



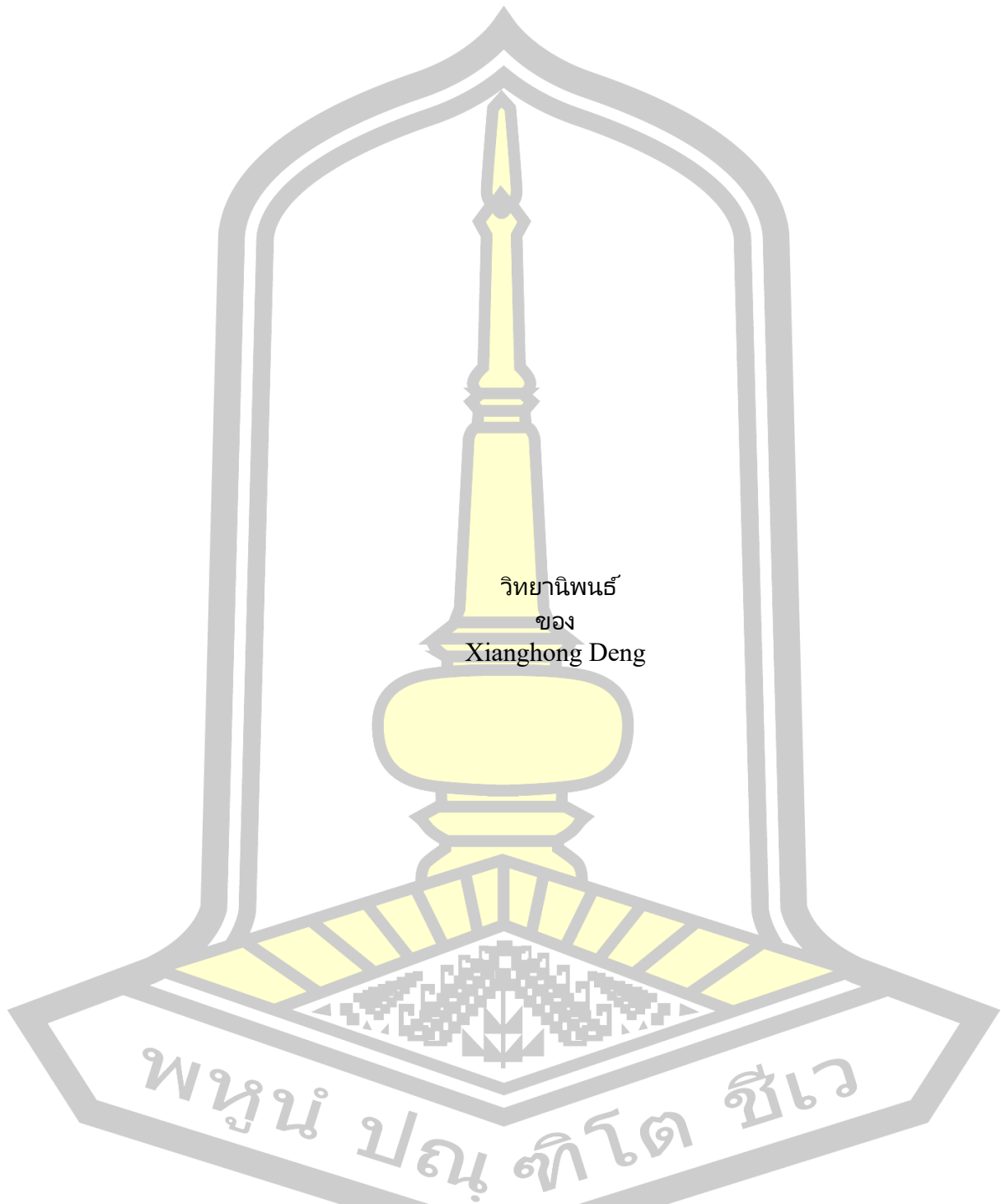
Identification of Diseases and Pests in Tea Leaves based on Deep Learning Techniques

Xianghong Deng

A Thesis Submitted in Partial Fulfillment of Requirements for
degree of Doctor of Philosophy in Electrical and Computer Engineering
February 2025

Copyright of Maharakham University

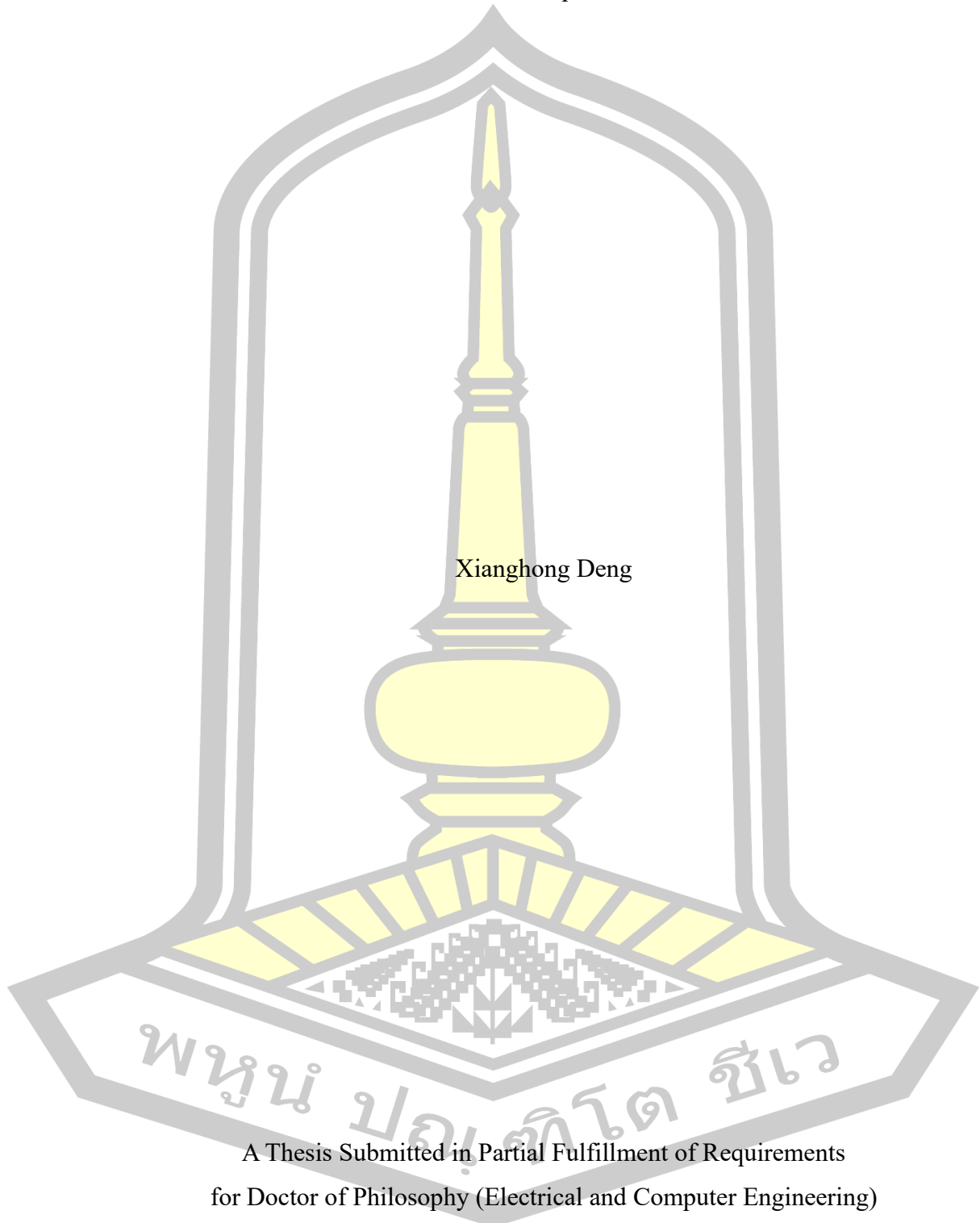
การระบุโรคและศัตรูพืชสำหรับใบชาด้วยเทคนิคการเรียนรู้ของเครื่อง



วิทยานิพนธ์
ของ
Xianghong Deng

เสนอต่อมหาวิทยาลัยมหาสารคาม เพื่อเป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาปรัชญาดุษฎีบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์
กุมภาพันธ์ 2568
ลิขสิทธิ์เป็นของมหาวิทยาลัยมหาสารคาม

Identification of Diseases and Pests in Tea Leaves based on Deep Learning
Techniques



Xianghong Deng

A Thesis Submitted in Partial Fulfillment of Requirements
for Doctor of Philosophy (Electrical and Computer Engineering)

February 2025

Copyright of Mahasarakham University



The examining committee has unanimously approved this Thesis, submitted by Mr. Xianghong Deng , as a partial fulfillment of the requirements for the Doctor of Philosophy Electrical and Computer Engineering at Mahasarakham University

Examining Committee

Chairman

(Assoc. Prof. Anan Kruesubthaworn , Ph.D.)

Advisor

(Assoc. Prof. Chonlatee Photong , Ph.D.)

Committee

(Prof. Worawat Sa-Ngiamvibool , Ph.D.)

Committee

(Asst. Prof. Buncha Wattana , Ph.D.)

Committee

(Asst. Prof. Nawarat Piladaeng , Ph.D.)

Mahasarakham University has granted approval to accept this Thesis as a partial fulfillment of the requirements for the Doctor of Philosophy Electrical and Computer Engineering

(Assoc. Prof. Juckamas Laohavanich , Ph.D.)

Dean of The Faculty of Engineering

(Prof. Anongrit Kangrang , Ph.D.)

Acting Dean of Graduate School

TITLE	Identification of Diseases and Pests in Tea Leaves based on Deep Learning Techniques		
AUTHOR	Xianghong Deng		
ADVISORS	Associate Professor Chonlatee Photong , Ph.D.		
DEGREE	Doctor of Philosophy	MAJOR	Electrical and Computer Engineering
UNIVERSITY	Maharakham University	YEAR	2025

ABSTRACT

Tea is an significant crop and deeply loved by individuals.However, various climatic factors and other factors may lead to tea diseases, resulting in the decline of tea quality and yield. In earlier times, 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, 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,so this paper introduces a method with improved efficiency and precision for intelligent identification in tea leaf diseases and pests.The main research contents and conclusions of this thesis are as follows:

(1)Three classification models VGG16, Resnet50 and DenseNet169 were used to categorize and identify common amongst diseases of tea leaf. By accommodating the relevant parameters of these three identification models and the learning rates, the performances of the three convolutional neural networks were analyzed and compared. The simulated test results showed that VGG16, Resnet50, and DenseNet169 achieved the highest accuracy rates of 90.6%, 95.2%, and 99.0%, respectively.DenseNet169 provided the fastest convergence speed and the highest accuracy among the models.

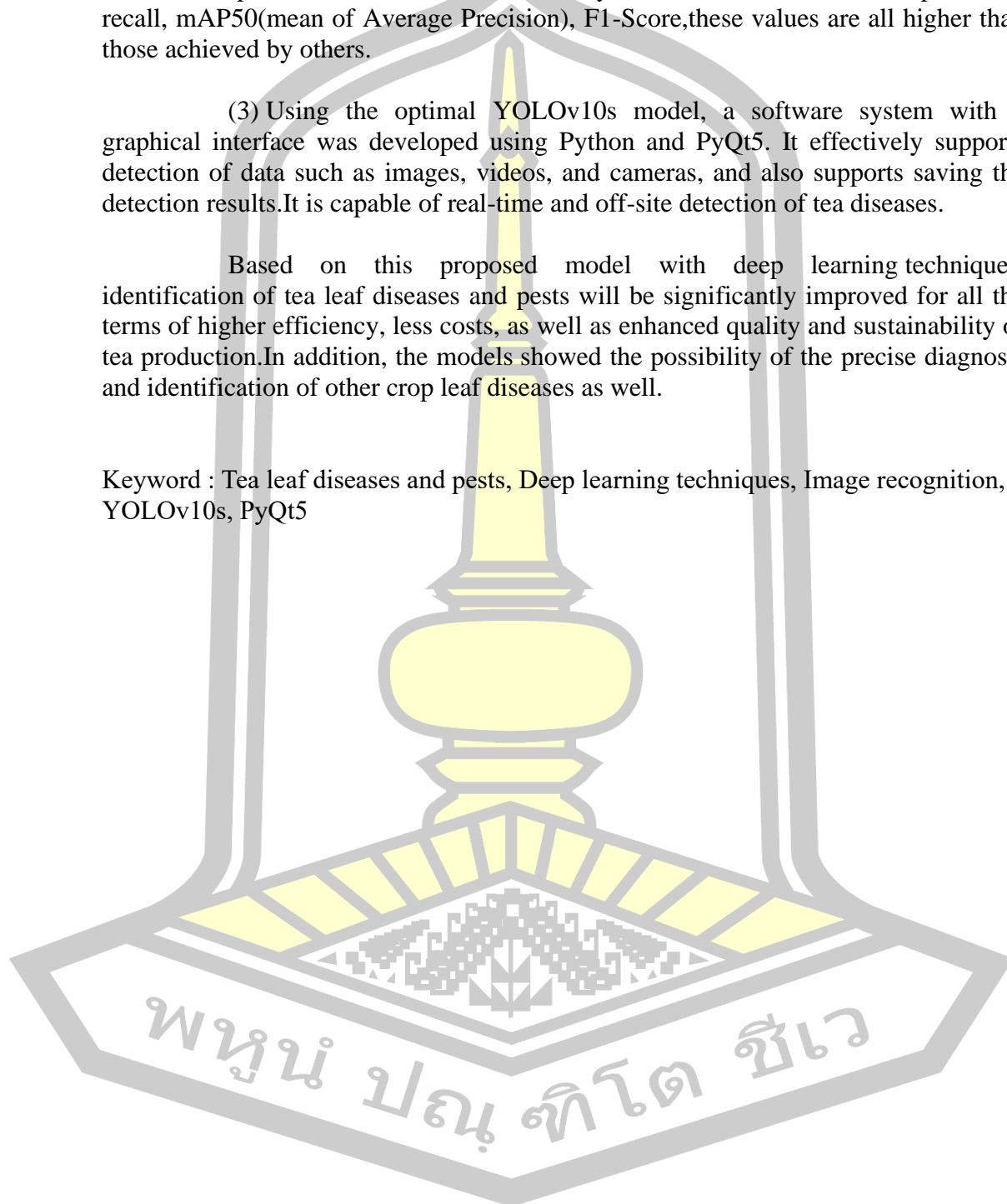
(2)The deep learning target detection models were proposed, for automated recognition of common tea leaf diseases and pests. Firstly,this study proposed an improved YOLOv5 (You Only Look Once version 5) algorithm method based on deep learning,which improved the model structure of YOLOv5 and integrate Omni Dimensional Dynamic Convolution (ODConv) and Convolutional Block Attention Module (CBAM) attention mechanisms into YOLOv5. The results show that the accuracy of the improved YOLOv5 is higher than that before the improvement and superior to other methods, including Faster R-CNN(Faster region-based convolutional neural networks), SSD(Single Shot MultiBox Detector) .Then,

the latest models of deep learning target detection YOLO such as YOLOv10s, YOLOv9s and YOLOv8s were applied to automatic identification of tea pests and diseases. 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 others.

(3) Using the optimal YOLOv10s model, 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.

Based on this proposed model with deep learning techniques, 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. In addition, the models showed the possibility of the precise diagnosis and identification of other crop leaf diseases as well.

Keyword : Tea leaf diseases and pests, Deep learning techniques, Image recognition, YOLOv10s, PyQt5



ACKNOWLEDGEMENTS

During my research and life as a PhD student at Mahasarakham University, I sincerely thank my teachers, classmates, relatives and university leaders in the College of Engineering who have given me guidance and help. During the three years of my doctoral degree, I have benefited a lot, not only to learn the theoretical knowledge of my research in depth and improve my professional and technical level, but also to learn the spirit of scientific research from my supervisors, lecturers and fellow students who are studying for the doctoral degree, and I will do my research seriously and rigorously, study hard and do research on the ground, no matter it is now or in the future. I would like to thank my teachers and classmates again.

It is not easy to study for a doctoral degree, and I have to thank Prof. Chonlatee Photong first of all for giving me this opportunity to receive a doctoral education, otherwise I would not be able to have this experience in my life. Once a teacher, always a teacher. During these three years, I have benefited a lot from my mentor, who took his precious time out of his busy schedule to communicate with me, enlighten me and set foot on the path of scientific research. He is very knowledgeable and kind, and took great pains to guide me into a good state of scientific research.

Finally, thanks to my family. I am deeply grateful to my loved ones for their support and encouragement of my studies!

Xianghong Deng

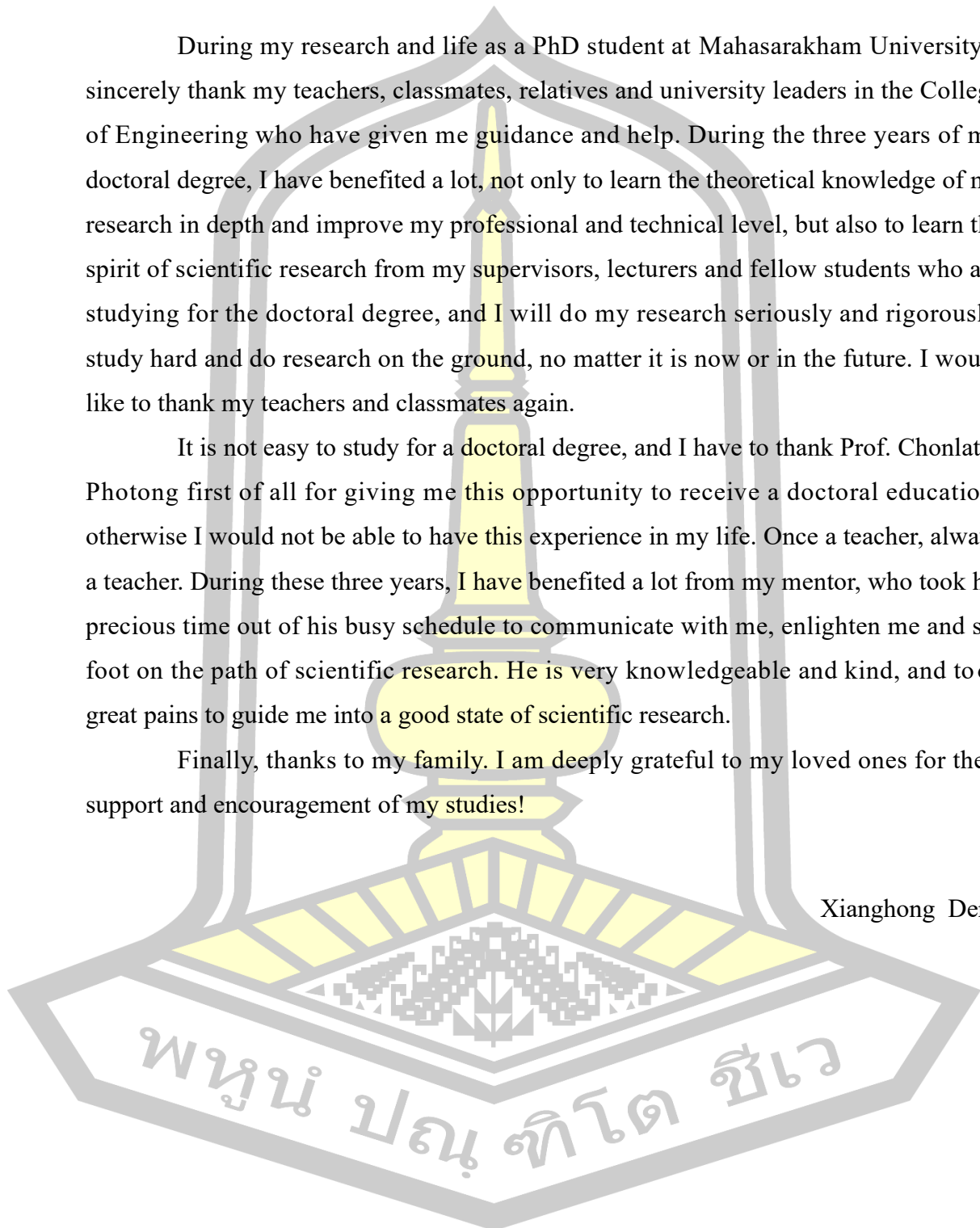
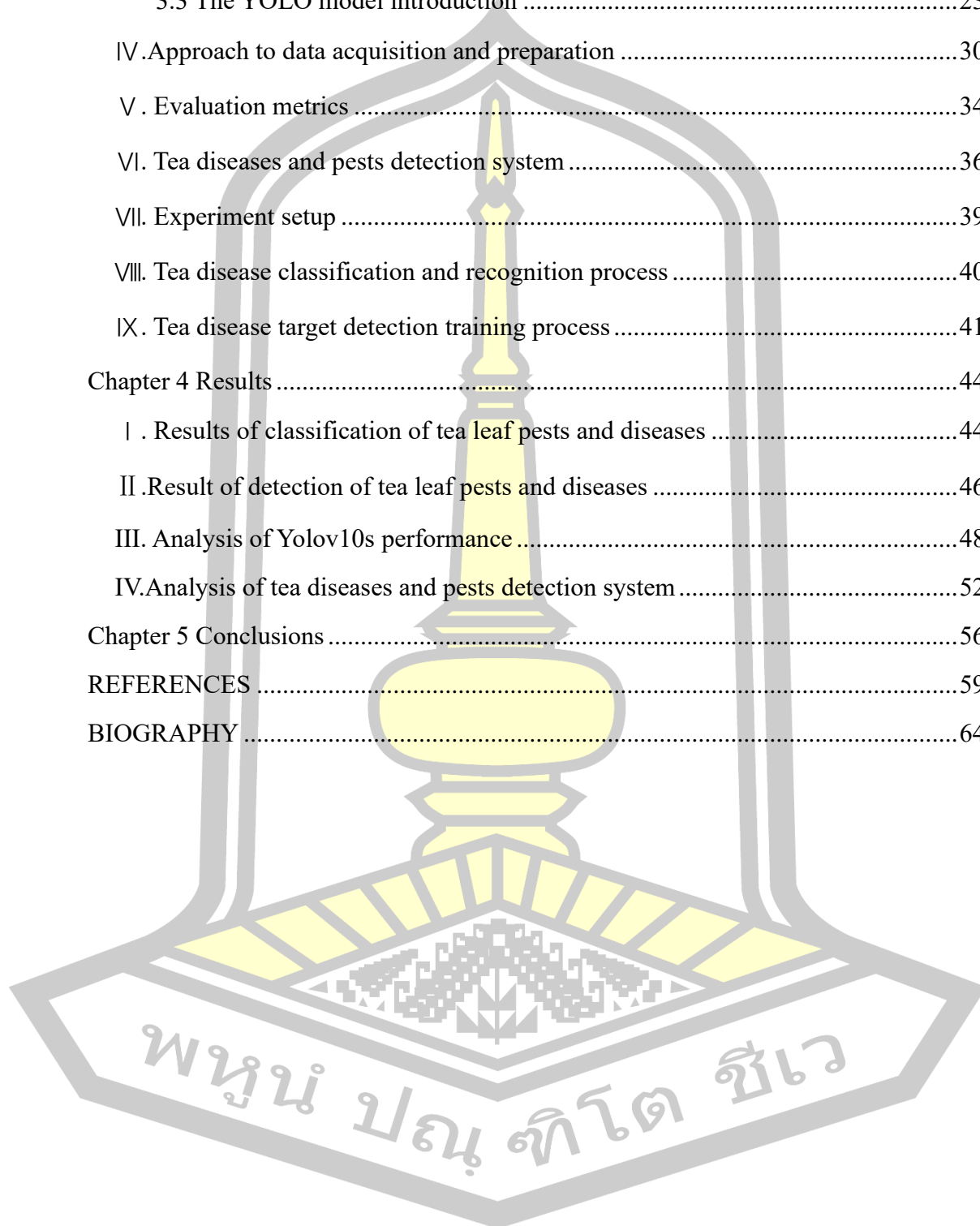


