Environment	Details
Mirroring	PyTorch 1.9.0 Python 3.8(ubuntu18.04) Cuda 11.1
GPU	RTX 3080 Ti(12GB) * 1
CPU	12 vCPU Intel(R) Xeon(R) Silver 4214R CPU @ 2.40GHz
RAM	90GB

The hyperparameters of automated diseases and pests detection in tea leaves with YOLO(v10,v9.v8) object detection models are shown in the Table 2. The setting of the batch size at 8 ensured an optimal equilibrium between memory utilization and processing speed. To standardize the input data across all models, photos were resized to a consistent resolution of 640x640 pixels. For efficiently updating model weights, particularly suitable for managing large-scale and intricate data, the Stochastic Gradient Descent (SGD) optimizer was utilized. A momentum value of 0.937 was sustained to guarantee coherent updates throughout epochs, accompanied by a negligible weight decay of 0.0005 to mitigate overfitting risks.

Table 2: Training hyperparameters

Specifications	Merit
Batch	8
Epochs	150
Image	640 × 640
Momentum	0.937
Weight decay	0.0005
Optimizer	SGD(Stochastic Gradient Descent)

3.4 Evaluation metrics

In this study, 150 epochs are used to train three models (YOLOv8s, YOLOv9s, and YOLOv10s) for comparative analysis using the newly introduced dataset. To determine which of these three models is best suited for this dataset, we primarily examine the training outcomes using performance metrics covering Precision, Recall, mAP50 and P-R curve and additional performance indicators, with the goal of choosing the best model that works well with this dataset.

The precision (P) and recall (R) coordinates illustrate precision and recall rate, respectively. The P-R curve is a curve bounded by the corresponding precision and recall coordinates. The average PR curves across all categories are shown by the thick blue lines, while the lines of various hues depict the PR curves of the other categories. Determine the average AP value for each category to find the mAP (mean of average precision). The YOLO model displays mAP@0.5. The value of mAP when the IOU (Intersection over Union) criterion is ranged at 0.5 is represented by this expression. The map under multiple IOU thresholds, or mAP @ [0.5:0.95], is another expression of the YOLO concept.

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \times 100\% \tag{1}$$

$$\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \times 100\% \tag{2}$$

$$\mathrm{APi} = \int_0^1 \mathrm{P(R)}\,\mathrm{dR} \tag{3}$$

$$\mathrm{mAP} = \frac{1}{\mathrm{N}} \sum_{\mathrm{i}=1}^{\mathrm{N}} \mathrm{APi} \times 100\% \tag{4}$$

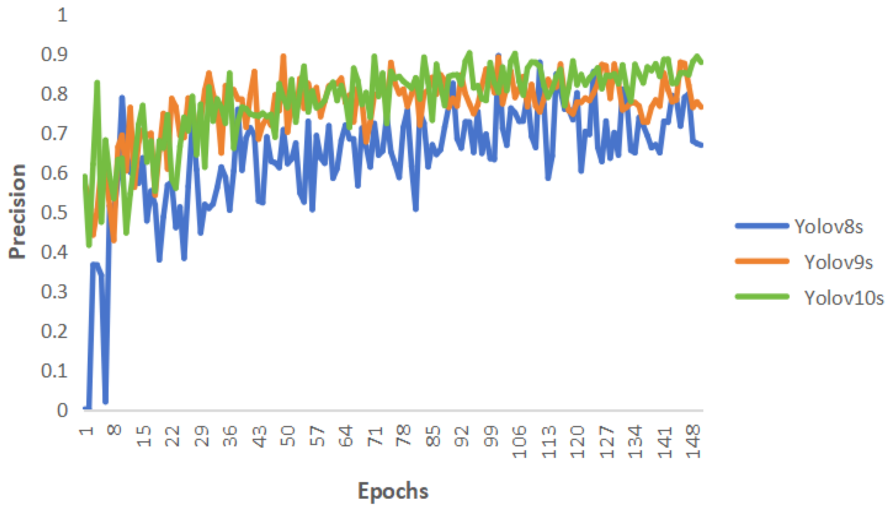
$$\mathrm{F1-Score} = \frac{2 \times \mathrm{Recall} \times \mathrm{Precision}}{\mathrm{Recall} + \mathrm{Precision}} \times 100\% \tag{5}$$

True Positive (TP): When the predicted value and the label value are both positive, indicating a correct prediction. False Negative (FN): When the predicted value is negative but the label value is positive, signifying a faulty forecast. False Positive (FP): When the prediction is positive but the label value is negative, also indicating an incorrect prediction. True Negative (TN): When both the predicted value and the label value are negative, signifying a correct prediction(Zhao et al., 2023).

