Revolutionize Cherry Tree Health by YOLO-CLD: An Advanced Model for Precise Detection of Cherry Leaf Diseases

Zumin Wang
School of Information Engineering
Dalian University
Dalian, China
wangzumin@dlu.edu.cn

Yuhao Zhang
School of Information Engineering
Dalian University
Dalian, China
zhangyuhao@s.dlu.edu.cn

Xiaomei Li
School of Information Engineering
Dalian University
Dalian, China
1781066949@qq.com

Lingyan Hu
School of Information Engineering
Dalian University
Dalian, China
hulingyan@dlu.edu.cn

Chunming Huang
School of Medical
Dalian University
Dalian, China
huangjunming@dlu.edu.cn

Liming Chen
School of Computing
Ulster University
Belfast, UK
l.chen@ulster.ac.uk

Mohand Tahar Kechadi
School of Computer Science
University College Dublin
Dublin, Ireland
tahar.kechadi@ucd.ie

Rongli Gai*
School of Information Engineering
Dalian University
Dalian, China
gairli@sict.ac.cn

2023 IEEE Smart World Congress (SWC) | 979-8-3503-1980-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/SWC57546.2023.10448998

***Abstract*—The detection of cherry leaf diseases holds a crucial significance for maintaining the health status of cherry trees and improving their quality. This paper proposes an advanced model, YOLO-Cherry Leaf Disease (YOLO-CLD), for the efficient and accurate detection of three common cherry leaf diseases: cherry leaf brown spot, bacterial leaf shot hole, and Cherry lethal yellow. First, we built the model on the fast and efficient YOLOv7 network, and we enhanced it with an additional convolutional block attention module and a context Transformer module. The model has resulted in improved feature extraction and representation capabilities. The model is compressed using knowledge distillation to meet deployment requirements for different hardware conditions. The evaluation results on an embedded device show an average recognition accuracy of 85% with a frame rate of 18.5 frames per second (FPS). The proposed model demonstrates the potential of detecting cherry leaf diseases on the fly. The solution can be deployed on the cloud for optimal use.**

***Index Terms*—Cherry Leaf Disease, Object Detection, YOLO, Attention Mechanism, Knowledge Distillation**

I. INTRODUCTION

The cultivation and consumption of cherries has been increasing due to their nutritious fruits which have the potential to reduce the risk of health issues in humans [1]. The substantial sale price of cherries has made cherry cultivation a popular choice for fruit farmers. However, the extended planting area has attracted a significant increase in the number and types of pests and diseases affecting fruit trees, making disease prevention and control a challenging problem. The leaf diseases limit the fruit industry's development and can lead to substantial losses in cherry production [2]. To effectively prevent and treat crop diseases, it becomes urgent to identify the disease types accurately. However, the disease's identification is challenging for naked eye ordinary farmers to make timely judgments to choose the appropriate treatments [3].

The conventional approach for diagnosing and identifying diseases in large cherry trees involves visual inspection by trained professionals. However, this technique is inefficient and costly. The proliferation of advanced sensor devices (e.g., digital cameras) and the recent advances in digital agriculture have led to the widespread adoption of modern agriculture monitoring management systems, which have significantly enhanced yield production. Nonetheless, these systems still have poor recognition at a larger scale, as they rely mainly on expert experience to detect pests and

* Corresponding author.

The research was funded by the Science and Technology Innovation Foundation of Dalian (NO.2020JJ26SN058)

979-8-3503-1980-4/23/$31.00 ©2023 IEEE

diseases. Consequently, traditional pest identification methods have numerous limitations and issues, including labour-intensive and costly processes, poor standardisation, and non-automatable processes. Therefore, progress has been slow in this area. In recent years, deep learning and convolutional neural networks have experienced momentous breakthroughs in computer vision, enabling the automatic extraction of patterns of interest and end-to-end training. Compared to general machine learning methods, deep learning network models use shared weights to reduce memory usage and enhance learning, thus outperforming traditional machine learning techniques in computer vision and pattern recognition.

The current state of research on cherry leaf disease recognition based on target detection in cherry planting scenes is limited. Although there are some similar solutions [4], [5] deployed in the cloud, these cloud solutions can put a significant strain on network bandwidth [6] and raise privacy concerns. Edge computing presents a solution to these problems, but the limited computational capacity of these devices often requires lightweight models. This study proposes a different approach for disease detection networks based on YOLOv7 [7] to address this issue. The proposed solution focuses more on model compression for compatibility with both platforms: cloud and edge computing. First, we collected datasets for three cherry leaf diseases (cherry leaf brown spot, bacterial leaf shot hole, and Cherry lethal yellow). Next, we incorporated a mechanism into the Backbone module of YOLOv7 to prioritize disease features in the feature map. We also added a Transformer to the YOLOv7 Head module to unify context mining and self-attention learning. The modified system, referred to as YOLO-Cherry Leaf Disease (YOLO-CLD), improved the performance and accuracy of the model. The model is further optimized through knowledge distillation and ultimately deployed on an edge computing platform for real-time disease detection. Finally, the experimental evaluation shows that the new approach is superior to existing detection methods.

II. RELATED WORK

Classical computer vision approaches for identifying diseased leaves rely on creating custom feature extractors. These features are then fed into machine learning algorithms for image classification. Many of the existing methods presented in the literature achieve good results in prediction. In addition to being especially good image classifiers, they can extract detailed patterns from images and label target objects. In the following, we give an overview of the recent related work on disease classification and computer vision technology in agriculture.