TABLE OF CONTENTS

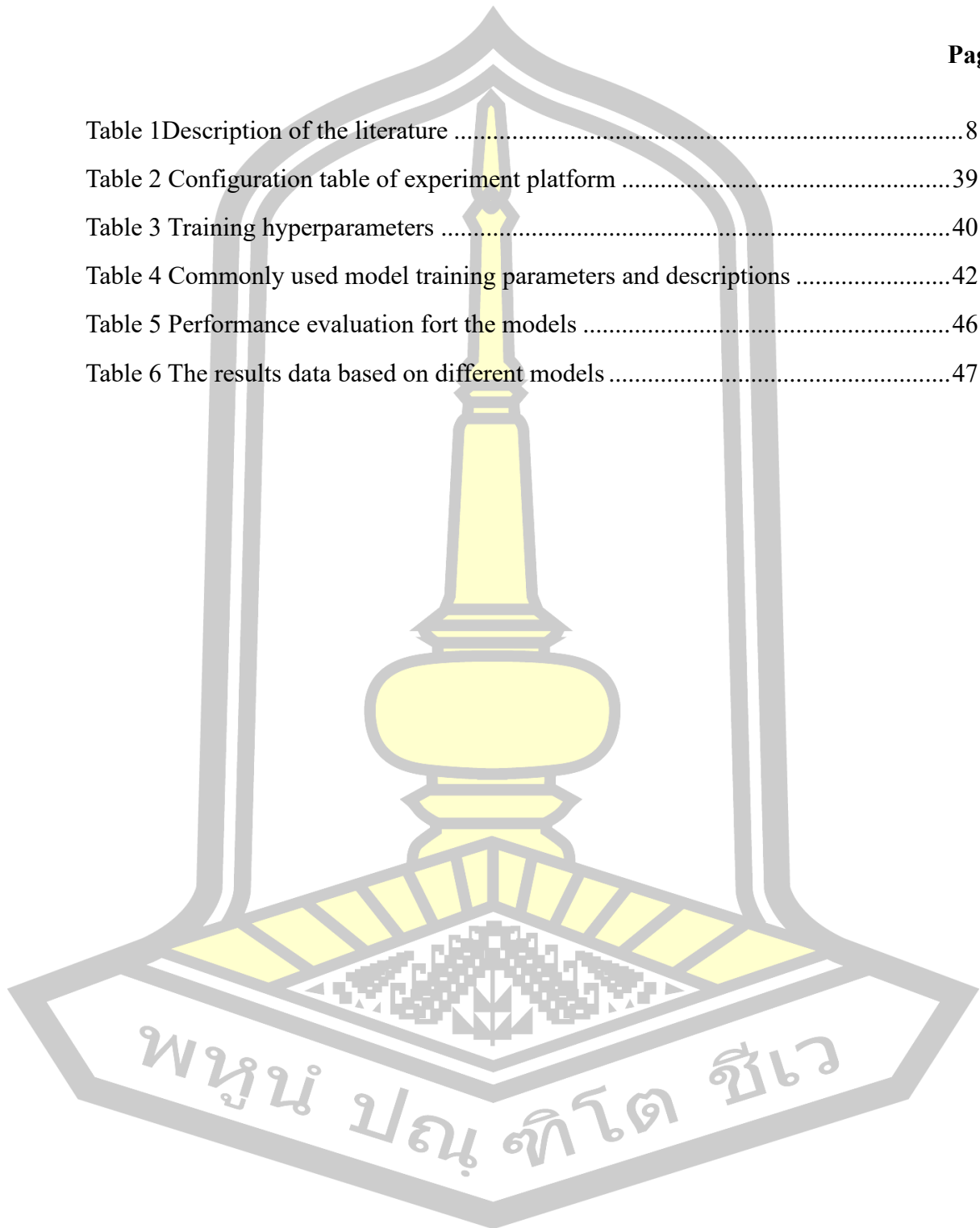
	Page
ABSTRACT.....	D
ACKNOWLEDGEMENTS.....	F
TABLE OF CONTENTS.....	G
List of Table.....	I
List of Figure.....	J
Chapter 1 Introduction.....	1
I . Research background.....	1
II . Objectives.....	4
III . Benefits of the research.....	5
Chapter 2 Literature Reviews.....	6
I . Research status of classification of tea leaf pests and diseases.....	6
1.1 Convolution neural network.....	6
1.2. Deep learning neural networks model.....	7
II . Research status of detection of tea leaf pests and diseases.....	9
III . Summary.....	12
Chapter 3 Research Methodology.....	14
I . Research methods.....	14
II . Classification of tea leaf pests and diseases based on deep learning techniques.....	14
2.1 The VGG16 model introduction.....	16
2.2 ResNet50 model introduction.....	17
2.3 DenseNet model introduction.....	19
III . Detection of tea leaf pests and diseases based on deep learning techniques.....	20
3.1 The Faster R-CNN model introduction.....	21

3.2 The SSD model introduction	22
3.3 The YOLO model introduction	23
IV. Approach to data acquisition and preparation	30
V. Evaluation metrics	34
VI. Tea diseases and pests detection system	36
VII. Experiment setup	39
VIII. Tea disease classification and recognition process	40
IX. Tea disease target detection training process	41
Chapter 4 Results	44
I. Results of classification of tea leaf pests and diseases	44
II. Result of detection of tea leaf pests and diseases	46
III. Analysis of Yolov10s performance	48
IV. Analysis of tea diseases and pests detection system	52
Chapter 5 Conclusions	56
REFERENCES	59
BIOGRAPHY	64



List of Table

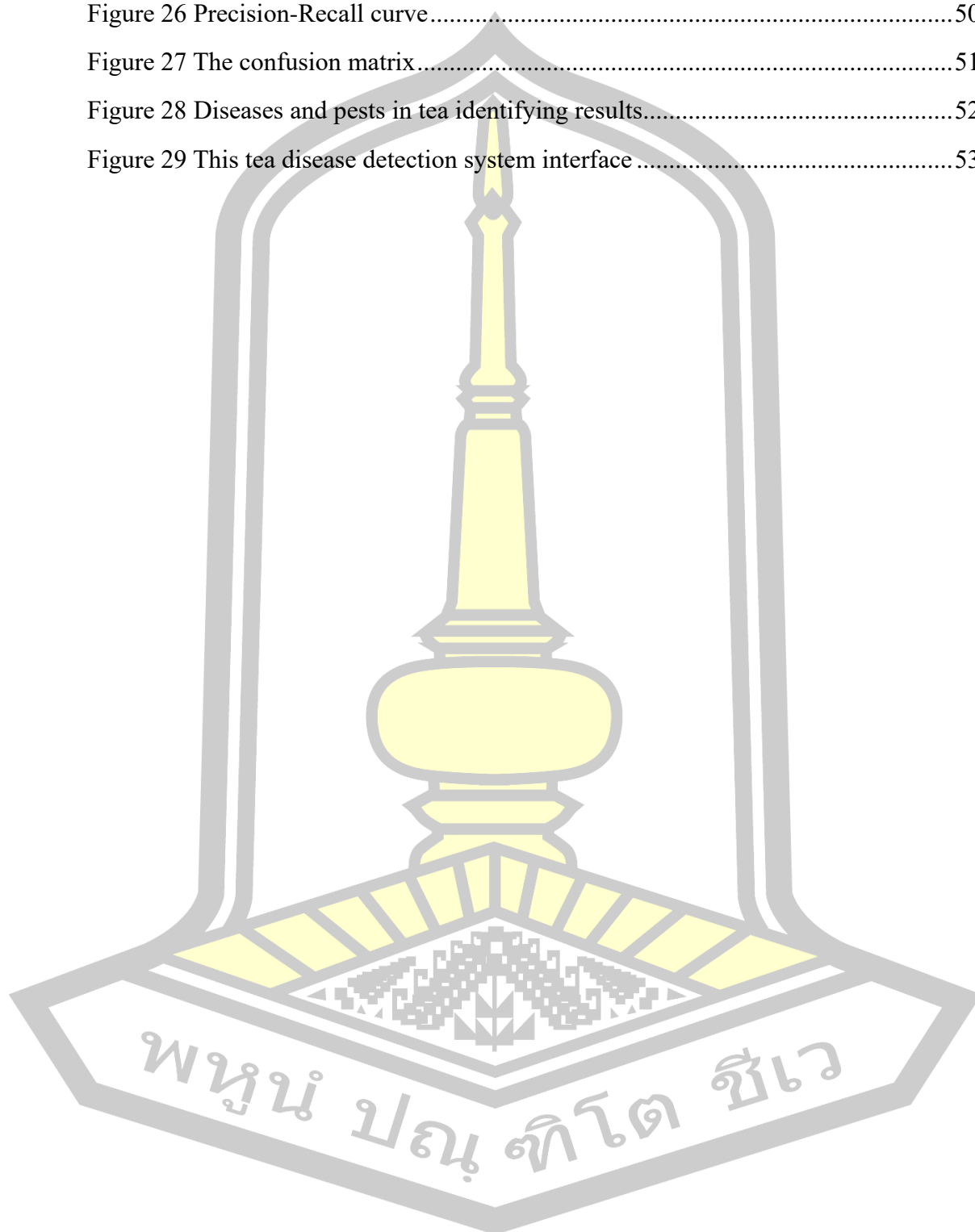
	Page
Table 1 Description of the literature	8
Table 2 Configuration table of experiment platform	39
Table 3 Training hyperparameters	40
Table 4 Commonly used model training parameters and descriptions	42
Table 5 Performance evaluation for the models	46
Table 6 The results data based on different models	47



List of Figure

	Page
Figure 1 Proposed Training flow chart of tea leaf pests and diseases classification model.....	15
Figure 2 VGG16 model architecture diagram.....	17
Figure 3 ResNet50 model architecture diagram	18
Figure 4 ResNet50 residual block structure.....	19
Figure 5 DenseNet model architecture diagram	19
Figure 6 Proposed Training flow chart of tea leaf pests and diseases detection model	20
Figure 7 Architectural sketch of Faster R-cnn	22
Figure 8 SSD model architecture diagram.....	23
Figure 9 The network structure of Yolov5s-5.3 model	24
Figure 10 The component networks in the original YOLOv5s.....	25
Figure 11 The network structure of the improved YOLOv5s	26
Figure 12 Comparison of CSP and CSPK Structure.....	27
Figure 13 The Yolov10 model network structure	29
Figure 14 The proposed tea diseases and healthy leaves	31
Figure 15 The dataset's representative samples.	32
Figure 16 The representative samples of image data enhancement.....	33
Figure 17 Image labeling fabrication.....	33
Figure 18 PyQT5 base applications interface	38
Figure 19 Data storage location	41
Figure 20 Comparison of the accuracy of three model validation.....	44
Figure 21 Comparison of loss function values for three model validation sets.....	45
Figure 22 Confusion matrix for DenseNet169.....	45
Figure 23 Box-validation loss comparison	46
Figure 24 The mAP50 comparison	47

Figure 25 Performance of the YOLOv10s model.....	49
Figure 26 Precision-Recall curve.....	50
Figure 27 The confusion matrix.....	51
Figure 28 Diseases and pests in tea identifying results.....	52
Figure 29 This tea disease detection system interface	53



Chapter 1 Introduction

I. Research background

Tea is an important economic crop [1]. Tea is famous as one of the three major beverages in the world and is called the emperor of oriental beverages. People love to drink tea, and tea contains many nutrients that people need. Therefore, there are many research works to increase the productivity of tea production [2, 3]. So the tea is an important cash crop, and tea trees are also planted in a large area on the earth.

However, in the process of tea planting, tea diseases can also reduce the quality of tea and cause serious economic losses to the tea plantation. Accurately identify tea diseases and take timely preventive and control measures to reduce tea production losses, improve tea quality, and improve the income of tea plantations is of great significance. According to statistics, there are more than 900 kinds of tea tree pests and diseases recorded, When the tea leaf is healthy the color is distinct, but when the leaf is affected by disease, the color of the leaf changes drastically. Each disease usually has a distinguishable leaf color and texture as symptoms. And the control of tea tree pests and diseases is also a major task of tea tree management.

Common tea leaf insects are found in tea gardens. Tea Green Leafhopper: This insect is one of the primary pests in tea gardens. It feeds on tea leaf sap, leading to leaf yellowing and reduced yields. Tea Tussock Moth: The larvae of this moth feed on tea leaves and can severely impact the growth of tea bushes. Tea Geometrid Moth: The larvae of this insect can also damage tea leaf foliage, causing leaf consumption. Tea Leafhopper: Another insect that feeds on tea leaf sap, potentially causing leaf to yellow and curling.

At present, tea leaf diseases are primarily identified by manual [4]. Advances in computing technology, such as the use of deep learning and digital image processing, can be used for the detection and identification of diseases in plants [5]. For the last few years, with the rapid development of smart agriculture and precision agriculture, computerized image recognition technology has been increasingly applied to solve many difficult problems in agricultural science [6]. With the rapid development of deep learning and the significant improvement in computational power, object detection algorithms have transitioned from traditional algorithms based on handcrafted features to object detection techniques based on deep learning.

Convolutional Neural Networks (CNNs) are used by the deep learning-based target detection technique to extract features[7]. Convolutional Neural Networks (CNNs) specialize in image and video recognition tasks. CNNs are able to automatically learn features from images, which makes them well suited for tasks such as image classification, object detection, and image segmentation. Deep learning-based models have outperformed previous artificial intelligence systems in a number of domains, including speech recognition, language processing, and computer vision[8].

Deep learning is a branch of machine learning that involves writing algorithms that mimic human brain functions. It is a change from manually selecting object features to selecting features through neural networks themselves. Deep learning also comprises some different deep learning algorithms. Deep learning refers to the use of artificial neural network architectures that contain a quite large number of processing layers. In order to improve learning performance, neural networks have a higher level of representation learning (deeper). Deep learning refers to changing the manual selection of object features to selecting features through neural networks themselves. In order to improve learning performance, neural networks have a higher level of representation learning (deeper).

Deep learning (especially Convolutional neural network, CNN) has made a breakthrough in image detection and identification. Compared with traditional methods, deep learning-based image classification methods have the following advantages:

Automatic feature extraction: Deep learning models can automatically learn features suitable for tasks without the need for manual design.

End-to-end training: Feature extraction and classifier training are conducted jointly, which helps to discover better feature representations.

High performance: Deep learning methods have achieved state-of-the-art performance in many image classification tasks, making them more accurate than traditional methods.

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.

Image classification and image object detection are two distinct tasks in deep learning, with their differences primarily manifested in their objectives and approaches. Image classification focuses on categorizing the entire image into a predefined class, while image object detection concerns recognizing multiple objects within an image along with their positions. Image classification is typically used for scenarios involving the recognition of single objects, whereas image object detection is applicable for detecting multiple objects.

The application of deep learning in the field of pest identification and the optimization of various algorithms have greatly improved the efficiency and accuracy of pest identification, but there are still many problems in the research and development of artificial intelligence in the identification of tea tree pests.

In practical applications, the environment of tea gardens is complex, light, weather, etc. That will have a certain impact on the collection of pictures, and when pests and diseases occur, they will be blocked by tea leaves and young shoots, which will have a certain impact on the recognition effect. The identification of dynamic tea garden pests is difficult to identify, and its identification accuracy needs to be improved. Target detection algorithm based on deep learning, as follows: 1. accuracy; 2. real-time performance; 3. multi-scale; 4. labeling scheme; 5. target overlap; 6. model training; 7. repetitive coding; 8. data enhancement; 9. sample imbalance, etc.

PyTorch is a fully functional framework for building deep learning models, commonly used in machine learning applications such as image recognition and language processing. PyTorch is mainly used to build neural networks and perform tensor operations. It is written in Python. It fully supports GPU and uses the reverse mode Automatic differentiation technology, so it can dynamically modify the calculation graph. PyTorch benefits from its streamlined and flexible interface design, allowing it to quickly build and debug network models, and has received rave reviews in the academic community.

II. Objectives

Regarding deep learning in the context of two distinct tasks: image classification and object detection, the key Objectives of this research are:

(1) Collecting and organizing datasets: collecting and organizing data pictures related to tea pests and diseases and carrying out corresponding data processing to provide training datasets for model training, Search and study more on how to classify the types of pests or diseases for tea leaves. An improved model for classifying tea plant diseases was proposed.

(2) Training model: The deep learning target detection models were proposed, for automated recognition of common tea leaf diseases and pests. Firstly, this study proposed an improved YOLOv5 (You Only Look Once version 5)[9] algorithm method based on deep learning, which improved the model structure of YOLOv5 and integrate Omni Dimensional Dynamic Convolution (ODConv)[10] and Convolutional Block Attention Module (CBAM)[11] attention mechanisms into YOLOv5. The results show that the accuracy of the improved YOLOv5 is higher than that before the improvement and superior to other methods, including Faster R-CNN(Faster region-based convolutional neural networks)[12], SSD(Single Shot MultiBox Detector)[13] .