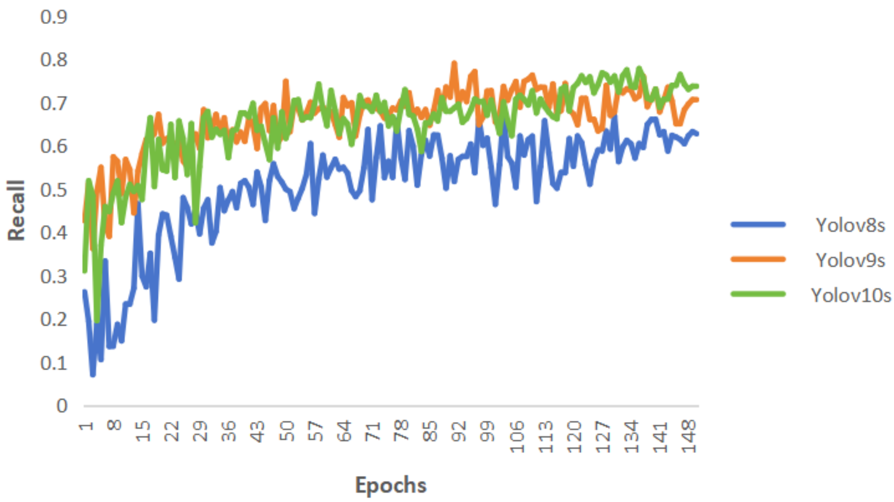
4. Outcomes and Discussion

4.1 Analysis of Different Model Outcomes

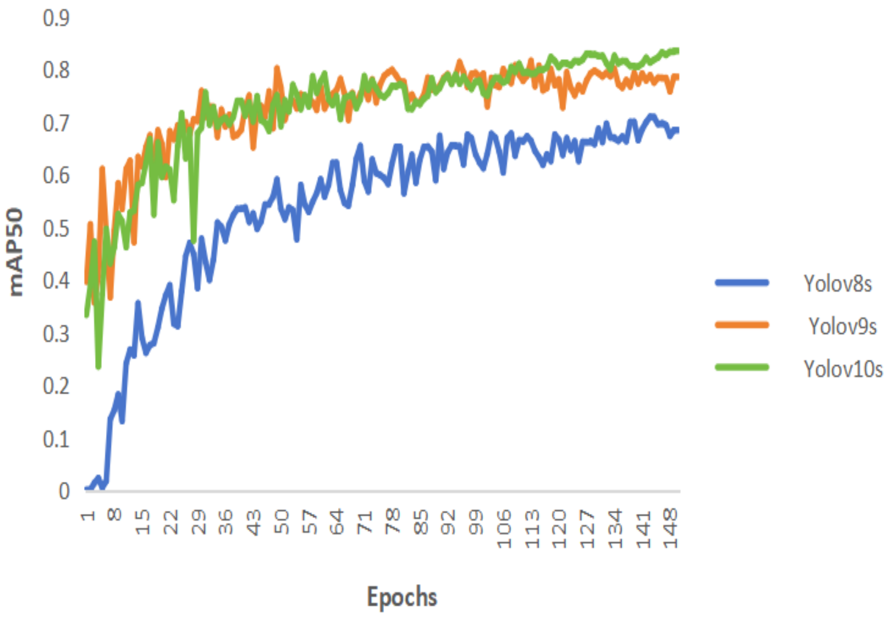
The target of the paper is to find a dependable and precise model for identifying pest and disease images in tea leaves. Three well-known object identification frameworks: YOLOv8s, YOLOv9s, and YOLOv10s, were employed in this investigation. We split the dataset into a training set, a validation set, and a test set in a 7:2:1 proportion. Due to the medium number of datasets, in order to balance the speed and accuracy, it is not easy to overfitting, YOLOv8s, YOLOv9s, and YOLOv10s are selected for this training. In this paper, we mainly conduct model training based on three models, namely YOLOv8s, YOLOv9s, and YOLOv10s, after numerous repeated experiments and hyperparameter adjustments, the optimal model was finally obtaine, and after the training is completed, we conduct a comprehensive performance evaluation and evaluation of performance relative to others with the three models on the validation set. The model training and evaluation process is basically the same, including dataset preparation, model training, and model evaluation. We analyze the training process of 150 rounds, at the beginning, the function of Yolov10s model and Yolov9s model is about the same, at the later stage the evaluation values of Yolov10s model are higher than that of Yolov9s model, and the mAP50 value of Yolov8s model has been lower than that of the other two models.



(a)



(b)



(c)

Figure 6: Performance comparison of each model. (a) precision curves (b) recall curves (c)mAP50 curves

Table 3 displays the comparison results of the three models on the test set. According to Table 3. Precision, Recall, mAP50 and F1-Score for Yolov10s were 87%,80 %,84%, 83% respectively, and these results are higher than both Yolov9s and Yolov8. Precision: Also known as the checking rate, this is the percentage of accurately predicted positive cases among all cases predicted as positive. Sometimes referred to as the check rate, recall is the percentage of correctly detected positive samples out of all the positive samples. Combining a binary classification model's precision and recall yields a metric called the F1-score, which is frequently used to evaluate the model's performance. Since mAP50 is a very important metric in target detection, it measures the average accuracy of the model at an IoU (intersection and concurrency ratio) threshold of 0.5. IoU quantifies the degree of overlap between the predicted bounding box and the actual bounding box. The F1-Score is a composite statistic that balances precision and recall, calculated using the harmonic mean.