A. Traditional Machine Learning

The traditional machine learning techniques for identifying crop diseases encompass four stages: image acquisition, image processing, feature extraction, and detection analysis. Early studies have used various techniques, such as support vector machines and genetic algorithms, to identify crop diseases. For instance, Bayesian discriminant analysis can distinguish between four types of tomato leaf diseases [8], [9], and K-means clustering methods can successfully identify leaf diseases in apple trees [10]. However, these traditional methods have some limitations. These include a cumbersome search process, sturdy subjectivity, heavy reliance on manual feature extraction, and low detection effectiveness in complex environmental conditions [11].

B. Deep Learning Approache

With the rapid development of digital agriculture, computer vision and data engineering, deep learning has proven to be more effective than traditional machine learning methods. The CNN-based deep learning model provides an end-to-end channel for automatic image feature learning and model parameter optimization making it easier to handle image-based crop disease diagnosis tasks. Recent studies have proposed various deep-learning models for crop disease diagnosis. For example, in [12] and [13], their respective authors proposed a grape leaf disease detection network using a dual attention mechanism and a rapid soybean disease identification method based on a residual attention network model.The limitations of image classification in real-world application scenarios necessitate alternative approaches that can capture finer details within images, such as the number and area of infected leaves. Object detection, in contrast, offers a better means of disease diagnosis by providing more granular information about the objects present in an image. For instance, a novel attention-enhanced YOLO model has been proposed by Son [14], which incorporates a leaf spot attention mechanism using region of interest (ROI) feature extraction. The model demonstrated superior performance in comparison to conventional object detection models. Albahli and Nawaz have previously proposed a novel approach [15], called the DenseNet-77-based CornerNet model, which demonstrates high accuracy in the localization and classification of tomato plant leaf anomalies.

Although many of the above solutions exist in agricultural disease problems, there are relatively few solutions for automation and deep learning applications for disease identification in growing cherries, and similar solutions for detection are not efficient for cherry leaf disease identification, while they cannot be efficiently identified on different hardware platforms according to actual needs.

III. METHODOLOGY

In this work, we first processed the dataset and then improved YOLOv7 in two ways. At the same time, we used knowledge distillation to compress the improved model. Finally, we deployed and tested the compressed model on Jetson Nano.

A. Dataset Acquisition and Processing

The experimental data samples consist of images of cherry leaf diseases. Images of diseased cherry tree leaves were captured using digital cameras, HD cell phones, and other tools from late July to early August 2022. To obtain images of diseased leaves from different angles, the diseased leaves were photographed at multiple angles approximately 5.0, 10.0, and 20.0 cm away from the leaves. The dataset contains three disease types: cherry leaf brown spot, bacterial leaf shot hole, and Cherry Lethal yellow (see Figure 1).

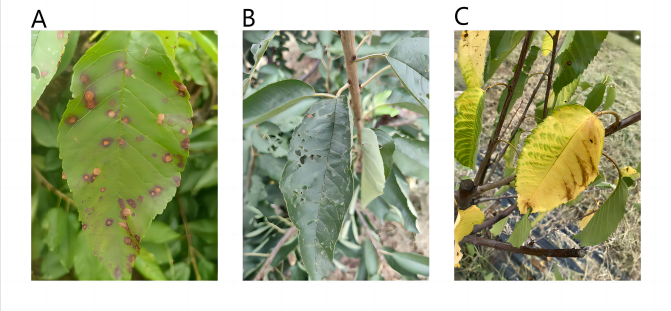

Fig. 1. Three diseases of cherry leaves.(A) Cherry leaf spot disease. (B) Bacterial leaf shot hole. (C) Cherry Lethal Yellow.

Firstly, the original images were processed using an image size adjustment technique to standardize the size to 640×480 or 480×640, which reduces the training and recognition time, improves the accuracy of the model and eliminates misleading images [16], leading to a dataset of 1069 images. The dataset is divided into three groups: training, validation, and test sets at a ratio of 8:1:1. Finally, image annotation is performed using the Make Sense data annotation tool [17], generating a TXT file with category and coordinate information. Additionally, the marked images underwent brightness enhancement and dimming using image enhancement techniques [18].

B. Overview of the YOLOv7 Network

YOLO (stands for "You Only Look Once") was introduced in 2016 as an object detection model with impressive detection speed and accuracy. After continuous refinement and enhancement, the latest version, YOLOv7, was released in 2022. Compared to other object detection models, YOLOv7 achieves higher detection accuracy at the same volume. Its speed and accuracy surpass those of currently known detectors in the range of 5 FPS to 160 FPS, making the YOLOv7 system highly suitable for real-time target detection.

The YOLOv7 architecture consists of two main parts: the Backbone layer structure and the Head layer structure. The Backbone layer structure contains three components: Conv-BN-SiLU (CBS), ELAN, and MP. CBS includes a convolutional layer, a BN layer, and a SiLU activation function, which facilitates the extraction of deep image features. The ELAN layer consists of multiple CBSs to maintain the input and output feature sizes, with the number of channels changing only in the first two CBSs before outputting the required channels in the last CBS. The MP layer, which comprises MaxPool and CBS, allows the conversion of any-sized feature map into a fixed-sized feature vector, thereby expanding the model's receptive field. The Head layer is responsible for network output and uses auxiliary head detection, re-parameterised convolution, and RepVGG for network structure transformation. During training, multi-channel branches can be used to enhance performance, while deployment guarantees optimal model performance. The Head layer also utilizes the ELAN structure, which plays a role like that of the Backbone layer, but with different "cats".We improved the YOLOv7 model (see Figure 2) so that the modified model can provide better identification of cherry leaf diseases.