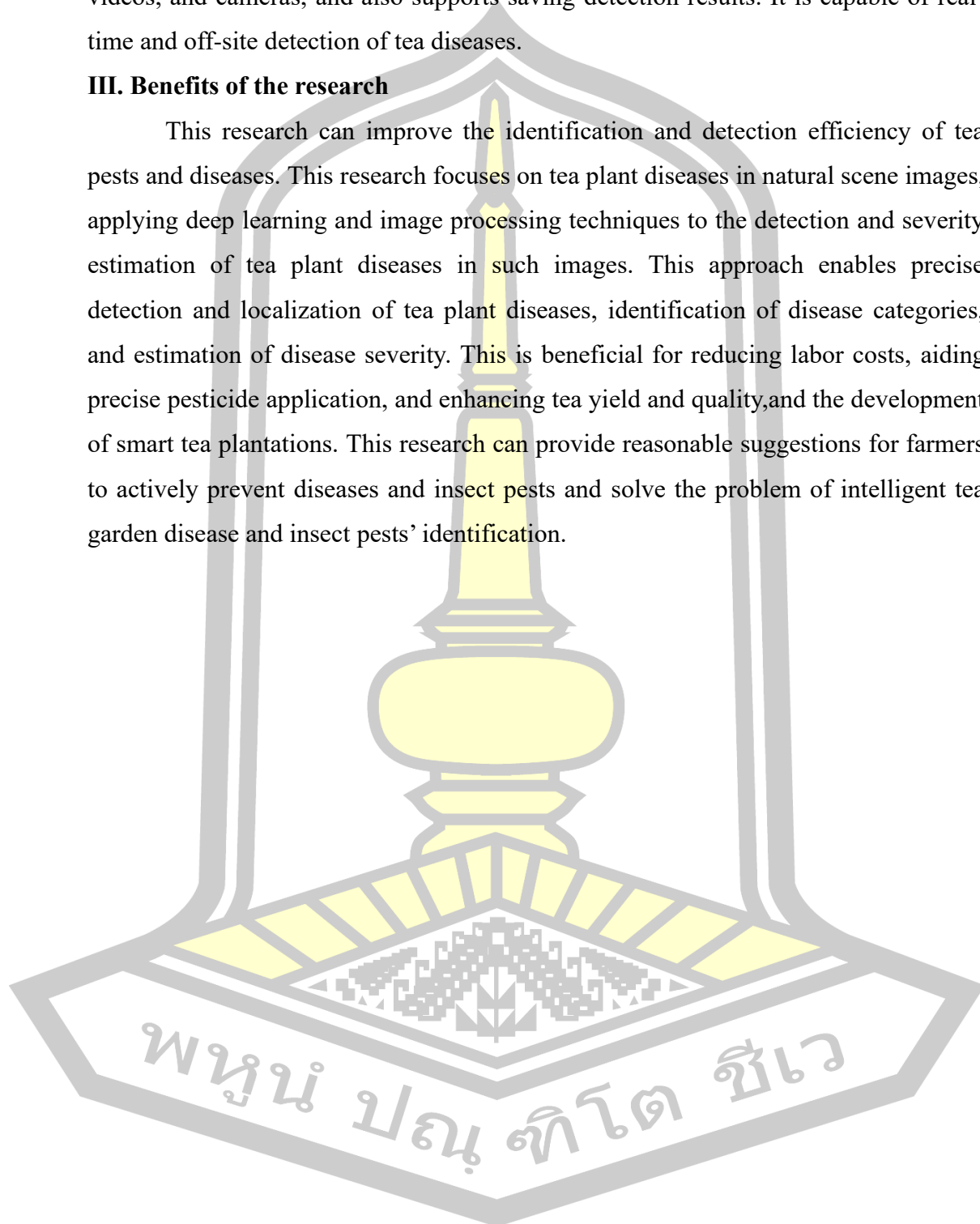
(3) Model Performance Comparison: Then, the latest models of deep learning target detection YOLO [14] such as YOLOv10s, YOLOv9s and YOLOv8s were applied to automatic identification of tea pests and diseases. The comparison of performance metrics indicated that the model based on YOLOv10s performed the best. As shown by the test evaluation results, precision, recall, mAP50(mean of Average Precision)[15], F1-Score[16], these values are all higher than those achieved by others. This not only helps us to select the most suitable model for specific needs in practical applications but also guides the subsequent model optimization and tuning efforts with a view to obtaining higher detection accuracy and speed. Ultimately, through this systematic comparison and analysis, we can better understand the robustness, generalization ability, and detection performance of the models on different categories, providing a solid foundation for developing more efficient computer vision systems.

(4) Visualization system production: Based on the best-performing 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 detection results. It is capable of real-time and off-site detection of tea diseases.

III. Benefits of the research

This research can improve the identification and detection efficiency of tea pests and diseases. This research focuses on tea plant diseases in natural scene images, applying deep learning and image processing techniques to the detection and severity estimation of tea plant diseases in such images. This approach enables precise detection and localization of tea plant diseases, identification of disease categories, and estimation of disease severity. This is beneficial for reducing labor costs, aiding precise pesticide application, and enhancing tea yield and quality, and the development of smart tea plantations. This research can provide reasonable suggestions for farmers to actively prevent diseases and insect pests and solve the problem of intelligent tea garden disease and insect pests' identification.



Chapter 2 Literature Reviews

In this chapter, the application techniques of deep learning for tea disease classification and target detection are studied, including traditional convolutional neural network techniques, existing deep learning techniques.

I. Research status of classification of tea leaf pests and diseases

1.1 Convolution neural network

In this research, an automated system has been developed for detecting three different types of tea leaf diseases using Support Vector Machine (SVM) with less number of features. The proposed method is able to classify the disease more accurately (93%) compared to the other classifiers and neural network (91%)[17]. In this paper, CNN model is used to recognize the image of tea leaf diseases. ReLU linear function is used as the activation function of neurons to accelerate the convergence of the network. By adjusting the learning rate, the number of iterations and the addition of dropout, the network can achieve high recognition accuracy, which is conducive to the identification and classification of tea diseases. The recognition algorithm based on CNN is better in classification and can improve the recognition efficiency of tea leaf diseases effectively[18]. This paper basically adopted convolutional neural network (CNN) model to detect and identified these two disease using convolutional neural network (CNN). The proposed works basically found a solution of the tea leaf disease detection using simplest method while keeping minimum computational complexity and minimal resource to gain fast and accurate result as convolutional neural network (CNN) automatically extracts features for classification of input image into various classes. The experimental resulted on the developed model achieved precision approximately 95.93%[19]. The CNN model was used to classify tea leaf disease using the MobileNetV2 backbone, Adam optimizer, and rectified linear unit (ReLU) activation function with 224×224 input data. The proposed model attained the highest performance, as evidenced by the accuracy value 0.9455[20]. In this proposed work, they have applied the Convolutional Neural Network(CNN) model with 1 input layer, 4 convolution layers, and 2 fully connected layers. The image is passed to the input layer. The convolution layers mainly extract features from the input image in the dataset and the output layer classifies the given

image to 8 classes such as the normal leaf, Algal leaf spot, Gray blight, White spot, Brown blight, Red scab, Bud blight, and Grey blight[21].

1.2. Deep learning neural networks model

In this study, the researchers propose the Convolutional Neural Network (CNN) for tea disease detections. The researchers focus on the implementation of concatenated CNN, namely GoogleNet, Xception, and Inception-ResNet-v2, for this task[22]. Specialized deep learning models were developed, based on specific convolutional neural networks architectures, for the identification of plant diseases through simple leaves images of healthy or diseased plants. The most successful model architecture, a VGG convolutional neural network, achieved a success rate of 99.53% (top-1 error of 0.47%) in the classification of 17,548 previously unseen by the model plant leaves images (testing set)[23].

Materials: A CNNs model named LeafNet was developed with different sized feature extractor filters that automatically extract the features of tea plant diseases from images. DSIFT (dense scale-invariant feature transform) features are also extracted and used to construct a bag of visual words (BOVW) model that is then used to classify diseases via support vector machine (SVM) and multi-layer perceptron (MLP) classifiers. The performance of the three classifiers in disease recognition were then individually evaluated. Results: The LeafNet algorithm identified tea leaf diseases most accurately, with an average classification accuracy of 90.16%[6].

The disease spot samples are augmented by C-DCGAN, and VGG16 deep learning networks are trained to realize disease spots identification. Experimental results show that the combination of traditional machine learning methods and deep learning methods can effectively identify tea leaf's diseases with complex background[4].

The purpose of this paper is to develop a deep CNNs called LeNet to discover the tea plant diseases from leaf image set. As a result, LeNet the perfect CNN model used in further applications to improve the diagnostic measurement of tea leaves as well as other plant leaves[24].

This paper adopted the Convolutional Neural Network (CNN) architectures, namely ResNet-50 and Inception V3, to classify six types of diseases that attack leaves. The batch sizes used are 16, 32, and 64. Using three different batch size scenarios for each model, it shows that ResNet-50 has better performance on batch size 32 with an accuracy value of 97.44%, while Inception V3 has the best performance on batch size 64 with an accuracy of

97.62%[5]. This study proposes a method based on an improved deep convolutional neural network (CNN) for tea leaf disease identification. A multiscale feature extraction module is added into the improved deep CNN of a CIFAR10-quick model to improve the ability to automatically extract image features of different tea leaf diseases. The depthwise separable convolution is used to reduce the number of model parameters and accelerate the calculation of the model[25]. Three models are used in this system in which two were pre-trained. They are Faster RCNN (Inception-v2), and VGG16. The other one is manually trained, that is Sequential model or CNN[26]. In this study, more than 1000 images of tea leaves are used to train the model based on faster region-based convolutional neural network (Faster R-CNN)[27]. The integration of multiple CNN modules enabled MergeModel to extract a variety of discriminative features. The identification ability of MergeModel was improved compared with a single neural network. The weight initialization method encoded the diseased leaf features into the convolution filter, which helped the model to concentrate on learning important features at the beginning of training. In addition, this study used a small number of diseased tea leaf images as the original training samples and generated new training samples through the unconditional generation model, namely, SinGAN, for data augmentation[28]. In this paper, They use pre-trained models on imageNet data and re-train them on our data to detect tea leaf diseases. Those pre-trained models use deep convolutional neural network (DCNN) architectures: VGGNet, ResNet, and Xception. To mitigate the difference tasks of ImageNet and themselves, they apply fine-tuning on the pre-trained models by replacing some parts of the pre-trained models with new structures[29]. Studies on the classification of tea diseases are summarized in Table 1.

Table 1 Description of the literature

The year of publication	Classification of Infections	The technique of training	Model Correctness
Xiaoxiao SUN et al[10]2018	Seven kinds	CNN	93.7%
Md. Selim Hossain et al[9]2018	Three typoss	Support Vector Machine	93%
Hu Gensheng et al [4]2019	Three kinds	VGG16	90%
Jing Chen et al[6]2019	Six kinds	CNN	90.16%
Dikdik Krisnandi et al[14]2019	Four kinds	GoogLeNet, Xception, and Inception-ResNet-v2	89.64%

The year of publication	Classification of Infections	The technique of training	Model Correctness
Hu Gensheng et al[20]2019	Three kinds	Improved deep CNN	92.5%
ShyamtanuBhowmik et al[11] 2020	Two kinds	CNN	95.93%
Gayathri et al[18]2020	Four kinds	LeNet model	90.23%
Mahadi Hasan Kamrul et al[21]2020	Five kinds	Faster RCNN	96.28%
Ade Ramdan et al [24]2020	Four kinds	ResNet	94.05%
R.S.Latha et al [13]2021	Six kinds	CNN	94.45%
Hu Gensheng et al[23] 2022	Five kinds	multi-convolutional neural network (CNN)	94%
Hairah et al[12]2024	Six kinds	MobileNetV2	94.55%
Liana et al[5] 2023	Five kinds	Inception V3	97%

II. Research status of detection of tea leaf pests and diseases

This study proposes an improved RetinaNet target detection and identification network, AX-RetinaNet, which is used for the automatic detection and identification of tea leaf diseases in natural scene images. AX-RetinaNet uses an improved multiscale feature fusion module of the X-module and adds a channel attention module, Attention. The feature fusion module of the X-module obtains feature maps with rich information through multiple fusions of multi-scale features. The attention module assigns a network adaptively optimized weight to each feature map channel so that the network can select more effective features and reduce the interference of redundant features. This study also uses data augmentation methods to solve the problem of insufficient samples[1]. A Retinex algorithm is utilized to enhance the original images and reduce the influence of light variation and shadow. The TLB (tea leaf blight) leaves are detected using a deep learning framework called Faster Region-based Convolutional Neural Networks, to improve the detection performance of blurred, occluded, and small pieces of diseased leaves. The detected TLB leaves are inputted into the trained VGG16 networks to achieve severity grading and facilitate disease severity analysis[30]. They proposed an improved tea disease detection model called TSBA-YOLO. They use the dataset of tea diseases collected at the Maoshan Tea Factory in China. The self-attention mechanism was used to enhance the ability of the

model to obtain global information on tea diseases. The BiFPN feature fusion network and adaptively spatial feature fusion (ASFF) technology were used to improve the multiscale feature fusion of tea diseases and enhance the ability of the model to resist complex background interference. They integrated the Shuffle Attention mechanism to solve the problem of difficult identifications of small-target tea diseases. Finally, SIoU was used to further improve the accuracy of the regression[31]. They propose YOLO-Tea, an improved model based on You Only Look Once version 5 (YOLOv5). Firstly, they integrated selfattention and convolution (ACmix), and convolutional block attention module (CBAM) to YOLOv5 to allow their proposed model to better focus on tea tree leaf diseases and insect pests. Secondly, to enhance the feature extraction capability of our model, they replaced the spatial pyramid pooling fast (SPPF) module in the original YOLOv5 with the receptive field block (RFB) module. Finally, they reduced the resource consumption of their model by incorporating a global context network (GCNet)[32].

The improved YOLOv5 has also been widely used in other plant diseases. In this study, a recognition method of soybean pods and estimation of pods weight per plant were proposed based on improved YOLOv5 model[33]. To accurately recognize plant diseases under complex natural conditions, an improved plant disease-recognition model based on the original YOLOv5 network model was established. First, a new InvolutionBottleneck module was used to reduce the numbers of parameters and calculations, and to capture long-distance information in the space. Second, an SE module was added to improve the sensitivity of the model to channel features. Finally, the loss function 'Generalized Intersection over Union' was changed to 'Efficient Intersection over Union' to address the former's degeneration into 'Intersection over Union'[34]. They proposed an improved YOLO v5 algorithm based on a shallow feature layer. There are two main core ideas: (1) to increase the perceptual field by adding quadruple down-sampling in the feature pyramid to improve the detection of small targets, and (2) introducing the CBAM attention mechanism into the neural network to solve the problem of gradient disappearance during training[35]. Aiming at the problem of peach tree leaf diseases in modern orchards, this work proposes a lightweight identification method based on the combination of improved YOLO v5 and ShuffleNet to achieve accurate identification of peach tree leaf diseases in the

natural environment[36]. In this paper, an improved SE-YOLOv5 network model is proposed for the recognition of tomato virus diseases[37]. this paper proposes a method based on YOLOv5s with an improved attention mechanism, which can accurately detect the number of small-scale wheat spikes and better solve the problems of occlusion and cross-overlapping of the wheat spikes[38]. Most current crop diseases present small targets, dense numbers, occlusions and similar appearance of different diseases, and the current target detection algorithms are not effective in identifying similar crop diseases. Therefore, in this paper, an improved model based on YOLOv5s was proposed to improve the detection of crop diseases[39]. Current detection methods for apple leaf diseases still suffer some challenges, such as the high number of parameters, low detection speed and poor detection performance for small dense spots, which limit the practical applications in agriculture. Therefore, an efficient and accurate model for apple leaf disease detection based on YOLOv5 is proposed and named EADD-YOLO[40].

Deep Learning Target Detection The latest models of YOLO, such as YOLOv10s, YOLOv9s and YOLOv8s, have also been used in crop detection, but less so in the detection of tea diseases. YOLOv8, which was released in 2023, is especially well-suited for edge deployment as it has additional improvements to build upon the success of YOLOv5 [41]. A new model called YOLOv8n-ShuffleNetv2-Ghost-SE was introduced in order to efficiently monitor apple fruits in smart orchards for the duration of their growth[42] . In the detection process, they introduced the self-developed lightweight down-sampling module (LDM) and the improved FC-FasterNet module. Additionally, they implemented the VoV-GSCSPC module to reduce computational complexity and integrate simAm to enhance attention to target details. To simulate real field conditions more realistically, we include samples of beneficial insects. The results indicate that the AW-YOLOv8 model achieves a 92.0% mean Average Precision (mAP) for cotton pest detection[43]. The utilization rate of automated tomato harvesting is low, which is addressed by a proposed tomato automatic identification approach based on an enhanced YOLOv8s model [44]. In order to maintain crop health, the Light-SA YOLOV8 (Lightweight Self-Attention YOLOV8) model is presented. It is designed especially for real-time detection in intricate environments[45]. The YOLOv8-DMAS model is suggested for the

identification of cotton weeds in intricate settings[46]. Xie et al.[47]proposed a original model named Tea-YOLOv8s to address the issues of low detection precision induced by the difficult environments of tea plant leaves. Based on YOLOv8, this work suggests YOLOv8-HD, a new wheat seed detection network[48]. The objective of the study is to introduce a new model based on the modified YOLOv8 architecture in order to improve the efficiency of tea diseases and pests' recognition [49]. With its strategic architectural changes aimed at overcoming the challenges of information loss in deep neural networks—a crucial factor in maintaining detection accuracy and system efficiency—YOLOv9 stands out in object identification [50]. The Generalized Efficient Layer Aggregation Network (GELAN) and the Programmable Gradient Information (PGI) framework are two of YOLOv9's primary developments[51]. Compared to YOLOv8, the deep model's superior design enables it to cut both the number of specifications and computations by 49% and 43% [52]. 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[53]. Comprehensive tests show that, in comparison to other cutting-edge detectors, the YOLOv10 achieves cutting-edge performance while balancing efficiency[54].

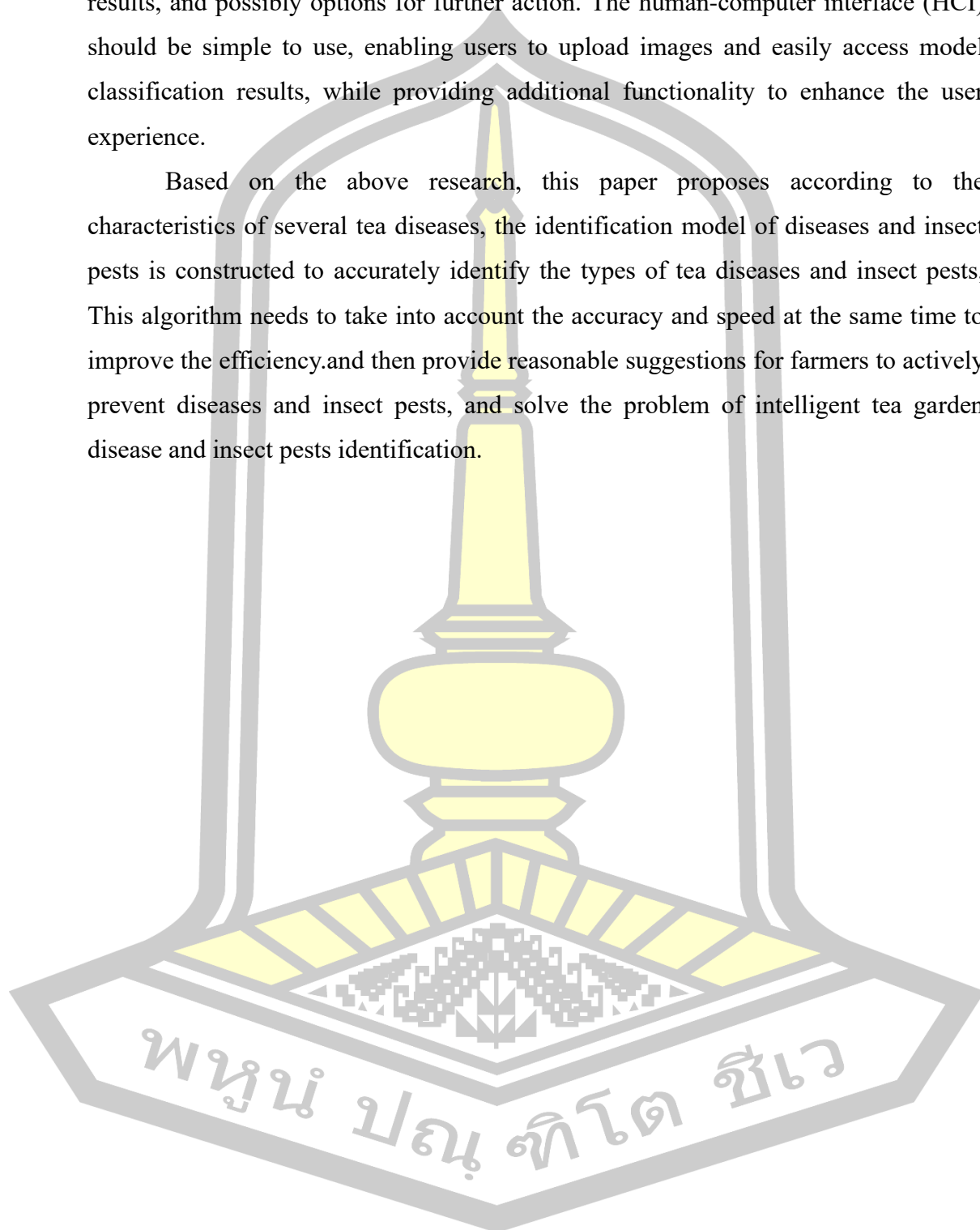
III. Summary

In conclusion, the research on status of classification,detection and identification of tea leaf pests and diseases in the literature cited in this paper has the following characteristics:

1. The speed and accuracy of traditional neural network models for detecting and recognizing tea leaf pests and diseases are not very high.
2. Deep learning models have achieved a certain improvement in the accuracy of tea leaf pests and diseases classification and recognition. However, there is still room for further improvement in terms of model parameters and performance.
3. We want to migrate the trained deep learning model to a real production environment and develop a graphical interface to provide users with an easy-to-use interactive interface.How to design an intuitive interface layout so that users can easily understand how to use the application. Typically, the interface should include

an area for uploading images, a button to start categorizing, an area for displaying results, and possibly options for further action. The human-computer interface (HCI) should be simple to use, enabling users to upload images and easily access model classification results, while providing additional functionality to enhance the user experience.

Based on the above research, this paper proposes according to the characteristics of several tea diseases, the identification model of diseases and insect pests is constructed to accurately identify the types of tea diseases and insect pests, This algorithm needs to take into account the accuracy and speed at the same time to improve the efficiency. and then provide reasonable suggestions for farmers to actively prevent diseases and insect pests, and solve the problem of intelligent tea garden disease and insect pests identification.



Chapter 3 Research Methodology

In this chapter, the basic methodology of the study is described, as well as common tea leaves diseases classification models and optimization algorithms, and then two common tea leaves diseases detection and identification models, as well as model improvement methods.

I. Research methods

(1) Literature research method: Searching for relevant literature materials, conducting in-depth study and research on technologies and theories related to tea disease detection and recognition techniques, deep learning models, algorithms, etc. Continuously improving the theoretical framework, and exploring research ideas and strategies.