Table 3: The results data based on different models

Model	Precision	Recall	mAP50	F1-Score
YOLOv8s	81%	69%	75%	75%
YOLOv9s	85%	71%	80%	77%
YOLOv10s	87%	80%	84%	83%

To better comprehend the assessment of the three models' performance, the results are displayed in the form of matrix diagrams in Figure 7. YOLOv10s achieves high mAP50 and F1 scores through its advanced model architecture, multi-scale feature fusion, and optimized loss function, which significantly strengthens the accuracy and decision-making capability of tea disease identification.

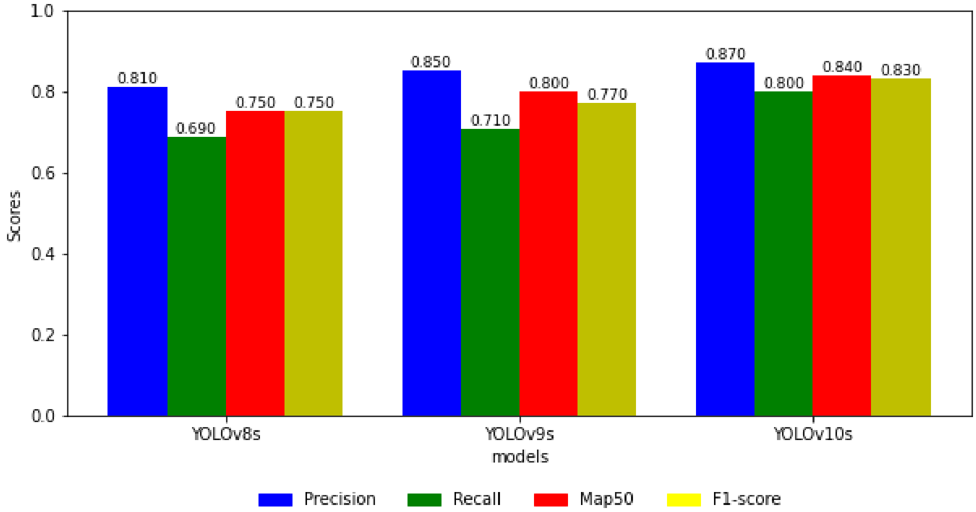


Figure 7: The best performance comparison of each model

The YOLOv10s shows superior performance in terms of preprocessing image speed, specifically, YOLOv10s has a higher postprocessing speed compared to the other two models, as shown in Table 4. YOLOv10s exhibits outstanding performance in post-processing speed, setting a new benchmark for fast image processing completion. This noteworthy benefit of the YOLOv10s during the last phases of image processing emphasizes how well-suited it is for quick output preparation, which is crucial for real-time applications (Sapkota et al., 2024). The faster post-processing speed enables YOLOv10 to quickly adapt and accurately detect targets in various scenarios. This aids the model in maintaining stable performance in complex environments and enhances its generalization capability. According to survey analysis, the efficient post-processing speed of YOLOv10s allows it to swiftly identify various objects in dynamic environments, providing real-time information for robots' path planning and decision-making.

Table 4: Processing speed of YOLOv8s, YOLOv9s and YOLOv10s

Model	Preprocessing Speed (ms)	Inference Speed (ms)	Postprocessing Speed (ms)
YOLOv8s	1.1	3.6	1.1
YOLOv9s	1.0	7.7	0.5
YOLOv10s	0.9	6.2	0.2

4.2 Analysis of Yolov10s Performance

Ten graphs depicting various metrics from the model's training and validation phases for the determination of diseases and pests in tea leaves are shown in Figure 8. Box loss, classification loss, distribution focal loss, precision, recall, and mean average precision (mAP) are some of the measurements. The graphs demonstrate an overall upward tendency over epochs. Model improvement is indicated by diminishing values for the loss measures (Box Loss, Classification Loss, Distribution Focal Loss). mAP50-95 is a robust metric that evaluates an object detection model's accuracy by considering both the precision and recall at multiple IoU thresholds, giving a thorough estimate of the model's detection capabilities across various scenarios. From Figure 8, we could see that the loss function curves are all gradually stabilizing. The precision, recall, and mean average precision (mAP) curves leveled off, indicating that there was no overfitting in the model training.