C. Convolutional Block Attention Module (CBAM)

CBAM [19] is a lightweight attention module designed to enhance the detection accuracy and speed of the deep learning model. It achieves this by using the channel attention feature map that extracts the global features of each channel as input for special attention and calculates the mixed domain feature map. CBAM is composed of two modules: the channel attention module (CAM) and the spatial attention module (SAM) (see Figure 3). CAM helps the network focus on the image foreground needed for Ground Truth areas. SAM pays attention to positions rich in contextual information within the whole image. More importantly, CBAM can replace convolutional layers while having less computational complexity, with less than 26M number of parameters and around 4 GFLOPs.

In order not to affect the overall complexity of the model, while making the CBAM module fit well into the model structure. We add the CBAM module to each of the four ELAN structure outputs in the Backbone module of YOLOv7 to improve the accuracy of feature extraction and recognition.

D. Transformer Module

In conventional self-attention mechanisms, the system attempts to learn all the pairwise query-key relations independently without exploring the rich contexts in between [20].

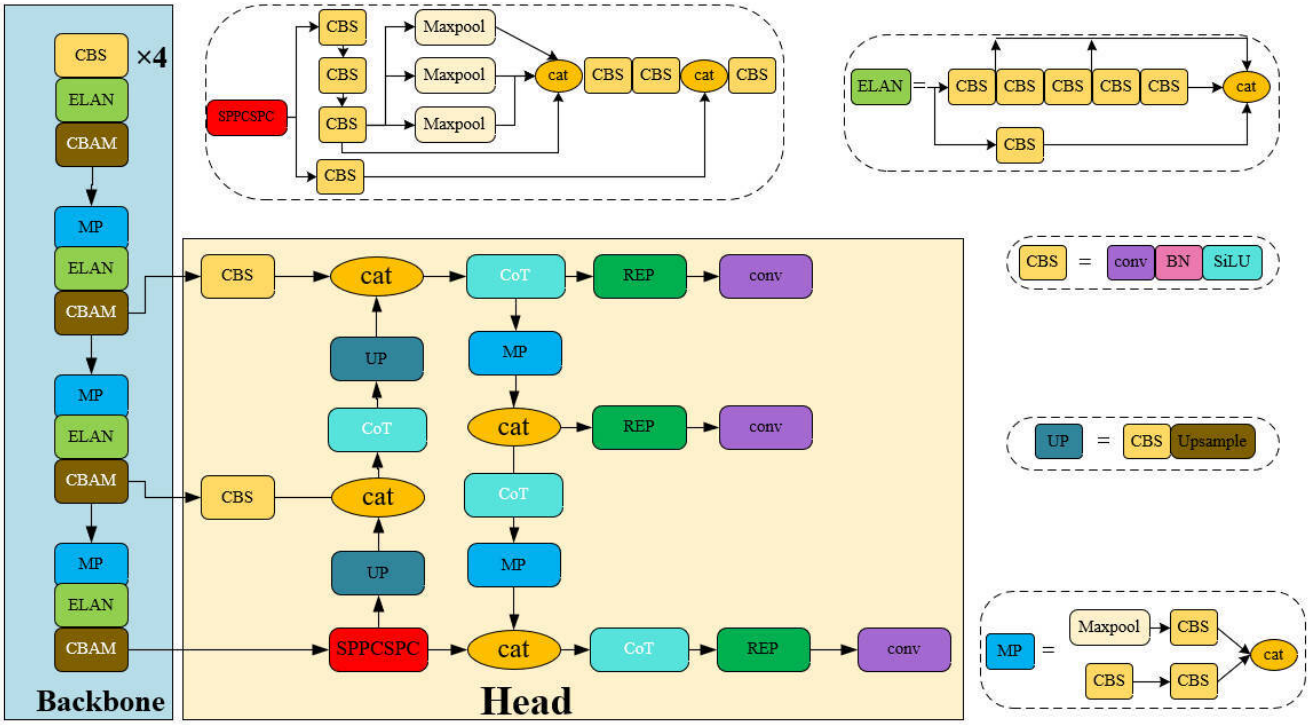

Fig. 2. The improved YOLOv7 structure.