(2) Experimental method: Collecting sample data, utilizing the Windows operating system, employing the PyTorch 1.7.0 framework, and conducting training using graphics cards. Writing code using the Python language. Through model training, calculating the accuracy and speed of tea disease classification, detection, and recognition.

(3) Comparative method: After conducting research and comparing the advantages and disadvantages of deep learning frameworks, convolutional neural networks, and object detection models, an object detection experimental platform suitable for this study was established. The various techniques used for training several target classification and detection models in this study were introduced, covering fundamental algorithm concepts, computational processes, specific operational methods, and improved algorithms. Through experimental comparisons, the foundational object detection model for this study was selected.

II. Classification of tea leaf pests and diseases based on deep learning techniques

This study first focuses on six categories of images, including common infections of tea leaves with five types of diseases and healthy leaves. The three improved deep convolutional neural network models, namely VGG-16, ResNet-50, and DenseNet-169, are employed. These models are loaded using the PyTorch learning framework. During the training process, the best-performing model parameters are saved. Through performance comparisons, a classification algorithm model with higher accuracy is identified. Additionally, this model can be utilized to

load the optimal model parameters for recognizing local images. As shown in Figure 1.

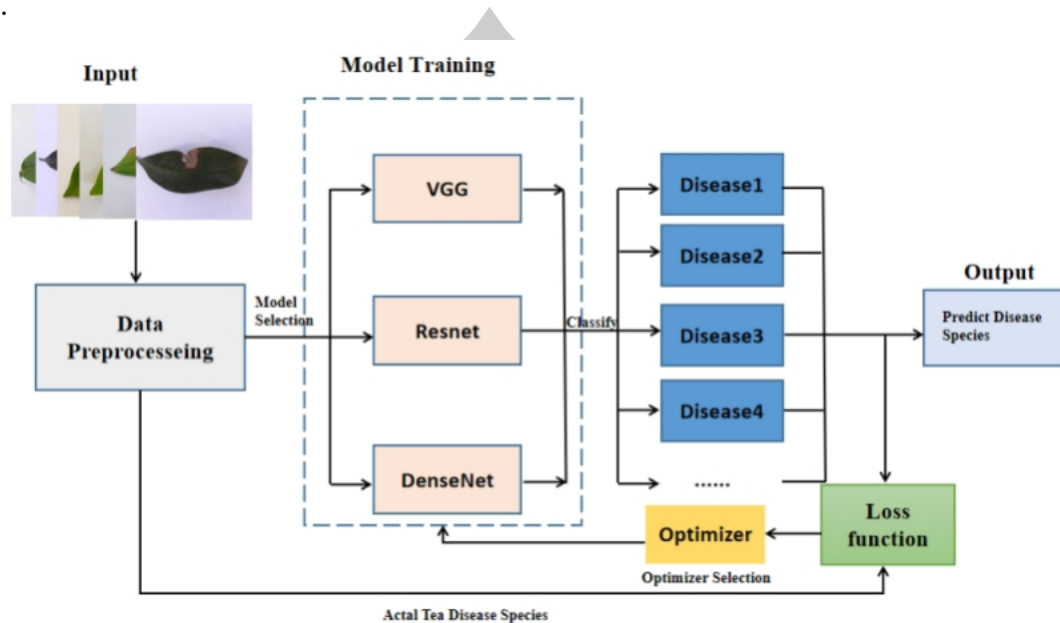


Figure 1 Proposed Training flow chart of tea leaf pests and diseases classification model

Tea disease image recognition refers to the classification of the input image into different disease categories, the input is a color image and the output is the classification label corresponding to the image, i.e., identifying the name of the disease as well as the specific location of the disease in the picture. The process of Deep Convolutional Neural Networks to handle the classification of tea disease images is as follows, Data preprocessing: the original image is converted into a format suitable for model input. The input to the CNN is to convert the image data into tensor (image_height, image_width, color_channels), which contains the image height, width and color information. Color_channels are (R,G,B) corresponding to the three RGB color channels. (color_channels are (R,G,B) corresponding to the three color channels of RGB. For example, the shape of our CNN input is (224, 224, 3) which is a color Image. Convolutional operations need to be performed In three dimensions, using 3D convolution, i.e. 3D convolution kernel, also called 3D filter.

The roles of each layer are summarized as follows: Input layer: used to input data to the training network; convolutional layer: uses convolutional kernel to extract image features; pooling layer: performs downsampling to represent the image features with a higher level of abstraction; tiling layer: one-dimensionalizes multi-dimensional

inputs and is often used in the transition from convolutional to fully-connected layer; fully-connected layer: acts as a feature extractor; output layer: outputs results, i.e., labels corresponding to the names of different diseases; and output layer: outputs results, i.e., labels corresponding to different diseases. “Output layer: outputs the results, i.e., the corresponding labels of different disease names. The output value of each neuron represents the probability or confidence level of the input sample under each category. The category with the highest confidence level is used as the predicted category of the sample on the model.

The data is processed by calculating the loss between the predictions of the trained model and the actual labels. The backpropagation algorithm is used to update the model parameters to reduce the loss function and improve the model performance. The model training process requires the adjustment of several parameters such as loss function (loss): the neural network uses a certain metric as a cue to find the optimal weighting parameters. The metrics used in the learning of the neural network are called loss function (loss function). It is the difference between the predicted value of the model output and the true value of the image, which is used to measure the accuracy of the model during training. Optimizer: determines how the model updates based on the data it sees and its own loss function. Enables efficient search of the parameter space. Deep Convolutional Neural Networks for recognizing tea leaf diseases involves several key steps such as data preparation, neural network construction, model training, hyperparameter tuning and model evaluation. By continuously optimizing these steps, a high-performance rice disease recognition model can be constructed.

2.1 The VGG16 model introduction

VGG is a fundamental CNN architecture comprised of convolutional layers and pooling layers. The name stands for the abbreviation of the Visual Geometry Group, the group to which the authors belong at the University of Oxford. In the case of the VGG16 model, it is particularly well-suited for classification and localization tasks. Its defining feature is the stacking of layers with learnable weights (either convolutional or fully connected) up to 16 layers, contributing to its depth (hence the name 'VGG16').

In VGG16, the convolutional layers with small 3x3 filters are applied in consecutive operations. The process involves repeating the pattern of 'stacking convolutional layers 2 to 4 times and then halving the size through pooling layers', culminating in the final output through fully connected layers. As both convolutional and fully connected layers have learnable weight coefficients, they are often referred to as weight layers.

Within the VGG16 architecture, there are 13 convolutional layers, 3 fully connected layers, and 5 pooling layers, making a total of $13 + 3 = 16$ layers. In the VGG16 convolutional neural network, the 13 convolutional layers and 5 pooling layers work together to extract features, while the final 3 fully connected layers are responsible for the classification task, as shown in Figure 2. The deeper the convolutional network, the more advanced features it can extract, resulting in better performance. However, traditional convolutional neural networks encounter issues such as network degradation, vanishing gradients, and gradient explosion as the depth of layers increases. This often leads to the paradox where the performance of deep networks is not as good as that of shallow networks.



Figure 2 VGG16 model architecture diagram

2.2 ResNet50 model introduction

ResNet50 is a deep convolutional neural network model used for tasks such as image classification and object detection. It consists of a network structure with 50 layers and employs residual connections to address optimization challenges in deep networks.

The model begins with convolutional operations on the input, followed by four residual blocks. Subsequently, a final fully connected layer is employed for classification purposes. The schematic representation of the network's structure is

illustrated below. ResNet50 comprises a total of 50 conv2d operations. The primary feature of ResNet50 is its utilization of skip connections, which involve adding the features from earlier layers directly to the outputs of subsequent layers. This approach helps retain and propagate more information, thereby enhancing the network's performance. Increasing depth is crucial for performance improvement. However, in deep learning, excessively increasing depth can often hinder learning and result in suboptimal performance. In ResNet, to tackle such issues, a 'shortcut structure' (also known as 'shortcut' or 'skip connection') is introduced. By incorporating this shortcut structure, performance can be progressively improved with increasing layer depth (within certain limits, of course) , as shown in Figure 3.

ResNet50 is a highly powerful deep learning model. Its depth and the residual learning approach enable it to learn more intricate features, thereby enhancing its accuracy. The global average pooling layer in ResNet50 reduces the number of model parameters, mitigating the risk of overfitting.

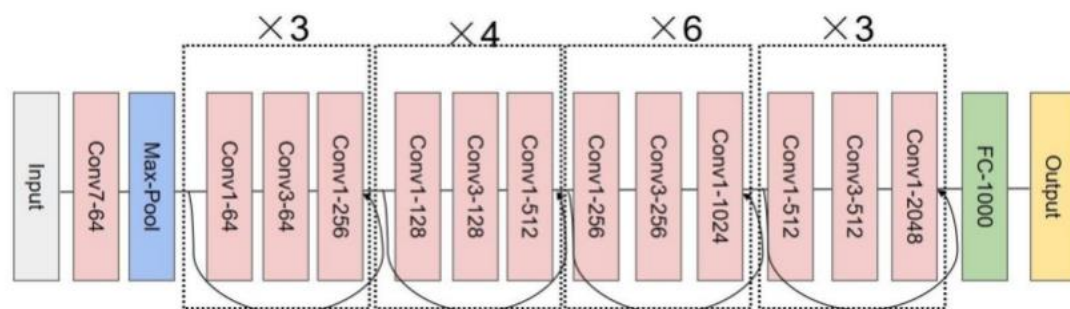


Figure 3 ResNet50 model architecture diagram

The shortcut structure spans (skips) across the convolutional layers of the input data, aggregating the input 'x' into the output. Within a sequence of three consecutive convolutional layers, the input 'x' is skip-connected to the output after the third layer. The key here is that through the shortcut structure, the original output 'F(x)' from the three layers of convolution becomes $F(x) + x$. By introducing this shortcut structure, the network can efficiently learn even with increased depth. This is because, with the shortcut structure, signals can be propagated through backpropagation without attenuation, as shown in Figure 4.

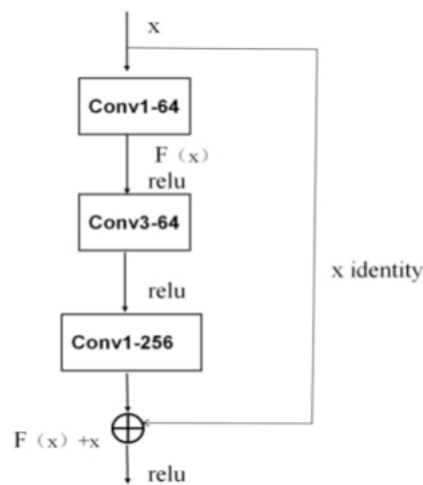


Figure 4 ResNet50 residual block structure

2.3 DenseNet model introduction

DenseNet (Densely Connected Convolutional Networks) model follows a similar underlying concept to ResNet, but it establishes densely connected connections between all preceding layers and the subsequent layer, hence its name. Another distinctive feature of DenseNet is the use of feature connections across channels to achieve feature reuse. The structure of the DenseNet network can be succinctly described as comprising multiple DenseNet Blocks. DenseNet is composed of several of these Dense Block structures, with its core idea being that each subsequent layer receives the outputs of all preceding layers as additional inputs.

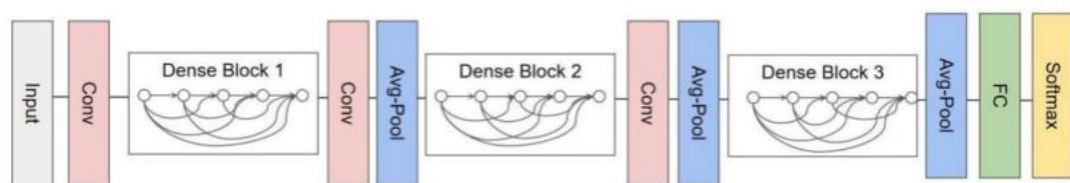


Figure 5 DenseNet model architecture diagram

As shown in Figure 5, The transition Layer is composed of BN, Conv (1×1), and 2×2 average-pooling. It serves as a module that connects different Dense Blocks. Its main purpose is to integrate features obtained from the preceding Dense Block, reduce the width and height dimensions of the previous Dense Block, achieving downsampling effects by halving the dimensions of the feature maps.

While evaluating the suggested job, accuracy is regarded as one of the most crucial criteria to take into consideration. The accuracy is measured as the ratio of successfully detected picture classes to the total number of image classes concerning the entire number of photos that were taken into consideration.

The role of the loss function in deep learning is to measure the error between the model's predictions and the true labels. Its value can be obtained through computation and assists in guiding the optimization algorithm for updating the model parameters.

III. Detection of tea leaf pests and diseases based on deep learning techniques

In response to the three common tea leaf disease images captured under natural conditions, due to the varying backgrounds in these images, this study employs a one-time output approach to detect target information, including category and location, through a detection model. We utilized improved deep learning object detection models such as YOLO , SSD and Faster R-CNN models, and optimized the model algorithms, aiming to identify the model that achieves the best performance in terms of accuracy and detection speed, as shown in Figure 6.

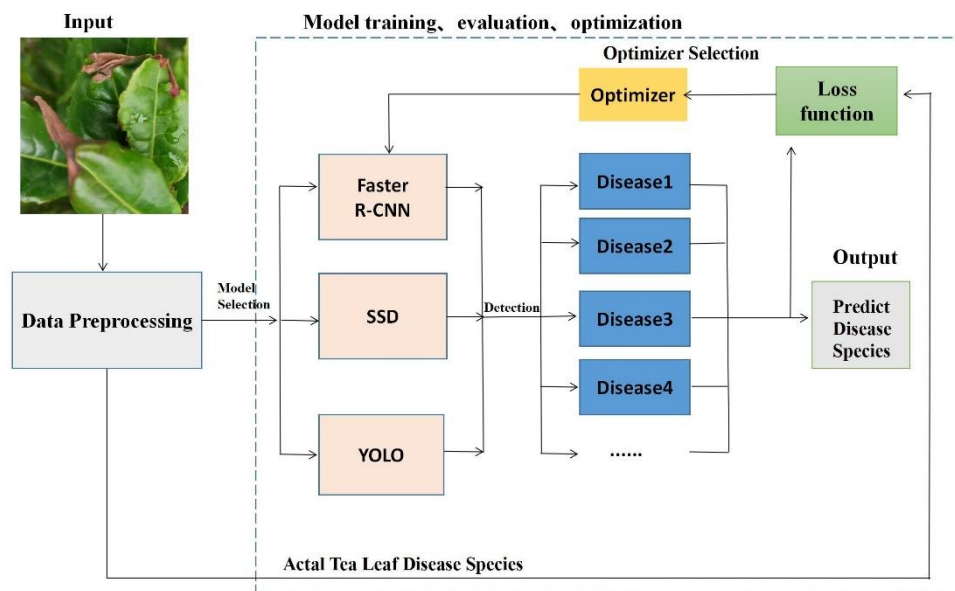


Figure 6 Proposed Training flow chart of tea leaf pests and diseases detection model

In deep object detection tasks, the loss functions typically need to consider both the accurate localization of objects and the correctness of their classification.

Here is an introduction to several common loss functions used in deep object detection:

1. Localization Loss:

For accurately localizing the positions of objects, the smooth L1 loss is commonly employed. This loss measures the difference between the predicted box coordinates and the true box coordinates. The Smooth L1 loss smooths out the loss when the differences are small, helping to prevent gradient explosions.

2. Classification Loss:

The classification loss in object detection measures whether an object is present within the predicted box and quantifies the disparity between the predicted class and the true class. Common classification loss functions include Cross-Entropy Loss and Focal Loss. Cross-Entropy Loss is suitable for general classification tasks, while Focal Loss is specifically designed to address class imbalance by reducing the weight of easily classified samples and focusing more on hard-to-classify samples.

3. IOU Loss:

The Intersection over Union (IOU) loss directly quantifies the degree of overlap between the predicted box and the true box. During training, a specific loss function can be designed that incorporates the difference between the IOU of the predicted box and the IOU of the true box. This contributes to improving localization accuracy.

Each of these loss functions serves a specific purpose, encompassing both accurate localization and classification precision. In practical applications, the choice of an appropriate loss function depends on the task characteristics and the nature of the data, enabling effective guidance of model training for better performance in object detection tasks.

3.1 The Faster R-CNN model introduction

Ross B. Girshick proposed the new Faster RCNN in 2016, in terms of structure, Faster RCNN has integrated feature extraction (feature extraction), proposal extraction, bounding box regression (rect refinement), and classification are all

integrated in one network, which makes the comprehensive performance improved greatly, especially in the detection speed.

As shown in Figure 7 Faster R-CNN can actually be divided into 4 main elements:

Conv layers. as a CNN network target detection method, Faster RCNN first uses a set of basic conv+relu+pooling layers to extract the image's feature maps, which are shared for the subsequent RPN layer and fully connected layer.

Region Proposal Networks: RPN networks are used to generate region proposals. this layer determines whether the anchors are positive or negative by softmax, and then uses bounding box regression to correct the anchors to obtain accurate proposals. **Roi Pooling.**

Roi Pooling: This layer collects the input feature maps and proposals, synthesizes the information and extracts the proposal feature maps, and sends them to the subsequent fully-connected layer to determine the target category.

Classification: The proposal feature maps are utilized to compute the category of the proposals, and at the same time bounding box regression is used again to obtain the final precise location of the detection box.

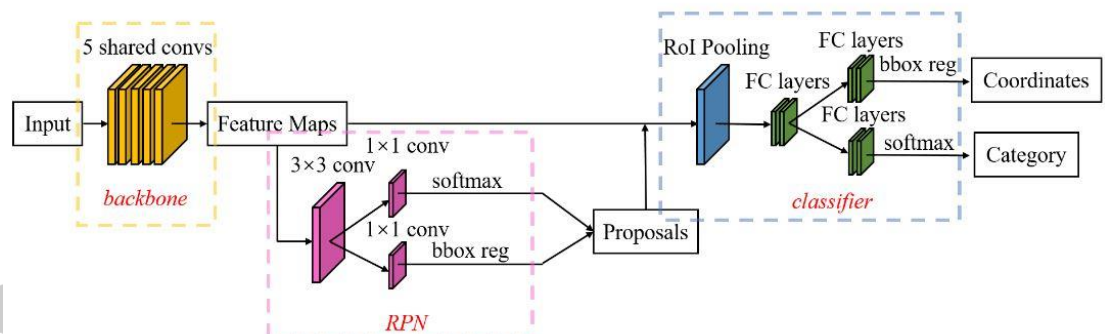


Figure 7 Architectural sketch of Faster R-cnn

3.2 The SSD model introduction

The full name of SSD is "Single Shot MultiBox Detector." It is a deep learning model used for object detection. SSD is capable of predicting the categories and locations of multiple objects in an image in a single forward pass, hence its name "Single Shot MultiBox Detector." This model demonstrates excellent performance in terms of real-time capabilities and accuracy and is commonly employed for object detection tasks in the field of computer vision.

The SSD model adds several feature layers to the end of a base network, which predict the offsets to default boxes of different scales and aspect ratios and their associated confidences, as shown in Figure 8.

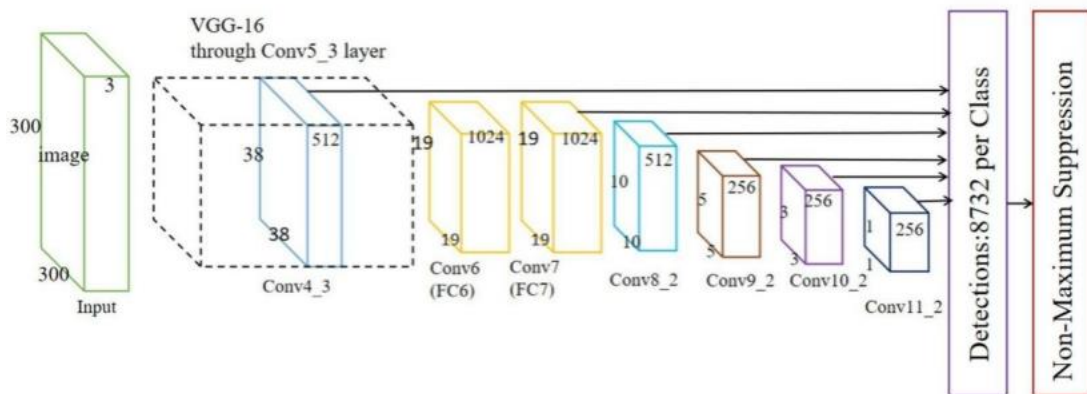


Figure 8 SSD model architecture diagram

The main characteristics of SSD are as follows: 1. Utilization of multiscale feature maps, 2. Employment of convolutions for detection, 3. Implementation of prior bounding boxes.

3.3 The YOLO model introduction

Real-time target detection has been the focus of research in computer vision, aiming to accurately predict the class and location of objects in an image with low latency. Among them, YOLO[14] is becoming increasingly popular due to its balance between performance and efficiency. It was first proposed in 2015 by Joseph Redmon et al. and has been subsequently improved several times. The core idea of YOLO is to divide the entire image into a fixed number of grids (grid cells) and then predict multiple bounding boxes and category probabilities simultaneously within each grid. The principle of YOLO is mainly as follows: it performs one-time inference to output detected target information, including class and location. A region is condensed into a multi-dimensional point (1 x 1 x N dimensions), and each 1x1 point generates 3 anchor boxes (or other values). By considering all anchor boxes (tens of thousands), they are compared to the ground truth labels to calculate the loss. The historical significance of YOLO is that it creatively combines recognition and localization into

one. Due to its extremely fast detection speed, it has gained significant popularity in practical applications.