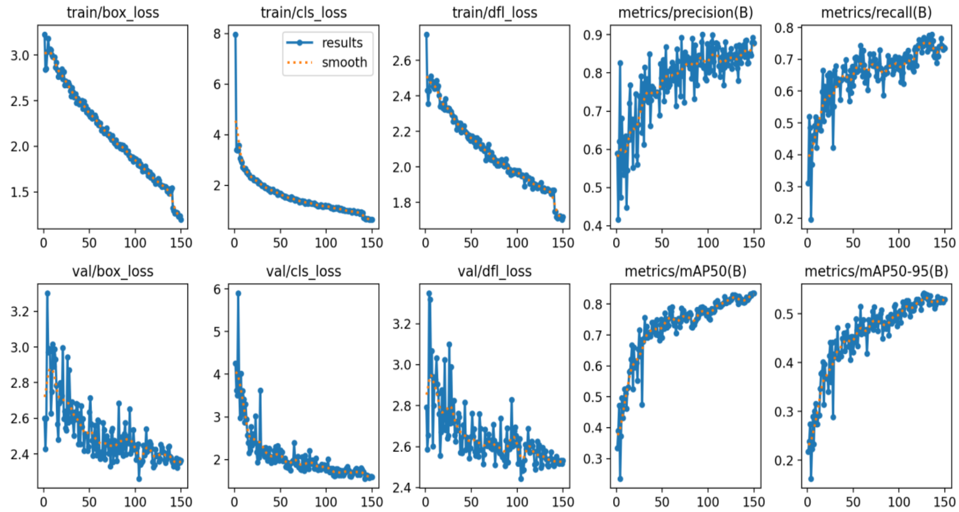


Figure 8: Performance of the YOLOv10s model

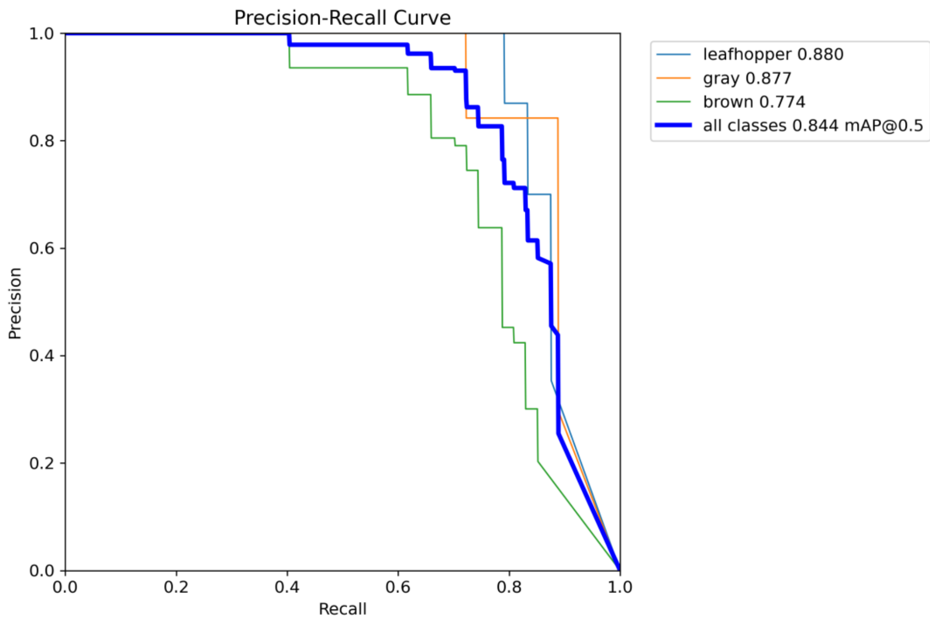


Figure 9: Precision-Recall curve

As shown in Figure 9, mAP represents the area conver by Precision and Recall when plotted as two axes, with 'm' indicating the average, and the number after '@' representing the threshold for determining positive and negative samples based on

IOU. We typically use the PR curve to illustrate the relationship between Precision and Recall. PR curves are generated by calculating the precision and recall of a model at different thresholds and plotting these points in a two-dimensional coordinate system. The horizontal axis usually symbolizes Recall and the vertical axis illustrates Precision. The PR curve for the training results in this paper is shown above. And mAP@0.5: This indicates the average mAP with a threshold greater than 0.5. It can be observed that the mAP@0.5 value for the object detection model in this paper is 0.844, which is quite good.

In the YOLOv10s training results, the normalized confusion matrix image is an important tool for evaluating model performance. A normalized confusion matrix scales each cell's value to a range between 0 and 1, usually by normalizing each row. This allows for a more intuitive display of the classification accuracy and error rates for each class. By identifying the main classes that are misclassified, you can target improvements in the dataset or model. For example, you might add more training data, adjust the model architecture, or fine-tune hyperparameters. Normalization helps fairly evaluate model performance in cases where class distribution is imbalanced, avoiding dominance by classes with more samples. This normalization allows us to visually compare classification accuracies between categories and identify the categories on which the model performs better or worse. As shown in Figure 10, The probabilities of correctly detecting brown_blight, gray_blight, and tea leafhopper are 0.77, 0.83, and 0.79.It has been found that there is cross-detection error between brown_blight and tea leafhopper. We need to ensure the diversity, representativeness, and unbiasedness of the data. Collecting a more diverse range of data can assist the model in generalizing more effectively.