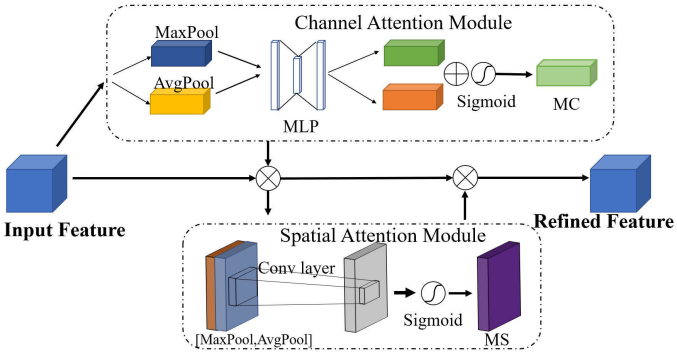

Fig. 3. Convolution block attention module

That severely limits the capacity of self-attention learning over 2D feature map for visual representation learning. However, contextual transformer (CoT) first employs $(k \times k)$ group convolutions over all adjacent keywords within a (k × k) grid (see Figure 4), where each key indicates the context. The learned contextualized keys $K^1 \in R^{H \times W \times C}$ naturally reflect the static context information between adjacent keys, where K^1 is the static context representation of the input X subject to the condition of connecting with query Q. The attention matrix is obtained by two consecutive 1×1 convolutions A.Such an approach enhances self-attention learning by mining static context K^1 , and then calculate the attended feature map K^2 by aggregating all values V in traditional self-attention to the contextual attention matrix A. Considering that the attended feature map K^2 captures the dynamic feature interaction between inputs, K^2 is represented as a dynamic context. Then the final output of CoT(Y) is the fusion of static K^1 and dynamic K^2 through the attention mechanism.

In the Head module of the YOLOv7 network, we replace each of the four ELAN modules in the original model with a CoT module. Using the CoT module can integrate the effect of contextual information mining and self-attentive learning into a unified architecture, which can reduce the interference of complex backgrounds on recognition features to focus on learning diseased leaf features. In addition, CoT has 22M parameters and about 3.3 FLOPs making it less computationally complex.

E. Knowledge Distillation

In this work, the original YOLOv7 model's relatively high computational cost poses a challenge for its deployment on edge devices. The proposed approach employs knowledge distillation, a type of transfer learning that compresses models to tackle the above challenge. The knowledge distillation mechanism uses two models: a more complex network with better performance and generalization ability, called the

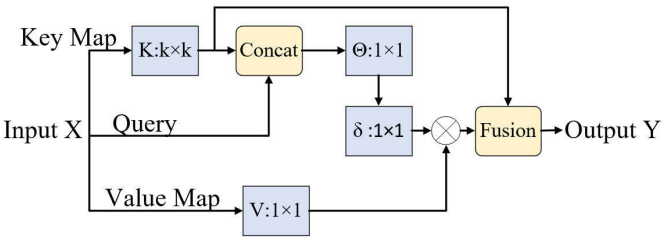

Fig. 4. CoT block.

teacher model, and a simpler network with fewer parameters, called the student model. The student model learns and imitates the teacher model to achieve similar or higher accuracy. In this study, the improved YOLOv7 network serves as the teacher model, while the YOLOv7-tiny model, a simple model structure, acts as the student model. During knowledge distillation, the student model learns the accuracy of the teacher model, enabling the deployment of the more accurate model on edge devices.

As mentioned by Hinton [21], traditional training uses maximum likelihood as the ground truth. However, the knowledge distillation mechanism uses the probability of the category output of the softmax layer as the ”soft target” for better generalisation. This approach introduces a temperature T to the original softmax function during the training process, where the increase of T results in a smoother output probability distribution of softmax and an increase in the entropy of the distribution. This approach amplifies the information carried by negative labels, which enables the model to pay more attention to them during training.

F. Model Deployment

We use the Pytorch framework to evaluate the performance of the proposed model on an edge computing platform. We started by generating a model file in PT format. Next, we deployed the model on Jetson Nano for real-time detection. Jetson Nano supports several widely used AI (artificial intelligence) frameworks and algorithms, including CUDA, cuDNN, and TensorRT, and exploits GPU acceleration to speed up the inference of the trained network [22].

TensorRT mainly implements two methods for network acceleration. The first is called Layer & Tensor Fusion. Taking the CSPNet module of YOLOv7-tiny as an example, it is usually called Conv. Conv, batch normalization (BN), and Leaky ReLu (LR) constitute a block, as shown in Figure 5. TensorRT reduces the number of layers by merging them horizontally or vertically. Horizontal merging can combine convolution, bias, and activation layers into one structure, occupying only one CUDA core. For instance, we merge Conv, BN, and LR into one block called CBL. Vertical merging can merge layers with the same structure but different weights into a meta-layer occupying only one CUDA core. The resulting multi-layer network (see Figure 5) has fewer layers and occupies fewer CUDA cores, so the entire model structure is compact, faster, and more efficient. The second method deals with data Precision Calibration and reduces the precision of the trained network from FP32 to either FP16 or INT8 data precision. Since backpropagation is not a requirement during reasoning, lower data precision reduces memory usage and the model size, speeding up its execution.

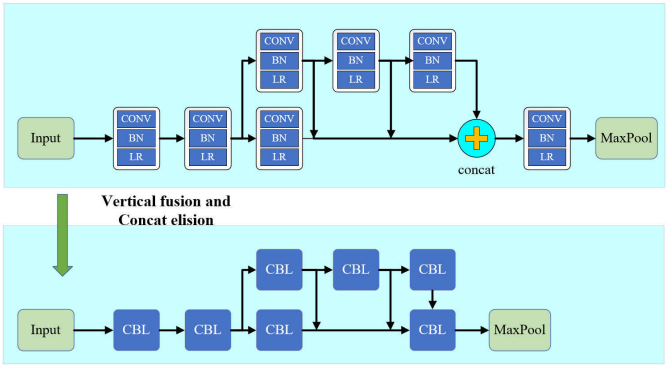

Fig. 5. The transformation of network structure is in TensorRT format.