Based on the compiled dataset, I trained the object detection model using the cutting-edge YOLOv10s, YOLOv9s, YOLOv8s and YOLOv5s object identification techniques to achieve real-time identification of the target objects.

YOLOv5 algorithm is one of the most widely used target detection algorithms, which is based on deep learning technology, adding modules such as feature pyramid network and SPP structure on the basis of convolutional neural network, so as to realize the balance of high accuracy and fast detection speed. Based on YOLOv5s-5.3, we present an upgraded version of the YOLOv5s disease and pest detection model in this study. The image inference speed of the Yolov5s model reaches 455 FPS on a subset of researchers' devices, which is widely used by a large number of scholars with this advantage[55]. The Yolov5 algorithm utilizes deep learning techniques to swiftly and accurately identify objects in images, while annotating their positions and class information, as shown in Figure 9.

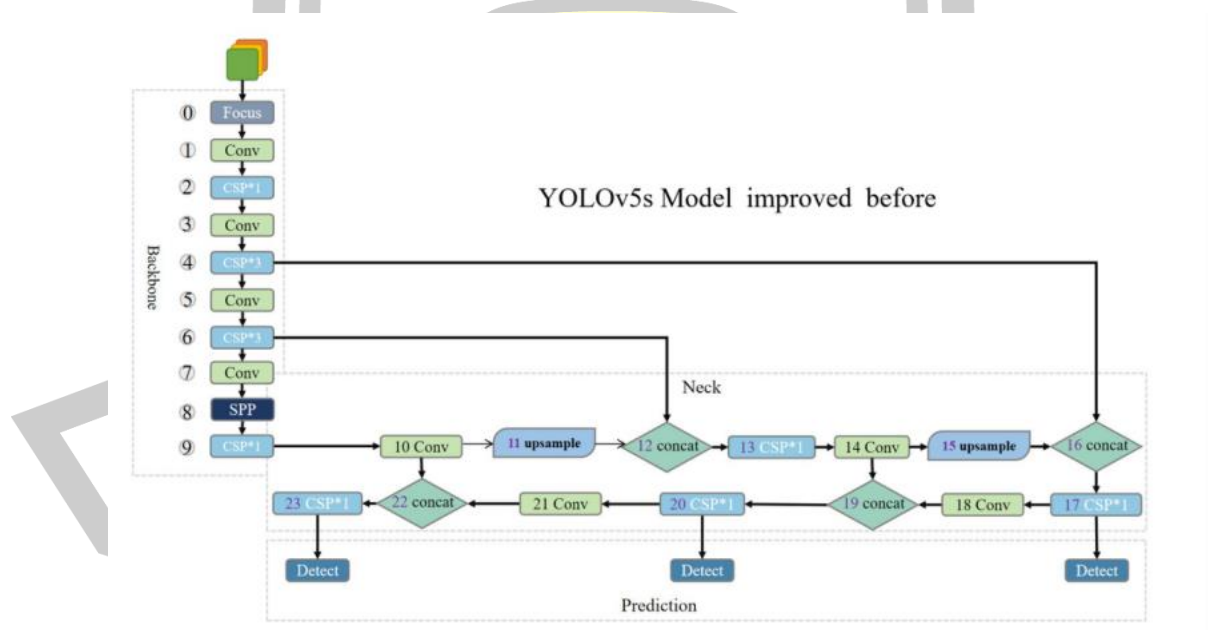


Figure 9 The network structure of Yolov5s-5.3 model

As shown in Figure 9, The Yolov5 Version 5.3 network structure model includes three parts: Backbone function: feature extraction. Neck function: mix and combine features in one wave and transfer these features to the prediction layer. Head function:

make final prediction output. The YOLOv5 network structure consists of Input, Backbone, Neck, and Prediction. The Input part of YOLOv5 is the input end of the network, using Mosaic data augmentation method to randomly crop the input data and then concatenate it. Backbone is the network part of YOLOv5 that extracts features, and the feature extraction ability directly affects the overall network performance. In the feature extraction stage, YOLOv5 uses the CSPNet (Cross Stage Partial Network) structure, which divides the input feature map into two parts. One part is processed through a series of convolutional layers, and the other part is directly down sampled. Finally, these two parts feature maps are fused. This design enables the network to have stronger non-linear expression ability, which can better handle complex backgrounds and diverse objects in object detection tasks, as shown in Figure 10.

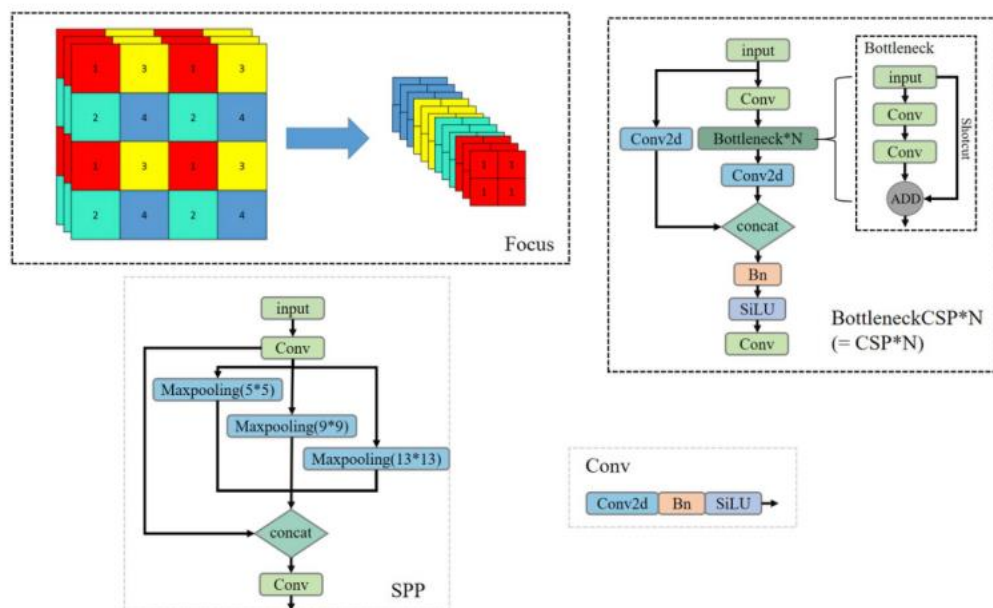


Figure 10 The component networks in the original YOLOv5s

The Conv module is a commonly used foundational module in convolutional neural networks, which mainly consists of convolutional layers, BN layers, and activation functions. Convolutional layers are one of the most fundamental layers in convolutional neural networks, used to extract local spatial information from input features. The BN layer is a normalization layer added after the convolutional layer to normalize the distribution of eigenvalues in neural networks. Activation function is a non-linear function used to introduce non-linear transformation capability into neural networks. Common activation functions include sigmoid, ReLU, LeakyReLU, ELU,

etc. The main body of the Head consists of three Detect detectors, which use grid-based anchors to perform object detection on feature maps of different scales.

In order to further improve the accuracy of YOLOv5 series models, the algorithm needs to be improved. As shown in Figure 11, the model's improvements involve changing from CSP to CSPK, enhancing feature map convergence, preventing feature loss, adding BN layer and LeakyReLU activation function to improve the stability and generalization performance of the model. This is to enable the model to learn more features. And by adding Omni Dimensional Dynamic Convolution (ODConv) and Convolutional Block Attention Module (CBAM) to the model's head to further enhance feature adaptability and improve the performance of focusing on essential features.

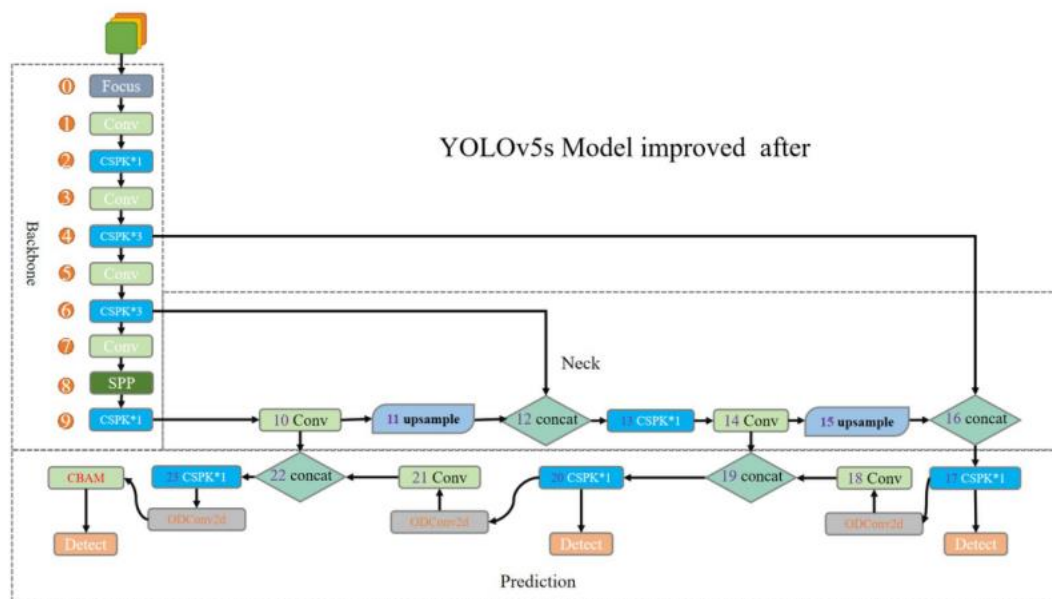


Figure 11 The network structure of the improved YOLOv5s

In neural network architectures, CSP (Cross Stage Partial) is a connection approach that is mostly used to improve the information transmission and interaction of features across several stages or scales. The term "conv2d" describes a convolution procedure that is frequently used to process picture data when it is applied to two-dimensional data. It takes in as input a three-dimensional tensor, of which the first and second dimensions stand for the image's height and breadth, and the third dimension for the image's channel count. CSP plays a crucial role in boosting the network's

representational capacity through improved feature extraction and representation, which leads to improved performance in a variety of computer vision applications.

The CSPK structure can enhance feature propagation more, aiding in better information transmission across different layers of the network, thereby improving performance, as shown in Figure 12.

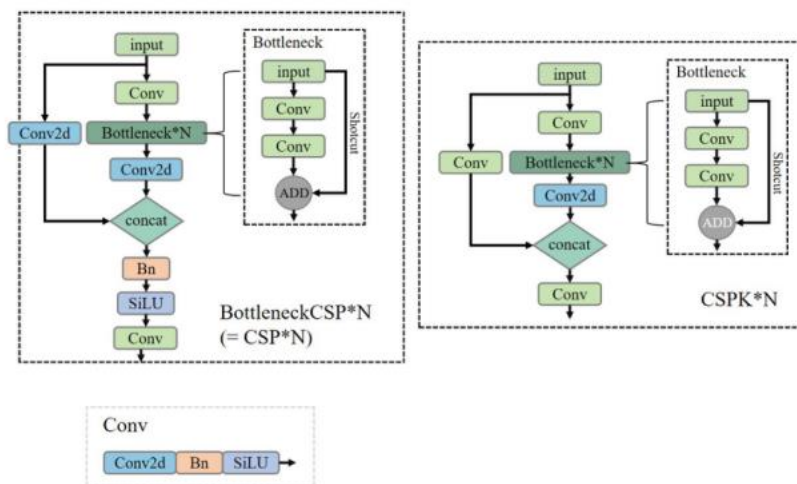


Figure 12 Comparison of CSP and CSPK Structure

Adding ODConv2d and CBAM to the model's head to further enhance feature adaptability and improve the performance of focusing on essential features. Dynamic convolution refers to the concept where the weights of the convolutional kernel are not fixed during the convolution process. Instead, they can be dynamically adjusted based on the input data. This adaptability of the convolutional kernel allows it to better capture the features present in the input data, thereby enhancing the performance of convolutional networks. Attention mechanisms are crucial neural network modules that enable networks to focus more effectively on relevant information when processing sequential data. By assigning different weights to different parts of the input data based on their significance, attention mechanisms ensure that the network pays greater attention to important information when transmitting and processing data.

As illustrated, the channel sub-module utilizes both max-pooling outputs and average-pooling outputs with a shared network; the spatial sub-module utilizes similar

two outputs that are pooled along the channel axis and forward them to a convolution layer. ODCnv utilizes a novel multidimensional attention mechanism and parallel strategy to learn the attention of convolutional kernels along the four dimensions of kernel space in any convolutional layer. These four different types of attention are: the number of input channels for the convolution kernel, the receptive field of the convolution kernel itself, the number of output channels for the convolution kernel, and the number of convolution kernels. As an alternative to conventional convolution, ODCnv can be inserted into many CNN architectures. That is to say, it is a plug and play module. As a result, while both dynamic convolution and attention mechanisms contribute to improving the performance of neural networks, they serve different purposes. Dynamic convolution enhances the adaptability of features, while attention mechanisms enhance the network's ability to focus on the specific information required for the current task.

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.

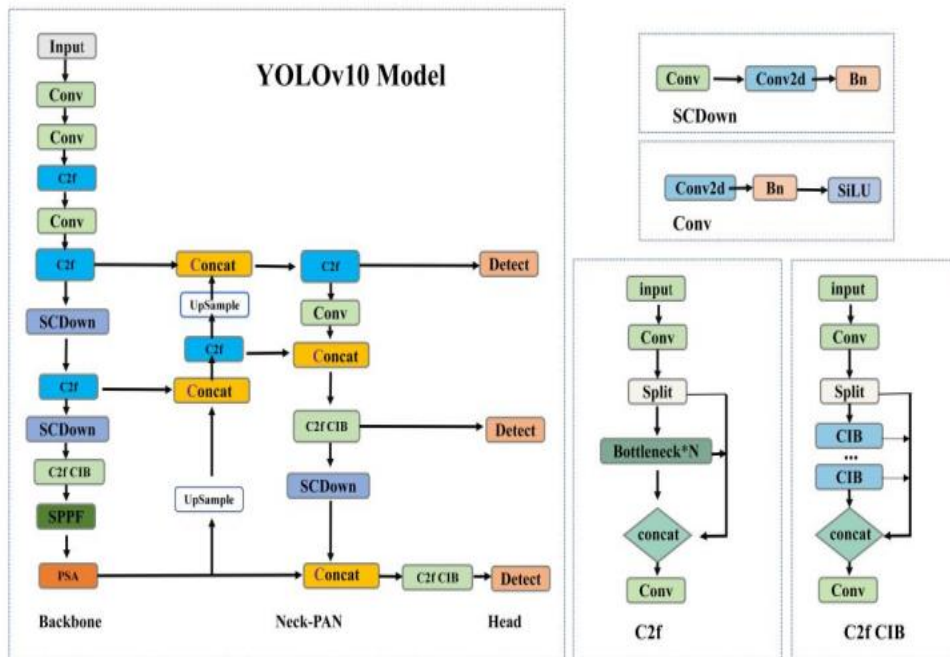


Figure 13 The Yolov10 model network structure

In Figure 13, 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.

IV. Approach to data acquisition and preparation

The dataset used for the classification of tea leaf diseases is derived from the publicly available tea disease dataset from the Kaggle platform, which is the largest data crowdsourcing platform on a global scale. The dataset can be downloaded at: <https://www.kaggle.com/datasets/mamun009/tea-leaf-disease-clasification>.

The dataset contains 'algal_spot', 'brown_blight', 'gray_blight', 'healthy', 'helopeltis', 'red_spot' five common tea diseases and healthy tea. Corresponding to the named categorized labels, it was observed that the number of each category was balanced and all of them were more than 1,000. The image samples of leaf diseases of tea leaves are shown in Figure 14.

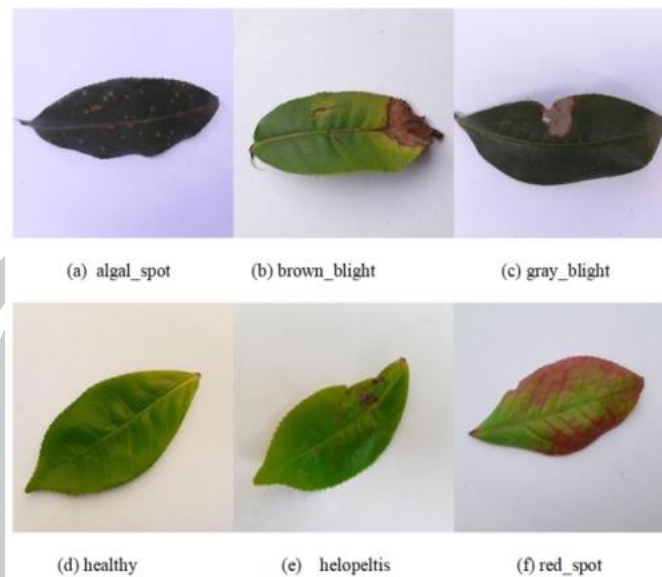


Figure 14 The proposed tea diseases and healthy leaves

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 15. Photographs of tea diseases and insect pests were taken at the Xiang tea farm in Changsha city, Hunan Province, China.



Figure 15 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 16.

Supervised machine learning is a machine learning technique in which neural networks learn to predict or classify data based on labeled datasets. Here, we input features as well as target variables. Neural network learning makes predictions based on the cost or error generated by the difference between the predicted goal and the actual goal in a process called back propagation. 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 [56, 57]. As shown in Figure 17, 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.

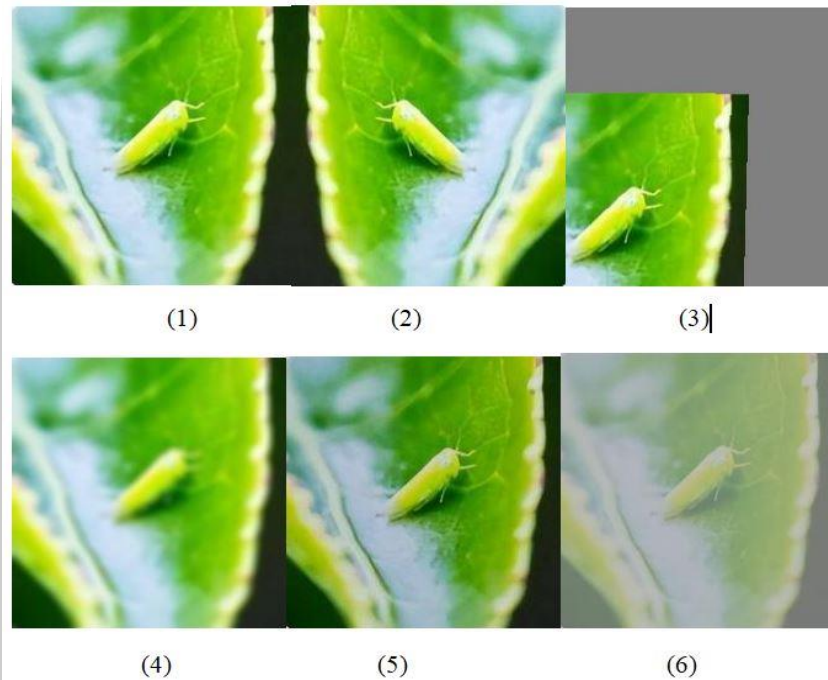


Figure 16 The representative samples of image data enhancement
 (1) original (2) flipping (3) random affine (4) gaussian blur (5) brightness (6) contrast

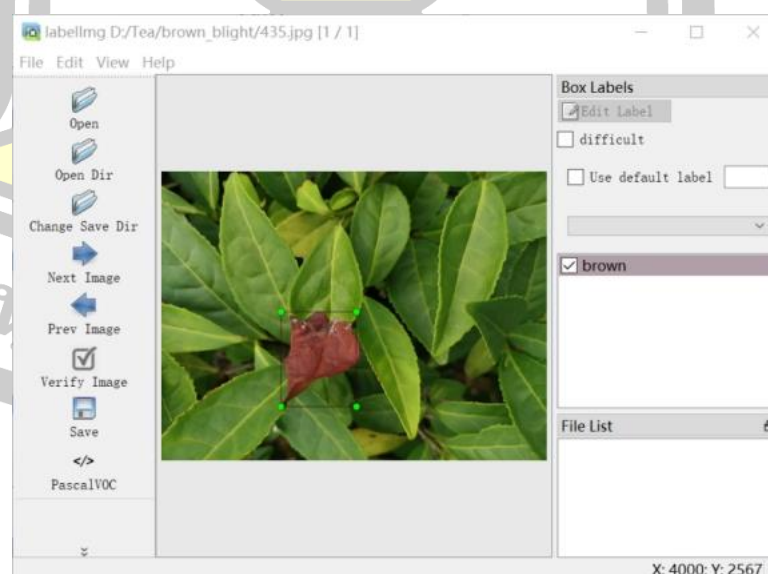


Figure 17 Image labeling fabrication

V. Evaluation metrics

1. Confusion matrix

In machine learning and deep learning, the prediction results of classification tasks are divided into the following four types, which are referred to as a confusion matrix: The confusion matrix is a method to visualize and summarize the performance of classification problems[58].

True Positive (TP)[59]: The predicted value is a positive case, and the label value is also a positive case, indicating that the prediction is correct. False Negative (FN): The predicted value is a negative case, while the label value is a positive case, indicating an incorrect prediction. False Positive (FP): The prediction is a positive example, but the label value is a negative example, indicating an incorrect prediction. True Negative (TN): The predicted value is a negative case, and the label value is also a negative case, indicating that the prediction is correct.