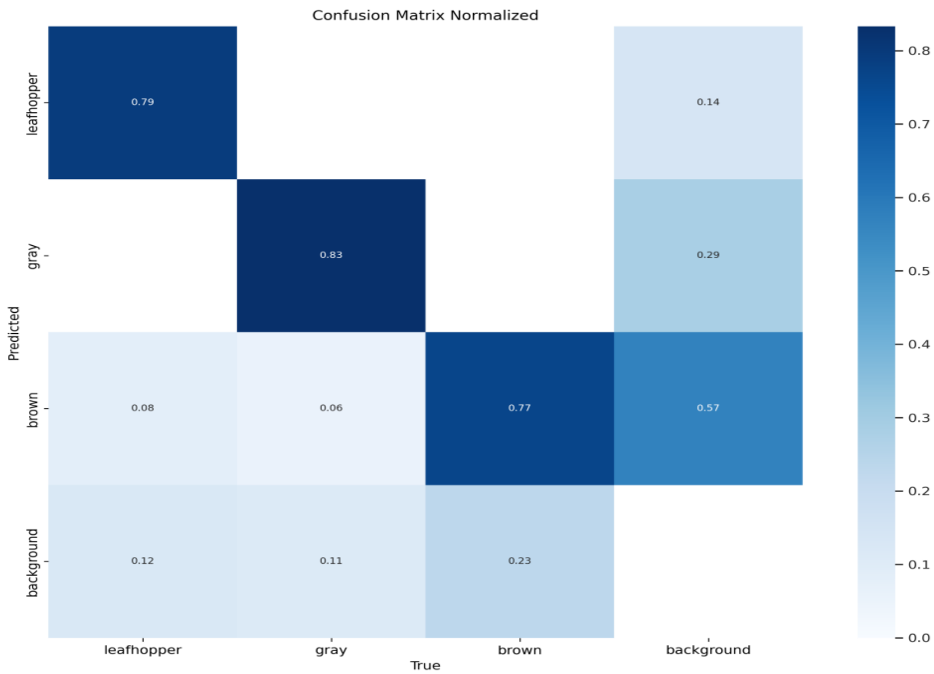


Figure 10: The normalized confusion matrix

Figure 11: Diseases and pests in tea identifying results

Here are the results of the images of tea diseases and pests detected by the model YOLOv10s as shown in Figure 11. All three tea diseases, gray, brown and leafhopper, were detected by the model, with cross-combination ratios exceeding 0.5. YOLOv10s performs exceptionally well when dealing with partially blurred leaves, thanks to its multi-scale feature fusion technology. However, when the blurring is extremely high, the model may encounter difficulties. When handling overlapping pests, the model can recognize multiple targets simultaneously, but it may struggle when the pests are very tightly overlapped. The model exhibits robustness to lighting conditions and maintains stable performance under varying lighting. Nevertheless, under extreme lighting conditions, the model may be disturbed and unable to accurately identify targets.

5. Tea diseases and pests detection system

In software development, graphical user interface(GUI), is essential components (Kirsan et al., 2024) .In order to design a graphical user interface to recognize tea pests and diseases, we used python language and PyQT5 to design it.PyQT5 is built on the famous Qt framework, which is backed by a large community of developers and documentation, and supports cross-platform desktop application development. The Qt Designer is a generic visual interface that enhances the functionality of PyQt5 by simplifying the process of creating user interfaces(Nabijonovich & Najmiddin, 2024).

First, we need to have the PyQt5 library installed. PyQt5 is a layer of interface between Python and the Qt library. The QWidget class is the top-level interface component in PyQT5, and can be used as a window, a parent for other subcomponents, or a panel to which various other UI component elements can be added.PyQT5 applications are started by QApplication class, which supports command line parameter selection and runs independently as the main program. QMainWindow is a container for desktop applications, which supports menu bar, toolbar and status bar(Rao et al., 2024).To build a desktop application, the first interface container should be a QMainWindow instance. As shown in Figure 12,We have used PyQt5 to design a basic interface with buttons, image display window, text display box.