IV. EXPERIMENTS AND RESULTS

A. Experimental Setup

We run all the experiments for evaluating the proposed model on a high-performance deep learning server equipped with an NVIDIA RTX 3080 graphics card with 10GB of graphics memory. The server uses Ubuntu 18.04 (64-bit) operating system. Table I gives the system's specifications used to deploy and run the experiments. We deployed the lightweight model implementation on the NVIDIA Jetson nano platform. The main objective of these experiments is to evaluate the model performance in real-world conditions.

Furthermore, the experimental setup involves configuring the model training with specific parameters. The initial learning rate is set to 0.01, while the final learning rate is 0.1. Stochastic gradient descent (SGD) is the method used for optimization. A momentum parameter, a weight decay rate, and batch size are set to 0.937, 0.0005, and 12, respectively. The model underwent 200 epochs of training. At the end of the training phase, the final network configuration is retained and saved. Then we perform the model evaluation on the test set.

B. Evaluation Metrics

This study evaluates target detection models based on various measurement standards. These include precision (P), recall (R), mean average precision (mAP), and FPS. In this evaluation, we consider two types of mAP: mAP@0.5 and mAP@0.5:0.95. The former represents the average AP at

TABLE I
EXPERIMENTAL ENVIRONMENT

Item	Specification
Operating system	Ubuntu 18.04
Central processing unit	Intel Xeon Platinum 8255C CPU @ 2.50GHz
Graphics processing unit	GeForce RTX3080(10GB)
Memory	45G
Programming environment	Python3.8,Pytorch1.9.0, CUDA11.1

an IoU threshold of 0.5 and primarily reflects the model's recognition ability. The latter represents the average value of each mAP at IoU thresholds ranging from 0.5 to 0.95, with a step size of 0.05, and mainly reflects the model's boundary regression ability and positioning effect. Note that "IoU" denotes the geometric relationship between the predicted and target boxes. More precisely, IoU is the ratio between the intersection area of the two bounding boxes and their union area, and the corresponding equation is as follows:

$$IoU = \frac{area(m_g \cap m_d)}{area(m_g \cup m_d)} \tag{1}$$

Where m_g is the real bounding box, and m_d is the predicted box obtained by the model. The geometric relationship between two boxes is converted into a loss function by comparing the geometric relationship of the two bounding boxes during training. The geometric relationship between the two boxes is corrected by backpropagation so that the two boxes overlap as much as possible. In the experiments, the IoU threshold is set to 0.5.

FPS indicates the number of detected pictures per second. The AP and the mAP measures are given by equations 4 and 5, respectively:

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$AP = \int_0^1 p(r)dr \tag{4}$$

$$mAP = \frac{\sum_{i=l}^{Q} AP_i}{Q} \tag{5}$$

TP (true positives) represents the number of lesions correctly detected by the model. FP (false positives) represents the number of lesions incorrectly detected by the model, FN (false negatives) represents the number of lesions that are not detected, and Q represents the number of lesion species.

TABLE II
COMPARISON OF THE DETECTION MODEL

Model	Precision	Recall	mAP@0.5	FPS	Size
Faster-RCNN	52.5%	82.0%	82.6%	15.9	108MB
SSD	90.1%	39.9%	70.4%	31.9	91.6MB
YOLOR-p6	86.7%	66.8%	78.2%	45.0	71.7MB
YOLOV 5L	97.0%	67.0%	82.7%	62.1	88.5MB
YOLO-CLD	92.4%	87.2%	88.1%	62.9	63MB

C. Improve Model Performance

The improved YOLOv7 model can locate and classify cherry leaf diseases. Tested on the prepared data set, YOLO-CLD accurately identifies three kinds of cherry leaf diseases, with a mAP@0.5 of 87%. Figure 6 shows an example of detected diseased cherry leaves.

Fig. 6. Examples of cherry leaf disease detected by YOLOv7-CTAM.

To showcase the performance effectiveness of our model architecture, we conduct a comparative analysis of the proposed YOLO-CLD system with several other object detection techniques, including Faster-RCNN [23], YOLOv5, YOLOR [24], and SSD [25]. We choose a comprehensive evaluation strategy, encompassing metrics such as precision, recall, mAP@0.5, memory used, and FPS, to assess the model's accuracy, detection speed, and memory usage. Table II presents the results of this comparative study. The results indicate that YOLO-CLD achieved the highest mAP of 88.1% on the test set, which is 5.4% higher than the second-best traditional model. Moreover, the YOLO-CLD has a similar FPS to YOLOv5L, far much better than the other models. Moreover, YOLO-CLD is much smaller (memory size) than the other models. Hence, we can conclude that the proposed model is much more compact, improves well the YOLOv7 model, and demonstrates high recognition accuracy, much faster than its competitors.