2. Precision

Precision(P) refers to precision, and its definition is as follows:

$$\text{Precision} = \frac{TP}{FP+TP} \times 100\% \quad (1)$$

According to the definition, the denominator of accuracy is the sum of TP and FP, where TP is the number of positive cases predicted, and the actual value is also positive; FP is the number of positive cases predicted and the actual number of negative cases. Accuracy is related to the predicted positive cases and the actual positive and negative cases. The higher the accuracy, the smaller the FP, and the fewer the number of other categories predicted as this category, which can be understood as the higher purity of the predicted positive examples. The higher the accuracy, the fewer false positives.

3. Recall

Recall(R) refers to the recall rate, and the definition of Recall is as follows:

$$\text{Recall} = \frac{TP}{FN + TP} \times 100\% \quad (2)$$

According to this definition, the denominator of Recall is the sum of TP and FN, where TP is the number of positive cases predicted, and the actual value is also positive; FN is the number of negative cases predicted, but the actual value is positive.

Through analysis of the formula, it can be seen that Recall involves both predicted positive and negative cases, as well as actual positive cases. When Recall is higher, FN is smaller, and the number of positive cases predicted as negative is also smaller, which can be understood as identifying more positive cases. A higher Recall indicates fewer missed detections.

4. F1-Score

The F1-Score is a composite statistic that balances precision and recall, calculated using the harmonic mean.

$$F1 = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \times 100\% \quad (3)$$

5. P-R curve

The P-R curve is a curve enclosed in Precision and Recall coordinates, respectively. The lines of different colors represent the PR curves of different categories, while the thick blue lines represent the average PR curves of all categories.

The area enclosed by the P-R curve and the axes can serve as a reference for evaluating the prediction results of a model. If the P-R curve of one model completely encloses the P-R curve of another model, it can be said that the prediction results of the former model are definitely better than the latter model.

6. AP

AP (Average Precision): Despite its name, the calculation of AP does not involve taking the average of precision values. Instead, it computes the area under the PR curve (Precision-Recall curve) for each class, which can be calculated through integration. If a model has a higher AP, meaning the area enclosed by the PR curve and the axes is larger, it indicates that the overall Precision and Recall are relatively higher.

$$AP = \int_0^1 P(R) dR \quad (4)$$

7. mAP

mAP (mean of Average Precision): calculate the average AP value for all categories. AP can reflect the accuracy of predictions for each category, and mAP is the average of the APs for all categories, used to reflect the accuracy of the entire model. The larger the mAP, the larger the area enclosed by the PR curve and

coordinate axis. We often talk about the accuracy of a certain object detection algorithm, which is generally referred to as mAP.

$$mAP = \frac{\sum_{i=1}^N AP_i}{N} \quad (5)$$

8. mAP50

In the YOLO model, you will see mAP50. This form of expression represents what the value of mAP is when the IOU (Intersection over Union) threshold is 0.5. When the IOU between the predicted box and the annotated box is greater than 0.5, it is considered that the object prediction is correct, and then mAP is calculated based on this premise. Generally speaking, map50. It is one of the indicators for evaluating the YOLO model.

8.mAP[0.5:0.95]

In the YOLO model, there is also a form of expression called mAP [0.5:0.95], which represents the mean Average Precision at multiple Intersection over Union (IoU) thresholds. Within the range of [0.5, 0.95], with a step size of 0.05, 10 IoU thresholds are taken, and the Average Precision at each of these 10 IoU thresholds is calculated. The mean of these values is then taken to obtain mAP [0.5:0.95].

VI. Tea diseases and pests detection system

In software development, graphical user interface(GUI), is essential components[60].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[61].

The core operating mechanism of PyQt5 revolves around an event-driven model. Each UI component is an object that responds to user actions (e.g., mouse clicks, key inputs, etc.), which are translated into internal events that in turn trigger the corresponding event handler functions. In addition, PyQt5 adopts Qt's signal and slot mechanism, which greatly simplifies the process of inter-component communication.Signal is an important concept in PyQt5, where an object can emit a signal when the state of an object changes or a specific event occurs. A slot is a

function that can be triggered by a signal. It can be any callable object, such as a normal function, a class method, or a lambda expression. In PyQt5, the connection between a signal and a slot can be realized by the `.connect()` method without explicit calls. The “front-end” in PyQt5 usually refers to the user interface part, while the “back-end” is the business logic or data processing part. Through the signaling and slotting mechanism, user interactions in the front-end can be easily passed to the back-end for processing. PyQt5 has become an important tool for Python developers to build beautiful and efficient desktop applications thanks to its flexible signaling and slotting mechanisms, highly modular architecture, and excellent cross-platform capabilities.

In PyQt5, a typical application life cycle is roughly divided into the following steps:

1. Application instantiation and event loop initiation. First, we need to create a `QApplication` instance, which is the basis of all Qt applications and is responsible for managing global tasks such as windows, menus, event loops, and so on.

2. Interface construction. User interfaces can be built in two ways: either by creating and organizing controls directly in Python code, or by designing the interface using Qt Designer and then converting the design files to Python code.

3. Event processing. Qt applications rely on an event-driven programming model. User actions (e.g., mouse clicks, keystrokes, window resizing, etc.) generate corresponding events that are placed in an event queue to be processed. When an event occurs, Qt converts the event to a `QEvent` object and distributes it to the control. The control handles these events by overriding `QObject.event()` or a specific type of event handler (e.g. `mousePressEvent()`, `keyPressEvent()`, etc.).

4. Signals and Slots. In PyQt5, the signals and slots mechanism is a key means of enabling communication between components. When an object's state changes (e.g. a button is clicked), it emits a signal. Other objects can connect to this signal and specify a slot function in response to the signal being emitted.

5. Event Loops and Program Exits. Finally, the `QApplication.exec_()` method is called to start the event loop, and the program begins listening for and processing events. When the event queue is empty or a `QCoreApplication.quit()` call is encountered, the event loop ends and the application closes.

Through the above mechanisms, PyQt5 applications can respond to user operations in real time, dynamically update the interface state, and ensure effective collaboration between front-end and back-end logic through signaling and slotting mechanisms to build feature-rich and responsive desktop applications.

So, we need to have the PyQt5 library installed. PyQt5 is a layer of interface between Python and the Qt library, through which we can access all the features of 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[62].

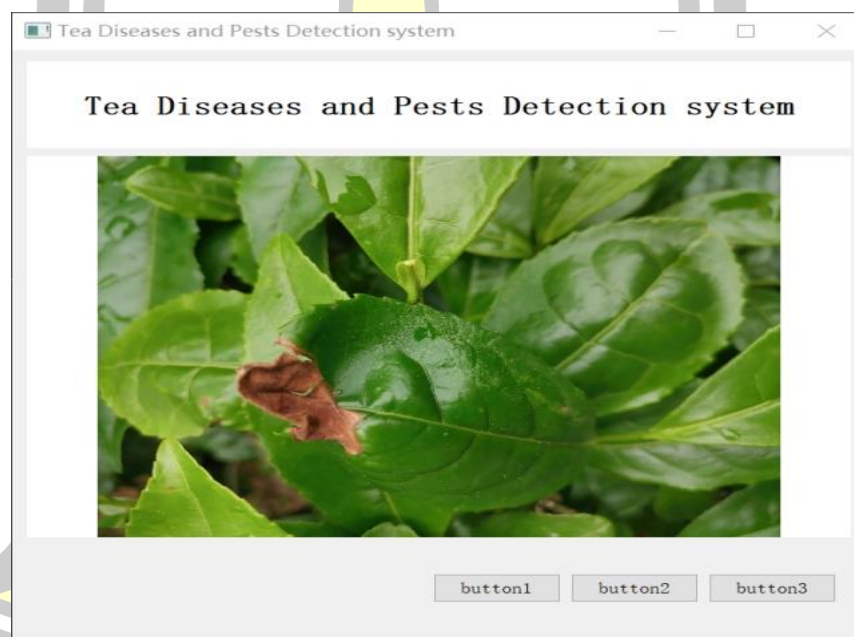


Figure 18 PyQt5 base applications interface

To build a desktop application, the first interface container should be a QMainWindow instance. As shown in Figure 18, we have used PyQt5 to design a basic interface with buttons, image display window, text display box.

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 at the same time, the picture or video detection results can be saved. The aim is to

provide 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 for real-time observation and analysis of results. 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 application of intelligent detection technology for tea diseases.

VII. 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 2 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 2 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
Memory	90GB

The hyperparameters of automated diseases and pests detection in tea leaves with object detection models are shown in the Table 3. 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, images 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 3 Training hyperparameters

Specifications	Merit
Batch size	8, 16
Epochs	40, 150
Image size	640 × 640
Momentum	0.937
Weight decay	0.0005
Optimizer	SGD(Stochastic Gradient Descent)

VIII. Tea disease classification and recognition process

The program process mainly includes the processes of image data preprocessing, model selection, model training and testing, and metrics evaluation. To use deep learning for image classification, the following steps are usually required:

Data preprocessing: This includes image size uniformity, color balance adjustment, etc. to ensure data consistency and comparability. At the same time, enhancement of the data, such as rotation, scaling, cropping, etc., can also be considered to improve the generalization ability of the model.
Model construction: Define the network structure of the model, including convolutional layers, pooling layers, fully connected layers, etc.

Model Training: VGG16, ResNet50 and DenseNet deep learning classification models are selected for training, respectively. Train the model using the training data and corresponding labels, and extract 32 samples from the training samples each time to participate in the training.

The model is trained with training data and corresponding labels, and 32 samples are extracted from the training samples each time to participate in the training, and the model parameters are constantly updated to minimize the classification error.

During the model training process, careful tuning work is performed on the parameters of the model. This includes the selection and tuning of hyperparameters such as choosing the dynamic learning rate, batch size of 16, and number of iterations of 40. Through continuous experimentation and optimization, the best combination of parameters can be found to improve the model's performance performance.

Model Evaluation: Evaluate the performance of the model on the test data, using metrics such as accuracy, loss function size, etc. The convolutional layer is responsible for extracting important features. After the model training is completed, adequate validation and testing efforts are required. This includes evaluating the model using independent validation and test sets to verify the generalization ability and reliability of the model.

The convolutional layer is responsible for extracting the important features, and finally, in order to have a complete image classification model, we only need a number of fully connected output nodes to decide the correct class of the image based on their weights! Repeatedly observe the experimental results and normalize the data separately, with a standard normal-to-oo distribution (Gaussian) to make the model easier to converge. Follow the 8:1: 1 ratio, the training set, validation set and test set are divided. After several tuning parameters and pre-training, the final model classification accuracy each reaches the highest.

IX. Tea disease target detection training process

1. Data set preparation and training

By collecting relevant images about tea pests and diseases in real scenarios and labeling each image using Labeling labeling tool in 3 detection categories, namely 'brown_blight ,gray_blight and tea leafhopper'. The dataset is divided into training set, validation set and test set in the ratio of 7:2:1.

2. Model training

After preparing the dataset, place the image data in the project directory in the following format. Create a new datasets directory in the project directory and also divide the detected images into training set and validation set into Data directory.

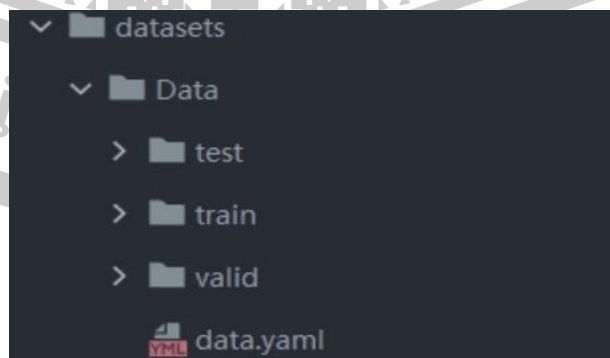


Figure 19 Data storage location

At the same time, we need to create a new data.yaml file, which is used to store the path of the training data and the categories that the model needs to detect. YOLOv10/9/8/5 will read the information in this file for model training and validation.

After data preparation, model training is performed by calling the train.py file, the epochs parameter is used to adjust the number of rounds of training, the batch parameter is used to adjust the batch size of training (adjusted according to the size of the memory, with a minimum of 1), and the optimizer set by the optimizer is SGD.

YOLOv10/9/8/5 The training settings for the model include various hyperparameters and configurations used during training. These settings affect the performance, speed, and accuracy of the model. Key training settings include batch size, learning rate, momentum, and weight decay. In addition, the choice of optimizer, loss function, and composition of the training dataset affects the training process. Careful tuning and experimentation with these settings is essential to optimize performance.

Table 4 Commonly used model training parameters and descriptions

Parameter name	Value	Instructions
Epochs	150	Total number of training rounds. Each epoch represents one complete training of the entire dataset.
Patience	120	The number of epochs required to stop training early when the validation metrics do not improve.
Batch	16	Batch size of images per training
Imgsz	640	The target image size used for training. All images are resized to this size before they are fed into the model.
Optimizer	auto	Select an optimizer for the training model. Options include SGD, Adam, AdamW, NAdam, RAdam, RMSProp and more.
lr0	0.01	initial learning rate (i.e., SGD=1E-2, Adam=1E-3) . Adjusting this value is crucial for the optimization process and affects the speed of updating the model weights.

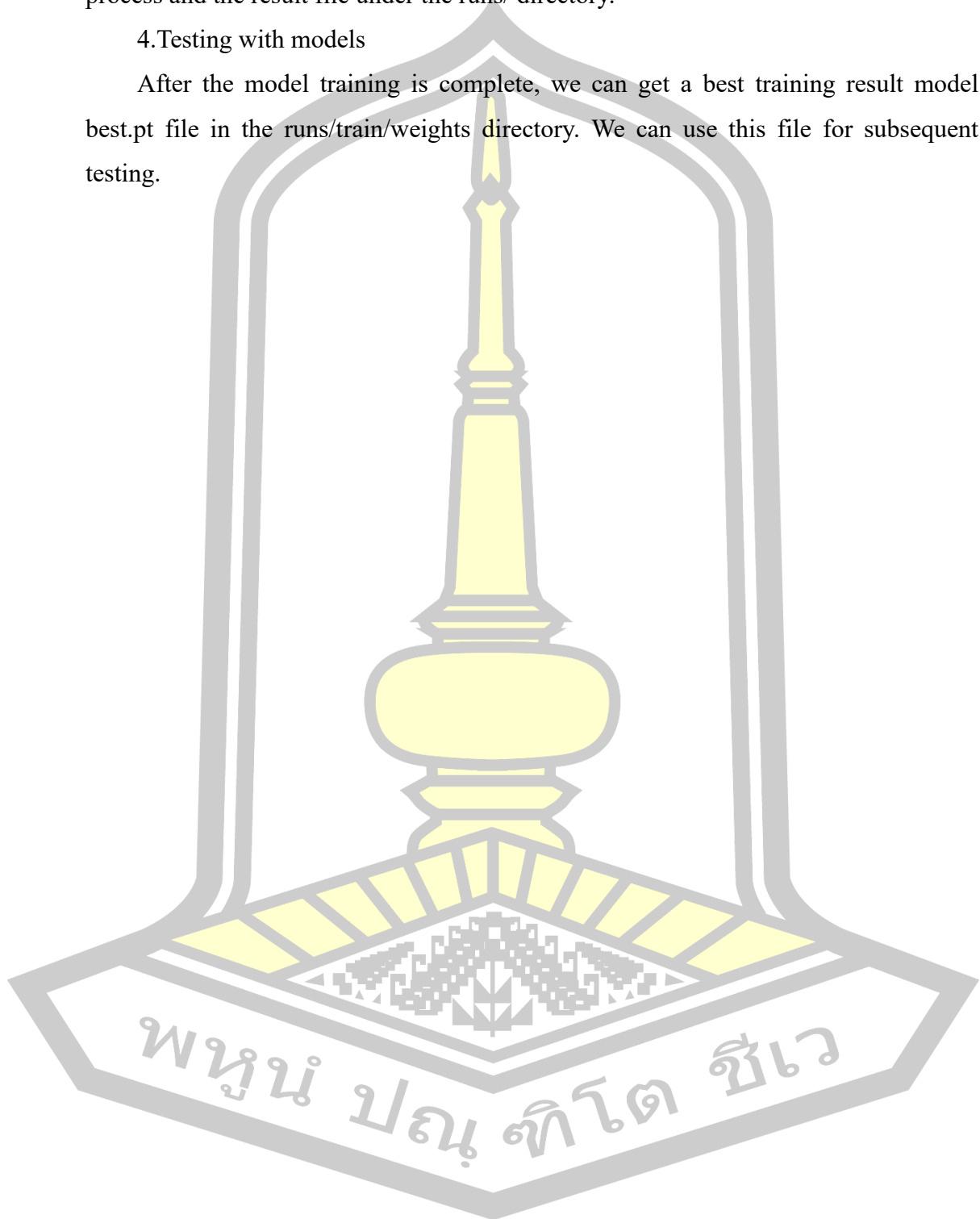
3. Evaluation of training results

In deep learning, we usually use the curve of the descending loss function to observe the training of the model. YOLO contains three main aspects of loss during training: localization loss (box_loss), classification loss (cls_loss), and dynamic

feature loss (dfl_loss), and at the end of the training process, you can find the training process and the result file under the runs/ directory.

4. Testing with models

After the model training is complete, we can get a best training result model best.pt file in the runs/train/weights directory. We can use this file for subsequent testing.



Chapter 4 Results

I. Results of classification of tea leaf pests and diseases

The dataset contains 'algal_spot', 'brown_blight', 'gray_blight', 'healthy', 'helopeltis', 'red_spot' five common tea diseases and healthy tea were divided into training, validation, and testing sets in a 7:2:1 ratio. The total number of data for each category is close to 1,000 sheets, and the dataset is relatively well-balanced. The architectural models used are VGG16, ResNet50 and DenseNet169 with the same batch size i.e. 32, for each batch size value each experiment will go through the same methodology i.e. using 40 epochs, because the loss curve has been relatively stable, Adam optimizer, same cross-entropy loss function and using the same dynamic learning rate, the best validation accuracy and loss function values are shown in Figure 20 and Figure 21.

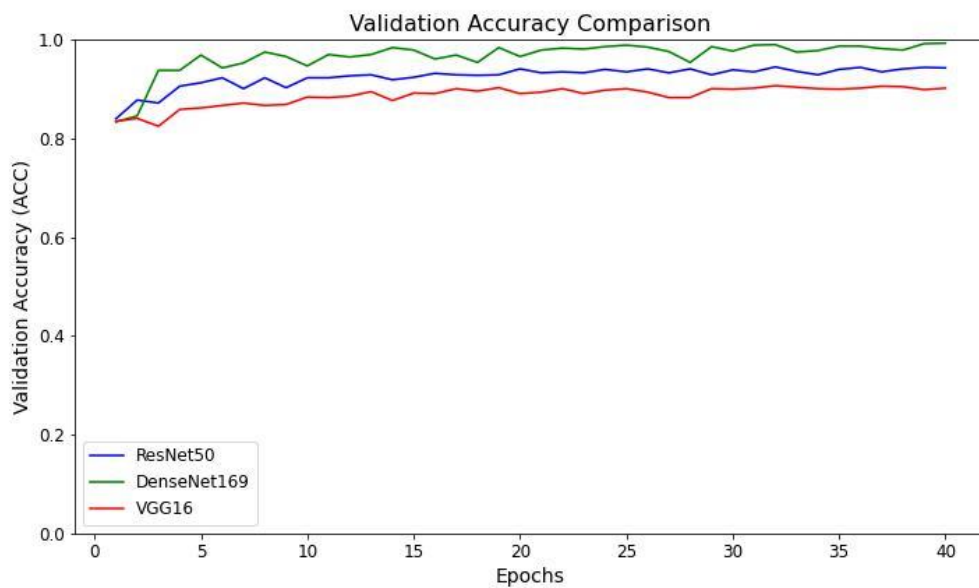


Figure 20 Comparison of the accuracy of three model validation

Through the analysis of the results, with the same learning rate, optimizer and each parameter, the test accuracy and convergence speed are the highest in the DenseNet169 model training, and its loss function is also significantly lower than the other two models.

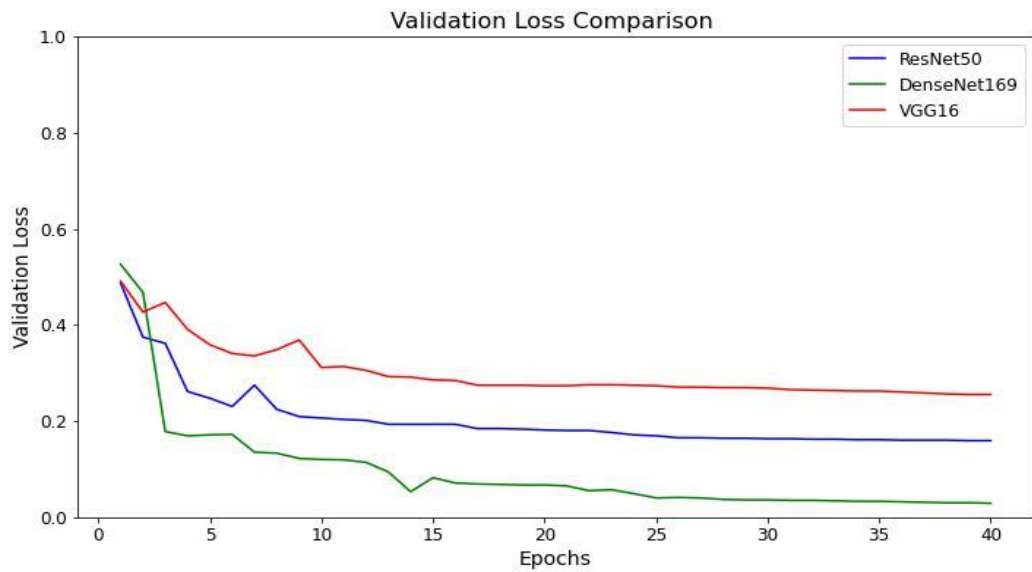


Figure 21 Comparison of loss function values for three model validation sets

Based on the results of the highest accuracy obtained during the validating of the model DenseNet169, the next step was to perform data testing to find out the accuracy generated during the testing process. The confusion matrix for data testing is shown in Figure 22.