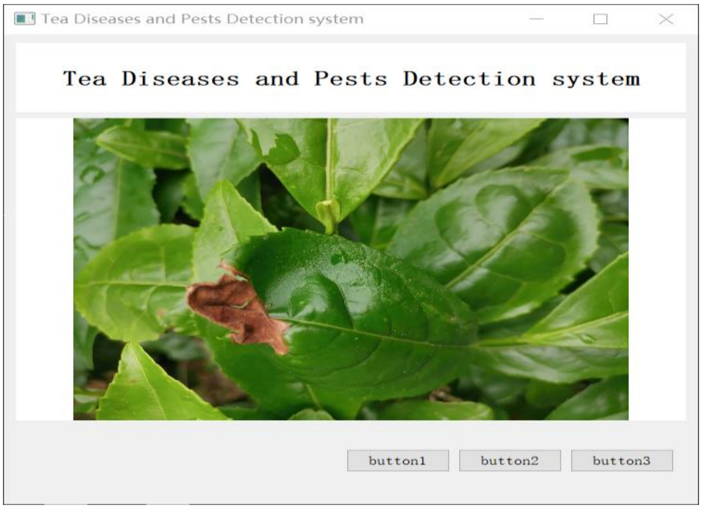


Figure 12: PyQT5 base applications interface

Based on the best trained Yolov10s target detection model with the UI interface made by Pyqt5, a tea disease intelligent detection system with simple interface is developed in python, which can support picture, video and camera detection, and meanwhile, the picture or video detection results can be saved. The objective is to supply a user-friendly operation platform for the inspection system so that the user can carry out the inspection task conveniently and efficiently. Through the graphical user interface (GUI), users can easily switch between picture, video and camera real-time inspection and operate the system without having to master complex programming skills. This not only improves system usability and user experience, but also makes the inspection process more intuitive and transparent enabling immediate monitoring and interpretation of outcomes. In addition, the GUI can be integrated with other functions, such as saving and exporting of detection results and adjustment of detection parameters, thus providing users with a comprehensive and integrated detection of tea diseases and promoting the wide use of intelligent detection technology for tea diseases.

In order to design a complete human-computer interface for basic detection of tea diseases, we need to design carefully and debug continuously. As shown in Figure 13, Confidence threshold, that is the conf parameter when target detection, the result will be displayed only if the confidence of the detected target is greater than this value.

Intersection and merge ratio threshold, that is the intersection over union (IOU)parameter when target detection, only if the intersection and merge ratio of the target detection frame is greater than this value, the result will be displayed.

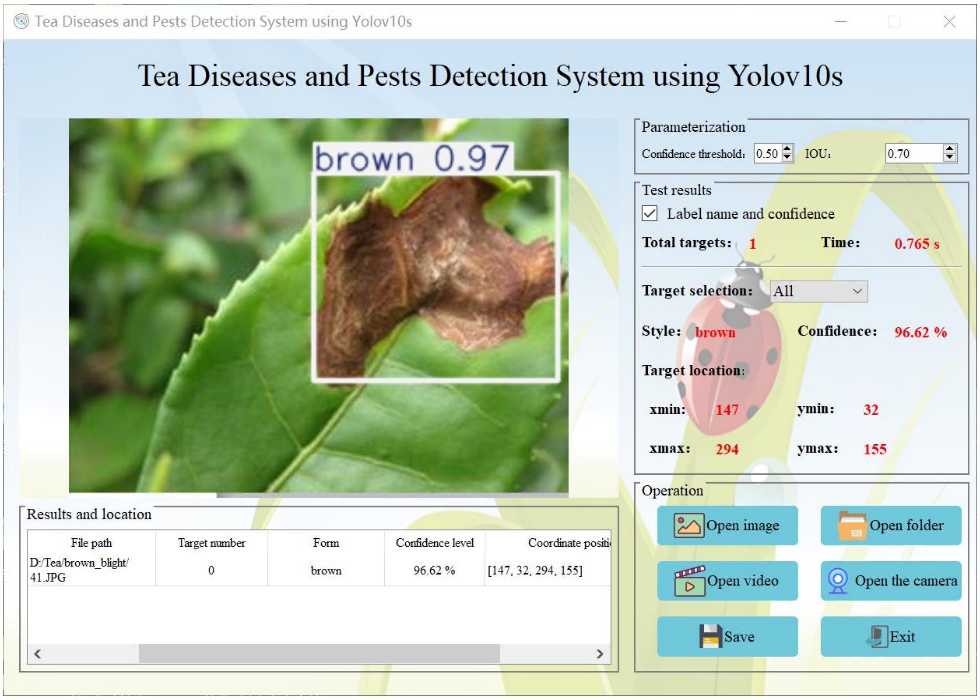


Figure 13: This tea disease detection system interface

From Figure 13, we can see that the detected tea disease of the image is brown_blight,the image has only one disease and the detection takes 0.765s.The tea disease style, location, and confidence level are shown. This tea disease detection system interface is designed to promote tea plantation disease control towards intelligence, precision and efficiency.

6. Conclusions, Study limitation, Implication and future work

6.1 Conclusions

At the moment, the ability of experts to identify diseases and pests in tea is primarily dependent on their experience. Hence, in order to recognize tea leaf pests and diseases image, this research suggests recent YOLOv10s object detection methods. The primary goal of the comprehensive evaluation and comparison of the three trained models: YOLOv8s, YOLOv9s, and YOLOv10s on the validation set was to identify the advantages and disadvantages of each model in terms of important metrics like Precision, Recall, mAP50 and F1-Score. The YOLOv10s model performs the most effective in recognition accuracy and speed. This not only aids in choosing the best model for a given set of requirements in real-world applications, but it also directs later attempts at model refining and optimization to increase detection speed and accuracy. Additionally, we conducted comprehensive testing of the entire system, ultimately developing a smooth, high-precision object detection system interface, which is the presentation interface of this paper, including a complete UI interface, test images, and videos.

In the end, we may get a deeper understanding of the resilience, generalization capacity, and detection performance across several categories thanks to this methodical comparison and study, which lays a strong foundation for creating computer vision systems that are more effective.