D. Ablation Experiment and Analysis

To investigate the contribution of various modules in YOLOv7-CLD, we conduct ablation experiments on the

TABLE III
RESULTS OF THE ABLATION EXPERIMENT

Model	Class	mAP @0.5	mAP @.5:95	Parameters	GFLOPs
YOLOv7	All	82.8%	55.1%	36492560	103.2
	BLSH	88.0%	48.8%		
	CLY	67.2%	53.3%		
	CLSD	93.2%	63.1%		
YOLOv7 -CBAM	All	86.5%	63.5%	36798872	103.8
	BLSH	87.9%	63.0%		
	CLY	83.8%	64.5%		
	CLSD	87.8%	63.0%		
YOLOv7 -CoT	All	85.8%	59.3%	31840912	91.7
	BLSH	82.3%	50.5%		
	CLY	83.4%	62.2%		
	CLSD	91.5%	65.1%		
YOLOv7 -CoT-CBAM	All	88.1%	60.2%	32147224	92.3
	BLSH	86.2%	53.8%		
	CLY	83.8%	59.6%		
	CLSD	94.2%	67.1%		

cherry disease leaf dataset. The original YOLOv7 model is considered the baseline, and we evaluate the impact of adding different modules to the model. Table III summarises the results of the experiments.

We replaced the last convolutional layer in the ELAN module with the CBAM module in the Backbone structure, resulting in a minor increase in parameters and GFLOP. While the detection accuracy for BLSH and CLSD slightly decreased, the detection accuracy for CLY significantly improved, resulting in an overall improvement in mAP@0.5 of 3.7%. In the Head structure, we replaced the entire ELAN with the CoT module, which has significantly reduced parameters and GFLOP and improved the detection accuracy for CLY and overall mAP@0.5. By separately applying the CBAM and CoT modules to the YOLOv7-CLD model, we achieve significant improvements in the detection accuracy of CLY and overall mAP@0.5 without increasing parameters and GFLOP. The mAP@0.5 for CLY was increased by 16.6% and 16.2%, respectively, compared to the baseline model.

Compared with the baseline YOLOv7, the final improved YOLOv7 model has significantly increased the overall detection, with mAP@0.5 jumped from 82.8% to 88.1% and mAP@0.5:95 from 55.1% to 60.2%, as FLOPS decreased from 103.2 to 92.3, the inference speed of the model has also improved. By adding the attention mechanism CBAM and the Transformer module CoT, the improved YOLOv7 has a better learning ability for feature extraction. The detection accuracy has improved while reducing the number of model parameters and FLOPs. Overall, the experimental results show that the YOLO-CLD model outperforms the baseline.

E. Model Compression and Testing

Despite the availability of the YOLOv7 tiny version for edge computing, its detection accuracy still lags behind the original and improved models we have proposed. To address its limitations, we employ the knowledge distillation approach, with our model serving as the teacher and the YOLOv7-tiny model as the student model. By varying the temperature parameter (T) during knowledge distillation training, we evaluate the impact of T on the student model's accuracy and identify the most accurate model. Training hyperparameters remained consistent with prior experiments.

Under the same teacher model structure, different temperature values T were used to evaluate the performance of the knowledge distillation method. The results are presented in Figure 7. The experimental results indicate that when T is equal to 3, the knowledge distillation has poor feature extraction performance, resulting in only a slight 1.5% improvement over the original YOLOv7-tiny. However, when T is 5 and 10, the lightweight models generally exhibit better feature extraction ability, and their performance is close to that of the improved YOLOv7 model.

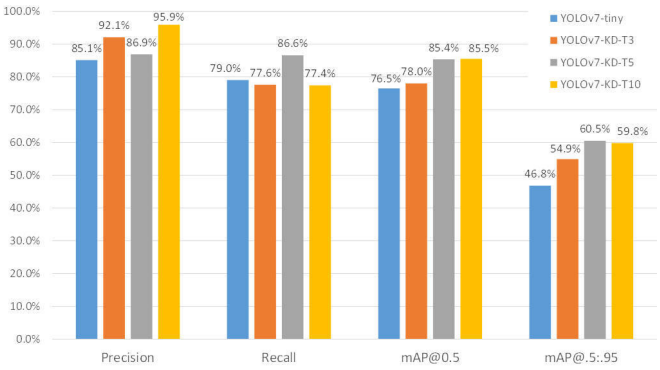

Fig. 7. Effect of temperature parameters on knowledge distillation.

F. Deployment Testing

Both YOLOv7 and the improved YOLOv7 have a large number of parameters that are not suitable for deployment on edge computing platforms. The lightweight YOLOv7-tiny model and the model generated through knowledge distillation have fewer network parameters, smaller sizes, and better performance. Table IV summarises the performance results obtained on a PC and Jetson Nano. The speed up of the YOLOv7-tiny and knowledge distillation models has improved significantly compared to YOLOv7-CLT. However, when these models are deployed on Jetson Nano, the lightweight models' speed up decreases to about 7.2 FPS. However, with TensorRT acceleration of the WTS file, speedup increases significantly, typically reaching over 17 FPS.

Based on the experimental results, the knowledge distillation models with T=5 and T=10 demonstrate similar performance. For the deployment testing, we select the

TABLE IV
SPEED OF FIVE YOLO SERIES ALGORITHMS.

Model	PT(PC)	PT(Jetson)	WTS(Jetson)
YOLOv7-CTAM	62.9FPS	Na	Na
YOLOv7-tiny	121.9FPS	7.1FPS	18.5FPS
YOLOv7-KD-T3	120.5FPS	7.2FPS	18.2FPS
YOLOv7-KD-T5	120.5FPS	7.2FPS	17.8FPS
YOLOv7-KD-T10	116.2FPS	7.2FPS	18.5FPS

knowledge distillation model with T=5, which achieve high recognition accuracy and detection speed of about 55ms when test on the Jetson Nano platform.