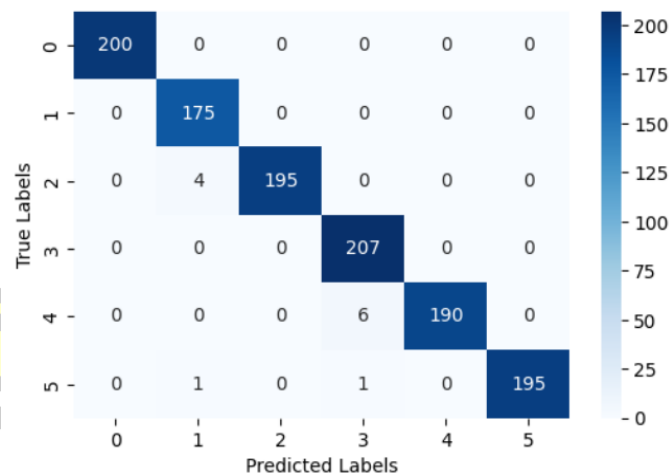


Figure 22 Confusion matrix for DenseNet169

Based on the results of the confusion matrix in Figure 22, with a batch size of 32, it can be seen that there are 1174 images of tea leaves that are correctly identified and 12 images that are not correctly identified. In confusion matrix, 0 represents algal_pot disease, 1 represents brown_blight, 2 represents gray_blight, 3 represents healthy

leaves, 4 represents helopeltis, and 5 represents red_spot. The performance evaluation for the three models can be seen in Table 5.

Table 5 Performance evaluation for the models

Models	Accuracy	Precision	Recall	F1-score
VGG16	0.906	0.906	0.898	0.902
ResNet50	0.952	0.952	0.945	0.948
DenseNet169	0.995	0.995	0.986	0.991

II. Result of detection of tea leaf pests and diseases

The results in Figures 23 and 24 demonstrate that the improved YOLOv5s outperforms the original YOLOv5s. The box validation loss displayed in Figure 23 measures the actual location of pests and diseases in tea images. It shows a consistent decreasing pattern, which can predict the training performance of the model. The box validation loss achieved by the improved YOLOv5s is lower than that of the original YOLOv5s, confirming the depth of the neural network. As the model learns, its performance also improves. As seen in Figure 24, the reduction in box validation loss contributes to the increase in mAP. The improved YOLOv5s achieves an mAP, which is higher than the original YOLOv5s model. This confirms that the training performance of the improved YOLOv5s is superior to the original YOLOv5s.

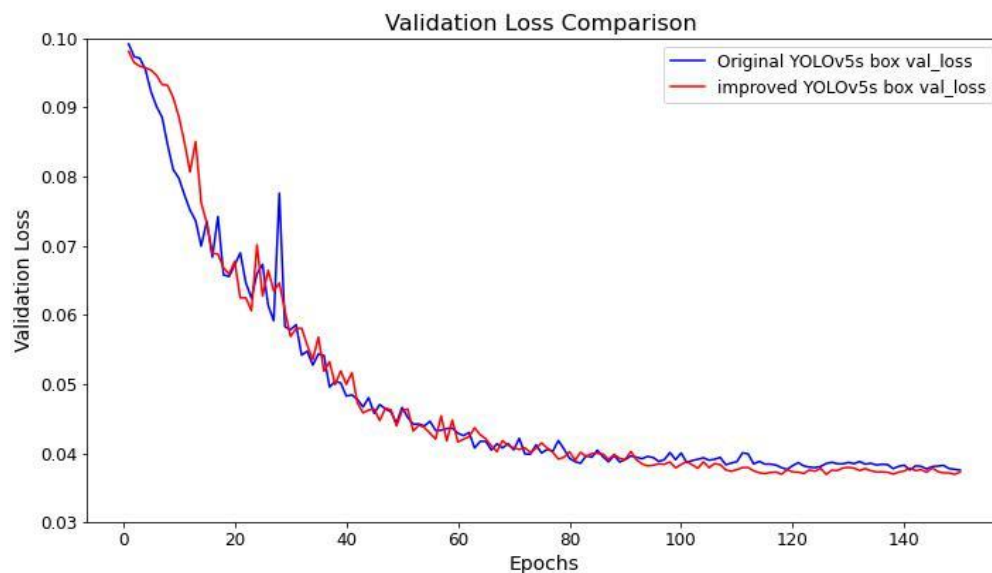


Figure 23 Box-validation loss comparison

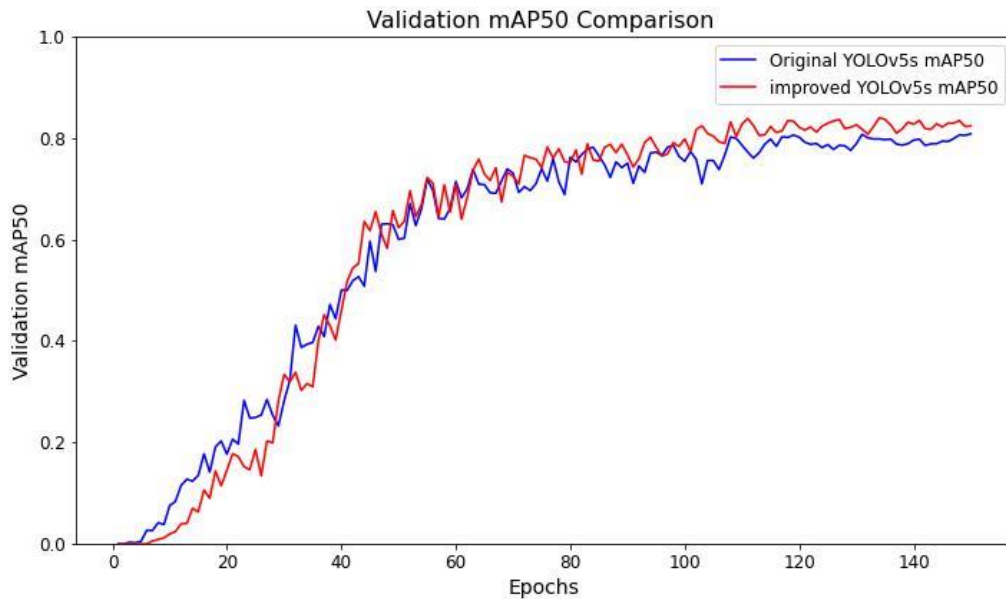


Figure 24 The mAP50 comparison

To evaluate the performance of the improved YOLO v5 model proposed in this study, we compared it with other typical convolutional neural networks such as Faster-RCNN, SSD, YOLO v8, YOLO v9 and YOLO v10, for tea disease detection using the test set. The results of the test set are shown in Table 6, which demonstrates that YOLOv10 model has achieved significant improvements compared to other models. Now the YOLOv10 model is beneficial for tea disease detection.

Table 6 The results data based on different models

Model	Precision	Recall	mAP50	F1-Score
Faster-RCNN	78%	70%	74%	74%
SSD	80%	71%	72%	75%
YOLO v5	81%	76%	80%	78%
Improved YOLO v5	82%	78%	82%	80%
YOLOv8s	84%	78%	82%	81%
YOLOv9s	85%	79%	83%	82%
YOLOv10s	87%	80%	84%	83%

Table 6 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 other models.

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.

The YOLOv10s shows superior performance in terms of preprocessing image speed, specifically, YOLOv10s has a higher postprocessing speed. 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[50]. 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.

III. 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 25. 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). Description of the role of each loss function: Localization loss `box_loss`: the error (GIOU) between the predicted box and the calibrated box, the smaller the more accurate the localization; Classification loss `cls_loss`: calculates whether the

anchor box is correctly classified with the corresponding calibration, the smaller the more accurately classified; Dynamic Feature Loss (dfl_loss): dfl_loss is a loss function used to regress the distance between the prediction frame and the target frame. When calculating the loss, the target box needs to be scaled to the feature map scale, i.e., divided by the corresponding stride, and the Ciou Loss is calculated with the predicted bounding box, while the regression DFLLoss is calculated with the distance from the center point of the predicted anchors to each side.

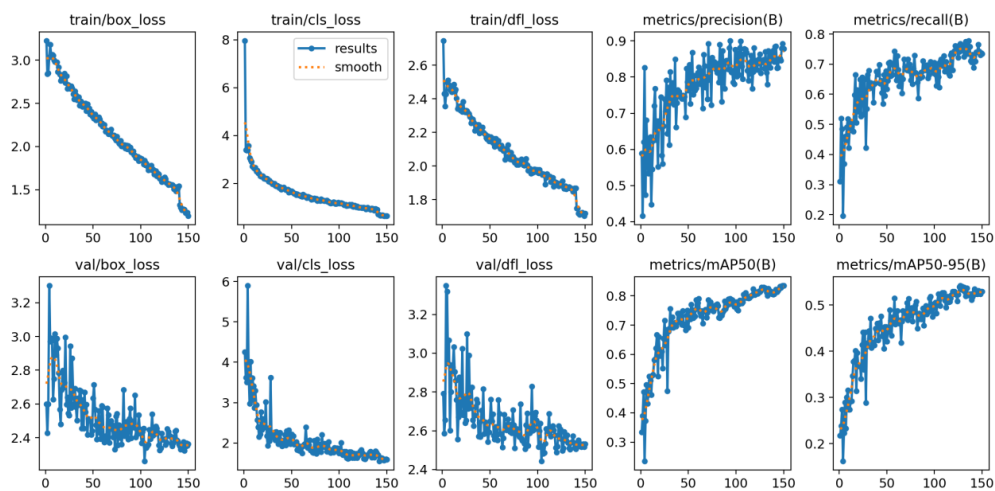


Figure 25 Performance of the YOLOv10s model

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 25, 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.

As shown in Figure 26, mAP represents the area covered 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.

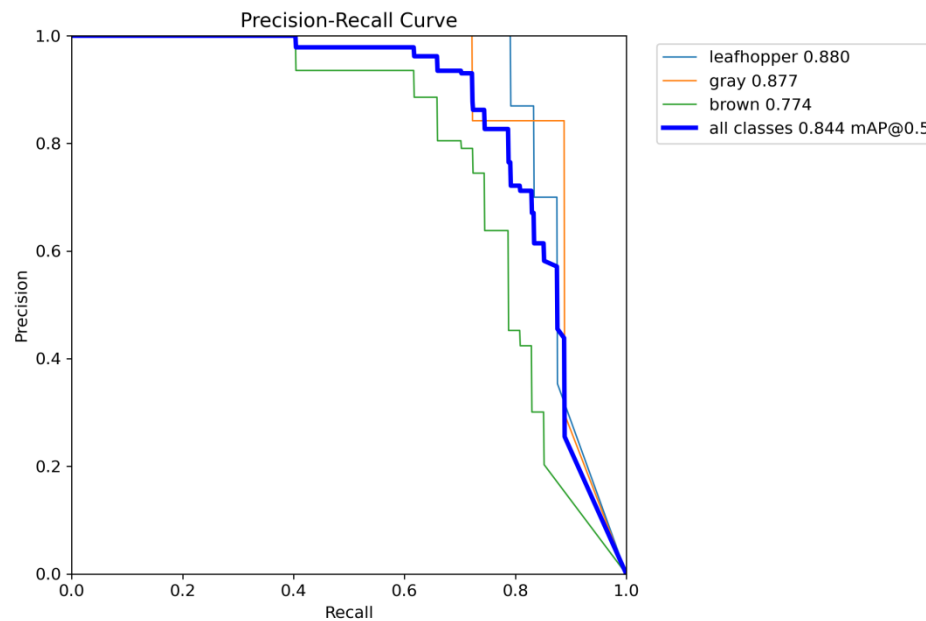


Figure 26 Precision-Recall curve

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 27, 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.

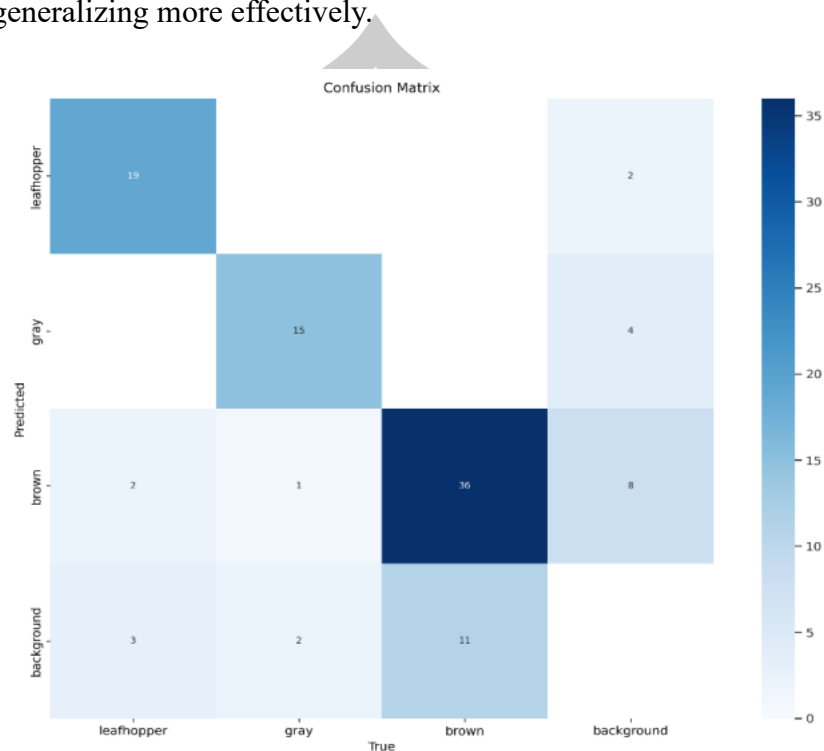


Figure 27 The confusion matrix

Here are the results of the images of tea diseases and pests detected by the model YOLOv10s as shown in Figure 28. 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.

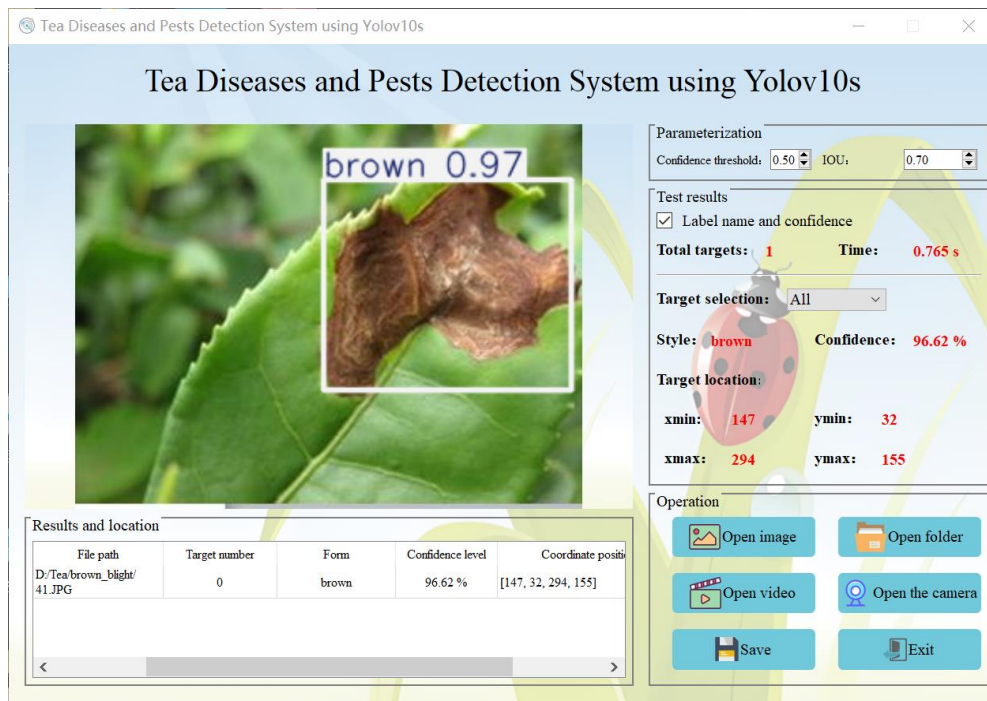


Figure 29 This tea disease detection system interface

Main functions of the software:

1. It can be used to detect and count the number of tea diseases in actual scenarios.
2. It supports pictures, videos and cameras for detection, and also support batch detection of pictures.
3. The interface can display real-time information such as target location, total number of targets, confidence level, time and so on; 4. support for saving the detection results of pictures or videos; 5. support for the detection of faces in real scenes.
4. It supports saving the detection results of pictures or videos.

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 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.

Show label name and confidence indicates whether the label name and confidence in the detection of the picture, show the default unchecked, if the check box will show the label name and confidence in the detection of the picture.

Total Targets: Indicates the number of faces in the screen, which can be used for headcount.

Target selection: a single tea disease target can be selected for location information, confidence level view.

Target position: indicates the detection frame of the selected target, the coordinate position of the upper left corner and the lower right corner. The default display is the largest confidence level of a tea disease target information.

(1) Picture detection instructions

Click the Open Picture button to select the picture to be detected, or click the Open Folder button to select the folder where the pictures need to be detected in bulk, the operation is demonstrated below:

After clicking the target drop-down box, you can select the specified target to display the result information.

Click the Save button, the results will be saved, the storage path is: save_data directory.

Note: 1. The target position on the right side displays the position of the target with the largest confidence level by default, and the target can be switched with the drop-down box. All test results are displayed in the table at the bottom left.

(2) Video detection instructions

Click on the video button to open the selection of the video to be detected, it will automatically display the detection results, click again to close the video.

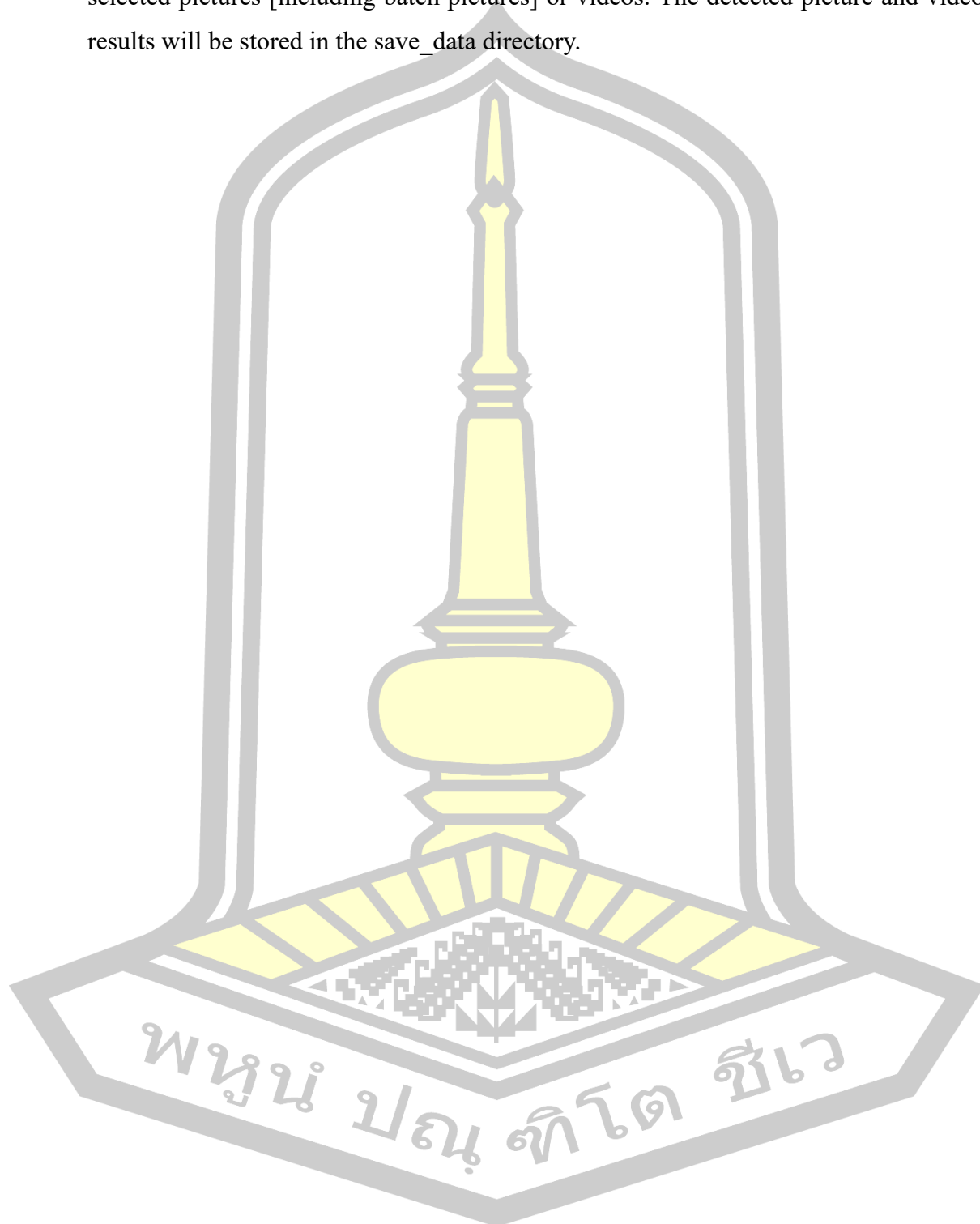
Click the Save button, the video detection results will be saved, the storage path is: save_data directory.