6.2 Study limitation

Despite the promising results, the study has several limitations. Firstly, the image used for training and validation was collected from a specific region, which may restrict the model's ability to generalize to other regions with different tea varieties and pest distributions. Secondly, the model's performance in identifying rare or uncommon pests and diseases has not been thoroughly evaluated, as these cases may not have been adequately represented in the image. Additionally, the current research focuses on still images, and the model's performance in real-time video surveillance or under varying lighting conditions has not been tested.

6.3 Implication

The results of this study carry significant importance for the tea industry. By utilizing advanced AI and deep learning methodologies, the efficiency and precision of identifying tea leaf diseases and pests can be considerably enhanced. This enables earlier detection and intervention, minimizing pesticide usage and enhancing the overall quality and sustainability of tea production. Additionally, the creation of a user-friendly recognition interface can ease the adoption of this technology by farmers and tea processors, empowering them to make more knowledgeable decisions regarding pest management and crop wellbeing.

6.4 Future work

We will keep refining the model in subsequent work, looking for more effective techniques with fewer parameters. Additionally, we'll investigate how to use the tea leaf diseases and pests recognition model. To fully capture the process of tea diseases data, more photos gathered at varying heights will be needed in the future. Additionally, more training samples will be needed to boost recognition rates. Drones and the detection technique can also be used together to decrease the amount of model parameters and accomplish real-time detection. The use of drones, sensors, and cameras to collect farm data raises serious privacy concerns. Personal information such as farmer identity, farm layout, and crop varieties may be inadvertently captured and misused. To address these issues, robust data encryption, secure storage solutions, and stringent access controls must be implemented. Farmers should be informed of data collection practices and obtain their consent before any data is collected. It was also mentioned that further consideration should be given to the potential ethical and environmental consequences arising from the application of pesticides based on insect detection.

Nonetheless, certain hardware configurations are still needed for quick detection. We think it has a lot to offer smart agriculture, green agriculture, and sustainable agriculture.

References

Chen, J., Liu, Q., & Gao, L. (2019). Visual Tea Leaf Disease Recognition Using a Convolutional Neural Network Model. *Symmetry*, *11*(3). https://doi.org/10.3390/sym11030343

Chen, X. (2024). AW-YOLOv8: A novel deep learning model for detecting insect pests in cotton growth systems. *Heliyon*. https://doi.org/10.1016/j.heliyon.2024.e32405

Gui, J., Wu, J., Wu, D., Chen, J., & Tong, J. (2024). A lightweight tea buds detection model with occlusion handling. *Journal of Food Measurement and Characterization*, *18*(9), 7533-7549. https://doi.org/10.1007/s11694-024-02746-w

Hossain, M. S., Mou, R. M., Hasan, M. M., Chakraborty, S., & Razzak, M. A. (2018). Recognition and detection of tea leaf's diseases using support vector machine. IEEE International Colloquium on Signal Processing & Its Applications, https://doi.org/10.1109/CSPA.2018.8368703

Hu, G., & Fang, M. (2022). Using a multi-convolutional neural network to automatically identify small-sample tea leaf diseases. *Sustain. Comput. Informatics Syst.*, *35*, 100696. https://doi.org/10.1016/j.suscom.2022.100696

Hu, G., Wang, H., Zhang, Y., & Wan, M. (2021). Detection and severity analysis of tea leaf blight based on deep learning. *Computers & Electrical Engineering*, *90*(1), 107023. https://doi.org/10.1016/j.compeleceng.2021.107023

Kirsan, A. S., Takano, K., & Mansurina, S. T. Z. (2024). EksPy: a new Python framework for developing graphical user interface based PyQt5. *International Journal of Electrical & Computer Engineering (2088-8708)*, *14*(1). https://doi.org/10.11591/ijece.v14i1.pp520-531

Krisnandi, D., Pardede, H. F., Yuwana, R. S., Zilvan, V., & Rahadi, V. P. (2019). Diseases Classification for Tea Plant Using Concatenated Convolution Neural Network. *CommIT (Communication and Information Technology) Journal*, *13*(2). https://doi.org/10.21512/commit.v13i2.5886

Lawal, O. M., Zhu, S., & Cheng, K. (2023). An improved YOLOv5s model using feature concatenation with attention mechanism for real-time fruit detection and counting. *Frontiers in Plant Science*, *14*. https://doi.org/10.3389/fpls.2023.1153505

Lee, S. H., Wu, C. C., & Chen, S. F. (2018). Development of Image Recognition and Classification Algorithm for Tea Leaf Diseases Using Convolutional Neural Network. 2018 Detroit, Michigan July 29 - August 1, 2018, https://doi.org/10.13031/aim.201801254

Lin, J., Bai, D., Xu, R., & Lin, H. (2023). TSBA-YOLO: An improved tea diseases detection model based on attention mechanisms and feature fusion. *Forests*, *14*(3), 619. https://doi.org/10.3390/f14030619

Liu, J. (2024). Research on Intelligent Recognition for Plant Pests and Diseases Based on Improved YOLOv8 Model. *Applied Sciences*, *14*. https://doi.org/10.3390/app14125353