V. DISCUSSION

This study aims to develop an intelligent, efficient, and accurate model for detecting diseased leaves in cherry plantations. This research is based on the YOLOv7 network and leverages the attention mechanism and Transformer structure to identify three types of infected leaves. The integration of the Convolutional Block Attention Module (CBAM) led to a mean Average Precision (mAP) improvement of 3.7%. Further addition of the Context Transformer (CoT) module has resulted in mAP extra increase of 1.6%, reaching a final value of 88.1%. As a practical application in agriculture, we provided a lighter version of the proposed model to enable its deployment on mobile devices. To achieve such an edge computing version, a trade-off must be made between the model accuracy and speedup. The edge computing version has slightly lower accuracy but better speed.

In the dataset of cherry leaf diseases, an uneven distribution of the three types of diseases presents a challenge for model training. This results in a comparatively low accuracy for the identification of a certain type of disease. To address this issue, the model structure is modified to increase the weight of features specific to a certain type of disease, ultimately leading to high levels of accuracy in the identification of all three types of disease features.

In this study, we proposed two models for the detection of cherry leaf diseases. These models can be deployed in agricultural planting scenarios, either on the cloud or the edge. Furthermore, environmental data collected by sensors can be uploaded to the cloud server for analysis through edge devices, thereby enabling the creation of a smart agriculture platform. The development of ML systems and intelligent hardware is continuously advancing, and it is anticipated that it will eventually lead to the era of artificial intelligence of things (AIoT) dominance. Previously, Li et al. [26] developed an intelligent monitoring system to detect and identify pests and diseases on rice canopies. The system comprised a network camera, an intelligent model for detecting rice canopy diseases and insect pests, a web client, and a server. The camera captured images in the rice field, the model was deployed in the cloud for disease detection, and the client software for automatic monitoring of rice canopy diseases and insect pests displays the results. The lightweight model proposed in this paper can also be integrated into such a system for real-time cherry leaf disease detection through cloud-edge collaboration. [27].

Lightweight models deploy on edge computing platforms have been widely used in various research fields. Zhang et al. [28] deployed the improved YOLOv4-tiny on Jetson nano for real-time detection of strawberries. Bi et al. [29] proposed an efficient apple leaf disease identification method using the MobileNet model, which can be easily deployed on mobile devices. Despite achieving good results in detecting cherry leaf diseases, there are still several areas of improvement that need to be addressed. Firstly, the current disease types only include three common autumn leaf diseases. In actual production, there may be a wider range of diseases that need to be detected, thus it is necessary to collect and create a comprehensive dataset to improve the model's detection performance. Secondly, in addition to disease detection, a smart agricultural management platform should be established to comprehensively manage cherry planting and ensure good production and operation. This will involve the use of sensors to collect environmental data, which will be uploaded to the cloud for analysis.Finally, although this paper designs a high-precision compression model with a deployable embedded platform, the generalization ability of such models is limited, especially in complex tasks that require large amounts of data. In the future, techniques such as data augmentation, migration learning or meta-learning can be used to learn more powerful features from limited data and bring better generalization ability.

VI. CONCLUSION

In this research, the aim is to address the inefficiencies, high costs, and subjective judgment errors associated with the detection of diseases in large cherry plantations. The most advanced deep neural network in object detection, YOLOv7, is selected as the basis for improvement in order to meet real-world system requirements. The structure of the original model is modified by incorporating an attention mechanism and a Transformer structure to create the YOLO-CLD model, which demonstrates high accuracy in detecting three types of cherry leaf diseases. To minimize the impact of model storage size, inference time, and deployment cost, the improved YOLOv7 model is compressed using knowledge distillation and is deployed on Jetson nano as a TensorRT format model, which shows significant improvement in speed and performance. These two models can be deployed on the cloud or edge, facilitating cloud-edge collaboration, and providing effective solutions to the common problems in agricultural disease image detection. They hold great potential for improving the efficiency of agricultural production and lowering the environmental impact.

ACKNOWLEDGMENT

The work was funded by a grant from the Study on Dynamic Feature Extraction and Modeling of Sweet Cherry Growth in Solar Greenhouse Based on Internet of Things Intelligence (No.2020JJ26SN058).

REFERENCES

[1] M. F. Faienza et al., "Novel insights in health-promoting properties of sweet cherries," Journal of Functional Foods, vol. 69, p. 103945, 2020.

[2] Q. Dai, C.Zhou, J. Ai, and J. Zhang, "Pathogen identification of sweet cherry leaf blight in Liaoning Province", China Fruits, no. 16-19+109.

[3] H. Li, and B. Wang, " Deep Feature Fusion Method and Its Application in Leaf Disease Recognition, " Computer Systems & Applications, no. 349–355, 2022.

[4] Y. H. Gu, H. Yin, D. Jin, J.-H. Park, and S. J. Yoo, "Image-Based Hot Pepper Disease and Pest Diagnosis Using Transfer Learning and Fine-Tuning," Frontiers in Plant Science, vol. 12, 2021.

[5] R. K. Lakshmi and N. Savarimuthu, "PLDD—A Deep Learning-Based Plant Leaf Disease Detection," in IEEE Consumer Electronics Magazine, vol. 11, no. 3, pp. 44-49, 1 May 2022.