(3) Camera Detection Instructions

Click the Open Camera button to open the camera, you can open the camera, you can real-time detection, click again, you can close the camera.

(4) Save picture and video detection instructions

After clicking the Save button, it will save the detection results of the currently selected pictures [including batch pictures] or videos. The detected picture and video results will be stored in the save_data directory.



Chapter 5 Conclusions

Tea diseases and pests are diverse, and currently, the detection of tea plant diseases and pests mostly relies on the expertise of specialists. Therefore, this paper proposes a tea plant disease and pest recognition model based on ensemble learning.

For the tea leaf disease classification and identification model, three advanced deep learning image classification models VGG16, Resnet50 and DenseNet169 models were selected, and by adjusting the model parameters for the training of the dataset, it was concluded that the DenseNet169 model had the highest test accuracy and the lowest loss function value. Developing corresponding tea disease identification systems and the problem of data imbalance and generalization capability will be the focus of future research.

To achieve effective disease and pest recognition, we conducted the following work. Firstly, the widely used YOLOv5 model is chosen in the field of object detection. Secondly, we made three improvements to the YOLOv5 model for disease and pest detection. The CSP is upgraded to CSPK, which better enhances feature propagation, aiding in better information transmission between different layers of the network, thus improving performance. The ODCnv and CBAM attention mechanisms are added to allow the model to focus better on targets such as leaf blight.

As the updated version of the model for target detection becomes faster and faster, 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 some trained models: YOLOv9s, YOLOv8s, YOLOv5s, the improved YOLOv5s, as well as Faster R-CNN, SSD models. 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.

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.

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.

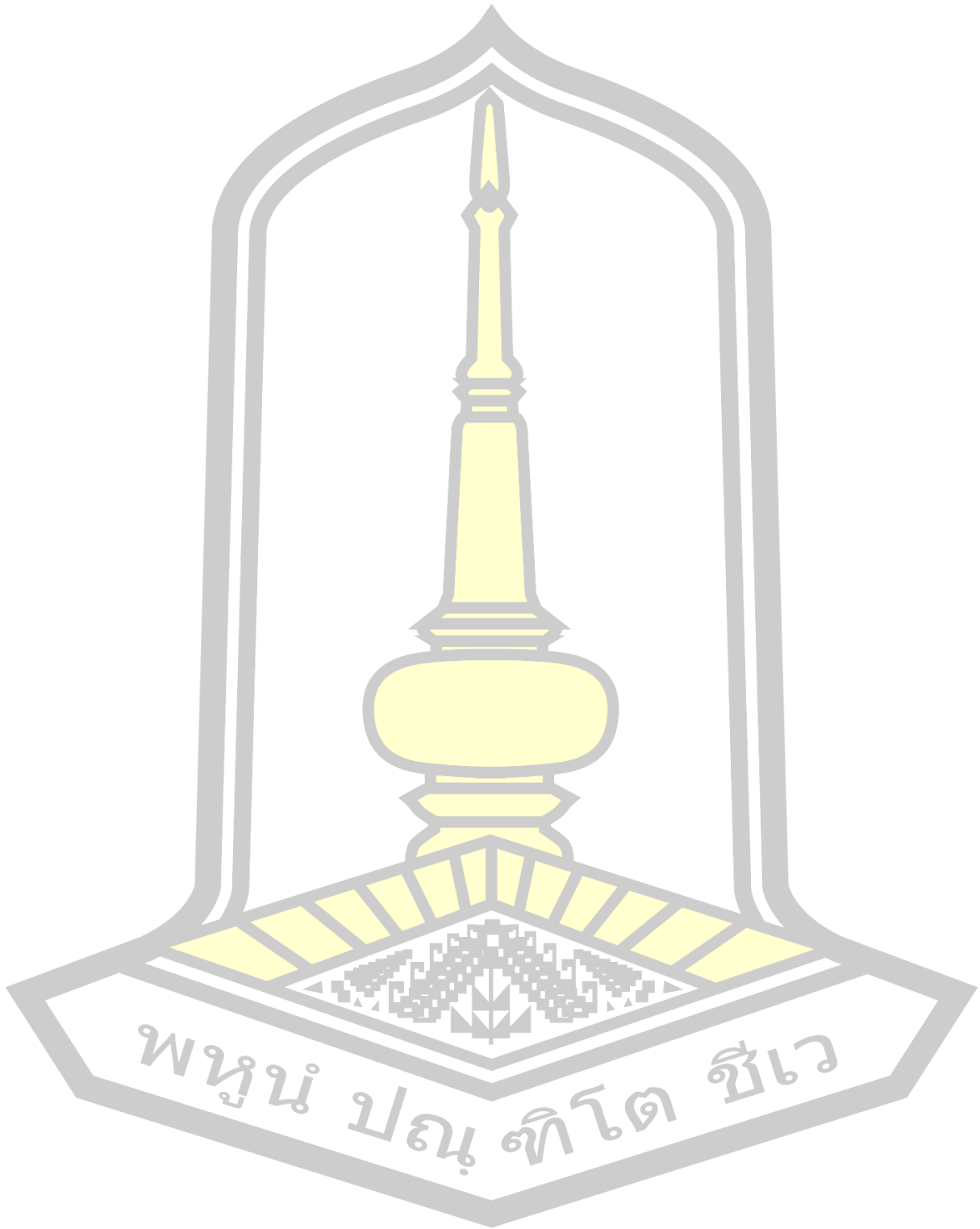
We will keep refining the model in subsequent work, looking for more effective techniques with fewer parameters. Additionally, we'll investigate how to use the tea leaf diseases and pests recognition model. To fully capture the process of tea diseases data, more photos gathered at varying heights will 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

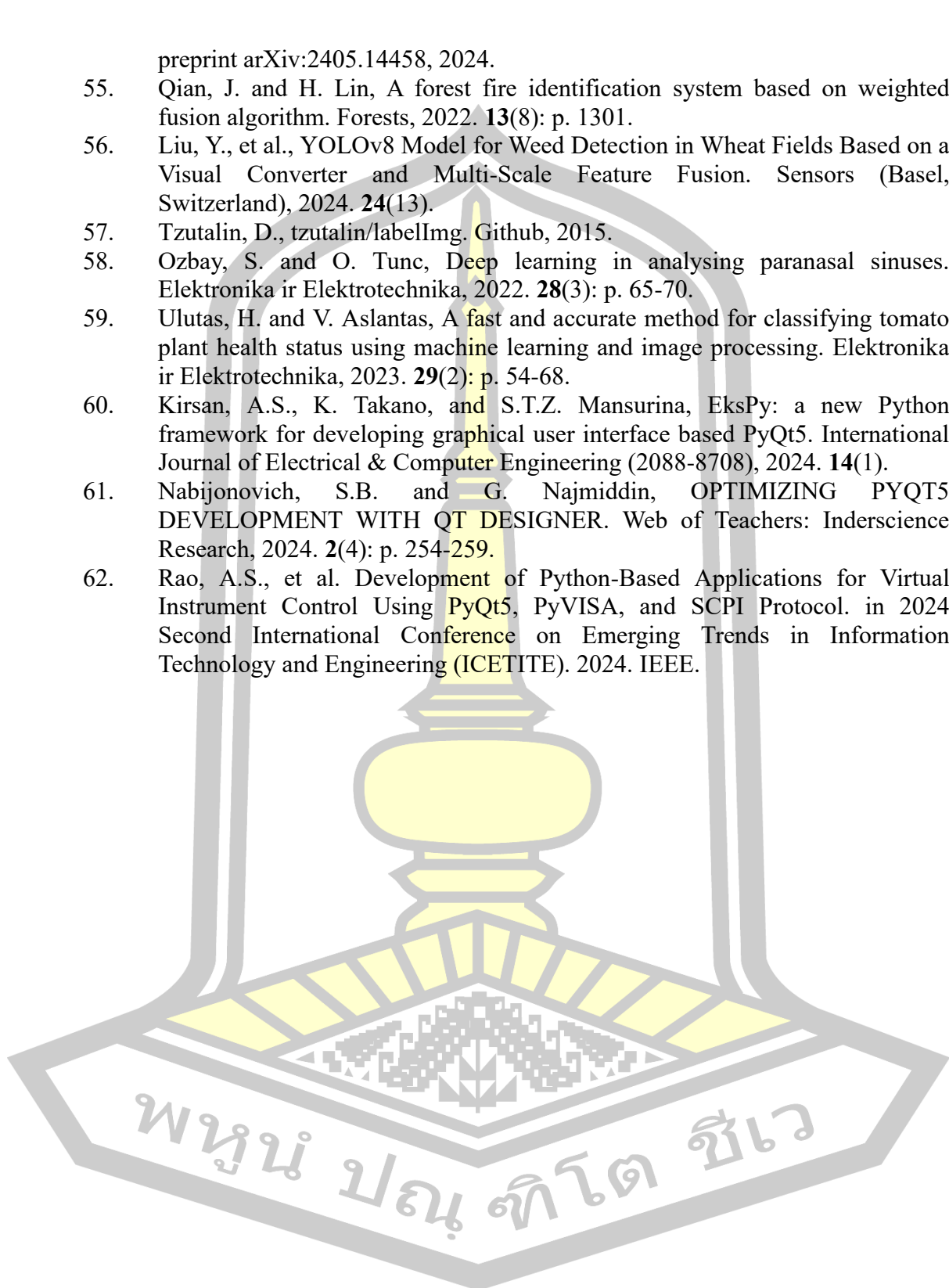


1. Bao, W., et al., Detection and identification of tea leaf diseases based on AX-RetinaNet. *Scientific Reports*, 2022. **12**(1): p. 2183.
2. Lertlam, K., et al., Development of A Semi-Auto Tea Packer Prototype for Agricultural SME in Thailand and Indonesia. *Engineering Access*, 2021. **7**(1): p. 30-35.
3. Ye, J.-H., et al., Bitterness and astringency of tea leaves and products: Formation mechanism and reducing strategies. *Trends in Food Science & Technology*, 2022. **123**: p. 130-143.
4. Hu, G., et al., A low shot learning method for tea leaf's disease identification. *Computers and Electronics in Agriculture*, 2019. **163**: p. 104852.
5. Trihardianingsih, L., A. Sunyoto, and T. Hidayat, Classification of Tea Leaf Diseases Based on ResNet-50 and Inception V3. *Sinkron: jurnal dan penelitian teknik informatika*, 2023. **8**(3): p. 1564-1573.
6. Chen, J., Q. Liu, and L. Gao, Visual tea leaf disease recognition using a convolutional neural network model. *Symmetry*, 2019. **11**(3): p. 343.
7. Wang, Y., et al., Research on Intelligent Recognition for Plant Pests and Diseases Based on Improved YOLOv8 Model. *Applied Sciences*, 2024. **14**(12): p. 5353.
8. Wang, C.-Y., I.-H. Yeh, and H.-Y.M. Liao, Yolov9: Learning what you want to learn using programmable gradient information. *arXiv preprint arXiv:2402.13616*, 2024.
9. Jocher, G., et al., ultralytics/yolov5: v6. 2-yolov5 classification models, apple ml, reproducibility, clearml and deci. ai integrations. *Zenodo*, 2022.
10. Li, C., A. Zhou, and A. Yao, Omni-dimensional dynamic convolution. *arXiv preprint arXiv:2209.07947*, 2022.
11. Woo, S., et al. Cbam: Convolutional block attention module. in *Proceedings of the European conference on computer vision (ECCV)*. 2018.
12. Ren, S., et al., Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, 2016. **39**(6): p. 1137-1149.
13. Liu, W., et al. Ssd: Single shot multibox detector. in *Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*. 2016. Springer.
14. Redmon, J. You only look once: Unified, real-time object detection. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.
15. Li, M., D. Pi, and S. Qin, An efficient single shot detector with weight-based feature fusion for small object detection. *Scientific Reports*, 2023. **13**(1): p. 9883.
16. Wei, X., et al., A study on Shine-Muscat grape detection at maturity based on deep learning. *Scientific Reports*, 2023. **13**(1): p. 4587.
17. Hossain, M.S., et al. Recognition and detection of tea leaf's diseases using support vector machine. in *IEEE International Colloquium on Signal Processing & Its Applications*. 2018.
18. Xiaoxiao, S., et al. Image recognition of tea leaf diseases based on convolutional neural network. in *2018 International Conference on Security, Pattern Analysis, and Cybernetics (SPAC)*. 2018. IEEE.

19. Bhowmik, S., A.K. Talukdar, and K.K. Sarma. Detection of disease in tea leaves using convolution neural network. in 2020 Advanced Communication Technologies and Signal Processing (ACTS). 2020. IEEE.
20. Hairah, U., et al., Classification of tea leaf disease using convolutional neural network approach. *International Journal of Electrical & Computer Engineering (2088-8708)*, 2024. **14**(3).
21. Latha, R., et al. Automatic detection of tea leaf diseases using deep convolution neural network. in 2021 International Conference on Computer Communication and Informatics (ICCCI). 2021. IEEE.
22. Krisnandi, D., et al., Diseases classification for tea plant using concatenated convolution neural network. *CommIT (Communication and Information Technology) Journal*, 2019. **13**(2): p. 67– 77-67– 77.
23. Ferentinos, K.P., Deep learning models for plant disease detection and diagnosis. *Computers and electronics in agriculture*, 2018. **145**: p. 311-318.
24. Gayathri, S., et al. Image analysis and detection of tea leaf disease using deep learning. in 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC). 2020. IEEE.
25. Hu, G., et al., Identification of tea leaf diseases by using an improved deep convolutional neural network. *Sustainable Computing: Informatics and Systems*, 2019. **24**: p. 100353.
26. Kamrul, M.H., et al. A deep learning based approach on categorization of tea leaf. in *Proceedings of the International Conference on Computing Advancements*. 2020.
27. Lee, S.-H., C.-C. Wu, and S.-F. Chen. Development of image recognition and classification algorithm for tea leaf diseases using convolutional neural network. in 2018 ASABE Annual International Meeting. 2018. American Society of Agricultural and Biological Engineers.
28. Hu, G. and M. Fang, Using a multi-convolutional neural network to automatically identify small-sample tea leaf diseases. *Sustainable Computing: Informatics and Systems*, 2022. **35**: p. 100696.
29. Ramdan, A., et al. Transfer learning and fine-tuning for deep learning-based tea diseases detection on small datasets. in 2020 International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications (ICRAMET). 2020. IEEE.
30. Hu, G., et al., Detection and severity analysis of tea leaf blight based on deep learning. *Computers & Electrical Engineering*, 2021. **90**: p. 107023.
31. Lin, J., et al., TSBA-YOLO: An improved tea diseases detection model based on attention mechanisms and feature fusion. *Forests*, 2023. **14**(3): p. 619.
32. Xue, Z., et al., YOLO-tea: A tea disease detection model improved by YOLOv5. *Forests*, 2023. **14**(2): p. 415.
33. He, H., et al., Recognition of soybean pods and yield prediction based on improved deep learning model. *Frontiers in Plant Science*, 2023. **13**: p. 1096619.
34. Chen, Z., et al., Plant disease recognition model based on improved YOLOv5. *Agronomy*, 2022. **12**(2): p. 365.
35. Li, R. and Y. Wu, Improved YOLO v5 wheat ear detection algorithm based on attention mechanism. *electronics*, 2022. **11**(11): p. 1673.

36. Li, Y., et al. Detection and identification of peach leaf diseases based on YOLO v5 improved model. in Proceedings of the 5th International Conference on Control and Computer Vision. 2022.
37. Qi, J., et al., An improved YOLOv5 model based on visual attention mechanism: Application to recognition of tomato virus disease. Computers and electronics in agriculture, 2022. **194**: p. 106780.
38. Zang, H., et al., Detection method of wheat spike improved YOLOv5s based on the attention mechanism. Frontiers in Plant Science, 2022. **13**: p. 993244.
39. Zhao, Y., et al., Precision detection of crop diseases based on improved YOLOv5 model. Frontiers in Plant Science, 2023. **13**: p. 1066835.
40. Zhu, S., et al., EADD-YOLO: An efficient and accurate disease detector for apple leaf using improved lightweight YOLOv5. Frontiers in Plant Science, 2023. **14**: p. 1120724.
41. Hussain, M., Yolov5, yolov8 and yolov10: The go-to detectors for real-time vision. arXiv preprint arXiv:2407.02988, 2024.
42. Ma, B., et al., Using an improved lightweight YOLOv8 model for real-time detection of multi-stage apple fruit in complex orchard environments. Artificial Intelligence in Agriculture, 2024. **11**: p. 70-82.
43. Chen, X., AW-YOLOv8: A novel deep learning model for detecting insect pests in cotton growth systems. Heliyon.
44. Yang, G., et al., A lightweight YOLOv8 tomato detection algorithm combining feature enhancement and attention. Agronomy, 2023. **13**(7): p. 1824.
45. Luo, D., et al., Citrus Diseases and Pests Detection Model Based on Self-Attention YOLOv8. IEEE Access, 2023.
46. Zheng, L., et al., Improvement of the YOLOv8 Model in the Optimization of the Weed Recognition Algorithm in Cotton Field. Plants, 2024. **13**(13): p. 1843.
47. Xie, S. and H. Sun, Tea-YOLOv8s: A tea bud detection model based on deep learning and computer vision. Sensors, 2023. **23**(14): p. 6576.
48. Ma, N., et al., Wheat Seed Detection and Counting Method Based on Improved YOLOv8 Model. Sensors, 2024. **24**(5): p. 1654.
49. Ye, R., et al., 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, 2024. **14**(5): p. 1034.
50. Sapkota, R., et al., Comprehensive performance evaluation of YOLOv10, YOLOv9 and YOLOv8 on detecting and counting fruitlet in complex orchard environments. Authorea Preprints, 2024.
51. Alif, M.A.R. and M. Hussain, YOLOv1 to YOLOv10: A comprehensive review of YOLO variants and their application in the agricultural domain. arXiv preprint arXiv:2406.10139, 2024.
52. Wang, C.-Y., I.-H. Yeh, and H.-Y. Mark Liao. Yolov9: Learning what you want to learn using programmable gradient information. in European Conference on Computer Vision. 2025. Springer.
53. Vo, H.-T., et al., Automating Tomato Ripeness Classification and Counting with YOLOv9. International Journal of Advanced Computer Science & Applications, 2024. **15**(4).
54. Wang, A., et al., Yolov10: Real-time end-to-end object detection. arXiv

- preprint arXiv:2405.14458, 2024.
55. Qian, J. and H. Lin, A forest fire identification system based on weighted fusion algorithm. *Forests*, 2022. **13**(8): p. 1301.
 56. Liu, Y., et al., YOLOv8 Model for Weed Detection in Wheat Fields Based on a Visual Converter and Multi-Scale Feature Fusion. *Sensors (Basel, Switzerland)*, 2024. **24**(13).
 57. Tzutalin, D., tzutalin/labellmg. Github, 2015.
 58. Ozbay, S. and O. Tunc, Deep learning in analysing paranasal sinuses. *Elektronika ir Elektrotechnika*, 2022. **28**(3): p. 65-70.
 59. Ulutas, H. and V. Aslantas, A fast and accurate method for classifying tomato plant health status using machine learning and image processing. *Elektronika ir Elektrotechnika*, 2023. **29**(2): p. 54-68.
 60. Kirsan, A.S., K. Takano, and S.T.Z. Mansurina, EksPy: a new Python framework for developing graphical user interface based PyQt5. *International Journal of Electrical & Computer Engineering (2088-8708)*, 2024. **14**(1).
 61. Nabijonovich, S.B. and G. Najmiddin, OPTIMIZING PYQT5 DEVELOPMENT WITH QT DESIGNER. *Web of Teachers: Inderscience Research*, 2024. **2**(4): p. 254-259.
 62. Rao, A.S., et al. Development of Python-Based Applications for Virtual Instrument Control Using PyQt5, PyVISA, and SCPI Protocol. in *2024 Second International Conference on Emerging Trends in Information Technology and Engineering (ICETITE)*. 2024. IEEE.



BIOGRAPHY

NAME XianghongDeng

DATE OF BIRTH January02,1986

PLACE OF BIRTH Hunan,China

ADDRESS Ningxiang City, ChangshaCity, Hunan Province, China

POSITION Kaifu District, ChangshaCity, Hunan Province, China

PLACE OF WORK No. 359, Sec. 1, Wanjiali North Road, Kaifu District, Changsha, China

EDUCATION 2001.9-2004.6 Niangxiang sixth Senior High School
2004.8-2005.6 Ningxiang YunFan high school
2005.9-2009.6 Bachelor's Degree of Huaihua College, majoring in electronic information science and technology
2009.9-2011.6 Full-time postgraduate study in Control Engineering, School of Electrical Engineering, Hunan University
2025, Philosophy of Degree(Ph.D),Electrical and Computer Engineering, Mahasarakham University