Ma, N., Su, Y., Yang, L., Li, Z., & Yan, H. (2024). Wheat Seed Detection and Counting Method Based on Improved YOLOv8 Model. *Sensors*, *24*(5), 1654. https://doi.org/10.3390/s24051654

Nabijonovich, S. B., & Najmiddin, G. (2024). Optimizing pyqt5 development with qt designer. *Web of Teachers: Inderscience Research*, *2*(4), 254-259. https://webofjournals.com/index.php/1/article/view/1224

Rao, A. S., Sai, D., Mahajan, A. T., Singh, V. P., Neiwal, R., Pasupuleti, H., & Sudarsan, S. (2024). Development of Python-Based Applications for Virtual Instrument Control Using PyQt5, PyVISA, and SCPI Protocol. 2024 Second International Conference on Emerging Trends in Information Technology and Engineering (ICETITE), https://doi.org/10.1109/ic-ETITE58242.2024.10493634

Sapkota, R., Meng, Z., Ahmed, D., Churuvija, M., Du, X., Ma, Z., & Karkee, M. (2024). Comprehensive Performance Evaluation of YOLOv10, YOLOv9 and YOLOv8 on Detecting and Counting Fruitlet in Complex Orchard Environments. *arXiv preprint arXiv:2407.12040*. https://doi.org/10.48550/arXiv.2407.12040

Terven, J., Cordova-Esparza, D. M., & Romero-Gonzalez, J. A. (2023). A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS. *Machine Learning and Knowledge Extraction*, *5*(4), 1680-1716. https://doi.org/10.3390/make5040083

Vo, H.-T., Mui, K. C., Thien, N. N., & Tien, P. P. (2024). Automating Tomato Ripeness Classification and Counting with YOLOv9. *International Journal of Advanced Computer Science & Applications*, *15*(4). https://doi.org/10.14569/ijacsa.2024.01504113

Wang, A., Chen, H., Liu, L., Chen, K., Lin, Z., Han, J., & Ding, G. (2024). Yolov10: Real-time end-to-end object detection. *arXiv preprint arXiv:2405.14458*. https://doi.org/10.48550/arXiv.2405.14458

Wang, A., Peng, T., Cao, H., Xu, Y., Wei, X., & Cui, B. (2022). TIA-YOLOv5: An improved YOLOv5 network for real-time detection of crop and weed in the field. *Frontiers in Plant Science*, *13*, 1091655. https://doi.org/10.3389/fpls.2022.1091655

Wang, C.-Y., Yeh, I.-H., & Liao, H.-Y. M. (2024). Yolov9: Learning What You Want To Learn Using Programmable Gradient Information. *arXiv preprint arXiv:2402.13616*. https://doi.org/10.48550/arXiv.2402.13616

Wang, J., & Wang, J. (2024). A lightweight YOLOv8 based on attention mechanism for mango pest and disease detection. *Journal of Real-Time Image Processing*, *21*(4). https://doi.org/10.1007/s11554-024-01505-w

Wang, Z., Zhang, S., Chen, L., Wu, W., Wang, H., Liu, X., Fan, Z., Wang, B., & Xuan, M. L. (2024). Microscopic Insect Pest Detection in Tea Plantations: Improved YOLOv8 Model Based on Deep Learning. *Agriculture*, *14*(10). https://doi.org/10.3390/agriculture14101739

Xiao, B., Nguyen, M., & Yan, W. (2023). Fruit ripeness identification using YOLOv8 model. *Multimedia Tools and Applications*, *83*, 1-18. https://doi.org/10.1007/s11042-023-16570-9

Xue, Z., Xu, R., & Bai, D. L., Haifeng. (2023). YOLO-Tea: A Tea Disease Detection Model Improved by YOLOv5. *Forests*, *14*(2). https://doi.org/10.3390/f14020415

Ye, R., Gao, Q., Qian, Y., Sun, J., & Li, T. (2024). Improved Yolov8 and Sahi Model for the Collaborative Detection of Small Targets at the Micro Scale: A Case Study of Pest Detection in Tea. *Agronomy*, *14*(5), 1034. https://doi.org/10.3390/agronomy14051034

Zhao, B., Guo, A., Ma, R., Zhang, Y., & Gong, J. (2024). YOLOv8s-CFB: a lightweight method for real-time detection of apple fruits in complex environments. *Journal of Real-Time Image Processing*, *21*(5), 1-13.

https://doi.org/10.1007/s11554-024-01543-4

Zhao, Y., Yang, Y., Xu, X., & Sun, C. (2023). Precision detection of crop diseases based on improved YOLOv5 model. *Frontiers in Plant Science*, *13*, 1066835. https://doi.org/10.3389/fpls.2022.1066835

Zhu, X., Jia, B., Huang, B., Li, H., Liu, X., & Seah, W. K. G. (2024). Pest-YOLO: A Lightweight Pest Detection Model Based on Multi-level Feature Fusion. International Conference on Intelligent Computing, https://doi.org/10.1007/978-981-97-5591-2_12