[6] J. Chen and X. Ran, "Deep Learning With Edge Computing: A Review," in Proceedings of the IEEE, vol. 107, no. 8, pp. 1655-1674, Aug. 2019.

[7] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors", arXiv preprint arXiv:2207.02696 (2022).

[8] M. Sammany and T. Medhat, "Dimensionality Reduction Using Rough Set Approach for Two Neural Networks-Based Applications," in Rough Sets and Intelligent Systems Paradigms, Berlin, Heidelberg, 2007, pp. 639–647.

[9] A. Chai, B. Li, Y. Shi, Z. Cen, H. Huang, and J. Liu, "Recognition of Tomato Foliage Disease Based on Computer Vision Technology,"Acta Horticulturae Sinica, no. 1423–1430, 2010.

[10] S. R. Dubey and A. S. Jalal, "Adapted Approach for Fruit Disease Identification using Images," Image processing: Concepts, methodologies, tools, and applications, IGI Global, 2013. pp.1395-1409.

[11] L. Li, S. Zhang, and B. Wang, "Plant Disease Detection and Classification by Deep Learning—A Review," IEEE Access, vol. 9, pp. 56683–56698, 2021.

[12] R. Dwivedi, S. Dey, C. Chakraborty, and S. Tiwari, "Grape Disease Detection Network Based on Multi-Task Learning and Attention Features," IEEE Sensors Journal, vol. 21, no. 16, pp. 17573–17580, 2021.

[13] M. Yu, X. Ma, H. Guan, M. Liu, and T. Zhang, "A Recognition Method of Soybean Leaf Diseases Based on an Improved Deep Learning Model," Frontiers in plant science, vol. 13, p. 878834, 2022.

[14] C.-H. Son, "Leaf Spot Attention Networks Based on Spot Feature Encoding for Leaf Disease Identification and Detection," Applied Sciences, vol. 11, no. 17, 2021.

[15] S. Albahli and M. Nawaz, "DCNet: DenseNet-77-based CornerNet model for the tomato plant leaf disease detection and classification," Frontiers in plant science, vol. 13, p. 957961, 2022.

[16] M. Rzanny, M. Seeland, J. Wäldchen, and P. Mäder, "Acquiring and preprocessing leaf images for automated plant identification: understanding the tradeoff between effort and information gain," Plant Methods, vol. 13, no. 1, p. 97, Nov. 2017

[17] Y. Zhang, Z. Guo, J. Wu, Y. Tian, H. Tang, and X. Guo, "Real-Time Vehicle Detection Based on Improved YOLO v5," Sustainability, vol. 14, no. 19, 2022.

[18] Á. Casado-García et al., "CLoDSA: a tool for augmentation in classification, localization, detection, semantic segmentation and instance segmentation tasks," BMC Bioinformatics, vol. 20, no. 1, p. 323, Jun. 2019.

[19] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional Block Attention Module," in Computer Vision – ECCV 2018: 15th European Conference, Munich, Germany, September 8–14, 2018, Proceedings, Part VII, Berlin, Heidelberg, 2018, pp. 3–19.

[20] Y. Li, T. Yao, Y. Pan and T. Mei, "Contextual Transformer Networks for Visual Recognition," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 45, no. 2, pp. 1489-1500, 1 Feb. 2023.

[21] G. Hinton, O. Vinyals, and J. Dean, "Distilling the Knowledge in a Neural Network." arXiv, Mar. 09, 2015. Accessed: Mar. 03, 2023.

[22] V. Sati, S. M. Sánchez, N. Shoeibi, A. Arora, and J. M. Corchado, "Face Detection and Recognition, Face Emotion Recognition Through NVIDIA Jetson Nano," in Ambient Intelligence – Software and Applications, Cham, 2021, pp. 177–185.

[23] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," IEEE Transactions on Pattern Analysis & Machine Intelligence, vol. 39, no. 06, pp. 1137–1149, Jun. 2017.

[24] C.-Y. Wang, I.-H. Yeh, and H.-Y. M. Liao, "You Only Learn One Representation: Unified Network for Multiple Tasks." arXiv, May 10, 2021. Accessed: Mar. 03, 2023. [Online].

[25] W. Liu et al., "SSD: Single Shot MultiBox Detector," in Computer Vision – ECCV 2016, Cham, 2016, pp. 21–37.

[26] S. Li et al., "An intelligent monitoring system of diseases and pests on rice canopy," Frontiers in plant science, vol. 13, p. 972286, 2022.

[27] Y. -Y. Chen, Y. -H. Lin, Y. -C. Hu, C. -H. Hsia, Y. -A. Lian and S. -Y. Jhong, "Distributed Real-Time Object Detection Based on Edge-Cloud Collaboration for Smart Video Surveillance Applications," in IEEE Access, vol. 10, pp. 93745-93759, 2022

[28] Y. Zhang, J. Yu, Y. Chen, W. Yang, W. Zhang, and Y. He, "Real-time strawberry detection using deep neural networks on embedded system (rtsd-net): An edge AI application," Computers and Electronics in Agriculture, vol. 192, p. 106586, 2022. [

[29] C. Bi, J. Wang, Y. Duan, B. Fu, J.-R. Kang, and Y. Shi, "MobileNet Based Apple Leaf Diseases Identification," Mobile Networks and Applications, vol. 27, no. 1, pp. 172–180, Feb. 2022.