agronomy

MDPI

Article

GYS-RT-DETR: A Lightweight Citrus Disease Detection Model Based on Integrated Adaptive Pruning and Dynamic Knowledge Distillation

Linlin Yang, Zhonghao Huang, Yi Huangfu, Rui Liu, Xuerui Wang, Zhiwei Pan and Jie Shi *

Mechanical and Electrical Engineering College, Yunnan Agricultural University, Kunming 650201, China; kgy200398@126.com (L.Y.); 18720500356@163.com (Z.H.); huangfuyi48@gmail.com (Y.H.); 15736033372@163.com (R.L.); 13619609645@163.com (X.W.); pp1234560424@163.com (Z.P.)
* Correspondence: ytbso@126.com

Abstract

Given the serious economic burden that citrus diseases impose on fruit farmers and related industries, achieving rapid and accurate disease detection is particularly crucial. In response to the challenges posed by resource-limited platforms and complex backgrounds, this paper designs and proposes a lightweight method for the identification and localization of citrus diseases based on the RT-DETR-r18 model—GYS-RT-DETR. This paper proposes an optimization method for target detection that significantly enhances model performance through multi-dimensional technology integration. First, this paper introduces the following innovations in model structure: (1) A Gather-and-Distribute Mechanism is introduced in the Neck section, which effectively enhances the model's ability to detect medium to large targets through global feature fusion and high-level information injection.(2) Scale Sequence Feature Fusion (SSFF) is used to optimize the Neck structure to improve the detection performance of the model for small targets in complex environments. (3) The Focaler-ShapeIoU loss function is used to solve the problems of unbalanced training samples and inaccurate positioning. Secondly, the model adopts two model optimization strategies: (1) The Group_taylor local pruning algorithm is used to reduce memory occupation and the number of computing parameters of the model. (2) The feature-logic knowledge distillation framework is proposed and adopted to solve the problem of information loss caused by the structural difference between teachers and students, and to ensure a good detection performance, while realizing the lightweight character of the model. The experimental results show that the GYS-RT-DETR model has a precision of 79.1%, a recall of 77.9%, an F1 score of 78.0%, a model size of 23.0 MB, and an mAP value of 77.8%. Compared to the original model, the precision, recall, the F1 score, the mAP value, and the FPS value have improved by 3.5%, 5.3%, 5.0%, 5.3%, and 10.3 f/s, respectively. Additionally, the memory usage of the GYS-RT-DETR model has decreased by 25.5 MB compared to the original model. The GYS-RT-DETR model proposed in this article can effectively detect various citrus diseases in complex backgrounds, addressing the time-consuming nature of manual detection and improving the accuracy of model detection, thereby providing an effective theoretical basis for the automated detection of citrus diseases.

Keywords: characteristic distillation; logic distillation; Group_taylor pruning; RT-DETR-r18; GD mechanism; SSFF mechanism

check for updates

Academic Editor: Valerio Cristofori

Received: 12 May 2025
Revised: 14 June 2025
Accepted: 16 June 2025
Published: 22 June 2025

Citation: Yang, L.; Huang, Z.; Huangfu, Y.; Liu, R.; Wang, X.; Pan, Z.; Shi, J. GYS-RT-DETR: A Lightweight Citrus Disease Detection Model Based on Integrated Adaptive Pruning and Dynamic Knowledge Distillation. *Agronomy* **2025**, *15*, 1515. https://doi.org/10.3390/agronomy15071515

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Yunnan is one of the central origins of citrus in the world, and it is also the dominant production area of extra-early and very late-maturing citrus in China [1]. Since the beginning of the 21st century, the citrus industry in Yunnan has continued to develop rapidly, and the planting area has expanded, making it one of the important citrus- producing areas [2]. According to the data from the National Bureau of Statistics and the Yunnan Statistical Yearbook, the total area of citrus planting in the Province in 2021 was 135,000 hm^2, with a total output of 2,437,700 tons, which was distributed among 16 prefectures. With the rapid development of science and technology, the timely detection and prevention of citrus diseases is an inevitable trend of citrus informatization and scientific planting [3–5].

In recent years, many researchers have achieved certain research results in the field of citrus disease identification [6–13]. However, these studies cannot be directly applied to real-world scenarios in complex contexts. The main challenges include the presence of multiple interfering factors, such as weeds, healthy leaves, overlapping leaves, and other diseased leaves, in a complex context, which make it difficult for the existing models to accurately and efficiently detect and locate citrus disease targets. In addition, the shape and scale of diseased citrus fruits are different, and some diseased fruits are easily blocked by leaves, so it is easy to miss them during the detection process. In 2023, the RT-DETR model by the Baidu Flying Pulp team outperformed the YOLO series in detecting small targets against complex backgrounds after 100 rounds of fine-tuning, demonstrating its technical advantages in object detection [14].

This paper designs and proposes a lightweight citrus disease detection and localization model, GYS-RT-DETR, which provides favorable conditions for deployment using edge devices, ensuring that citrus disease detection can be applied in real-time within agricultural scenarios. Based on the RT-DETR-r18 model as the benchmark model, this paper introduces the GD collection and distribution mechanism and the SSFF (Scale Sequence Feature Fusion) mechanism into the Neck network, which, respectively, improves the feature extraction and feature fusion capabilities of the Neck network for feature maps of different scales, and expands the receptive field of feature maps of different scales. The Focaler-ShapeIoU [15] loss function is used to alleviate the problem of imbalance in the model training samples and the accuracy of model prediction and positioning. In order to make the model lightweight, the Group_taylor [16] pruning strategy and the feature-logic distillation framework are used to optimize the recognition performance, memory footprint, and computational complexity of the model.

The main contributions of this article are as follows:

1. Model innovation: An innovative, GYS-RT-DETR, first-order object detection model is proposed. By introducing the GD mechanism and the SSFF mechanism, the Neck network structure of the original model is greatly improved. By using the Focaler-shapeIoU loss function, the recognition performance of the model is improved, and the problem of an unbalanced sample size during model training is solved.
2. Lightweight model: Compared with the original model, the proposed GYS-RT-DETR model has improved memory usage and computational complexity. The Group_taylor pruning strategy greatly reduces the memory footprint and computational complexity of the GYS-RT-DETR model, which makes it easy to deploy edge devices.
3. Knowledge distillation framework: A knowledge framework combining feature-logic distillation is proposed, which takes the pruning optimization model as the student model and RT-DETR-r50 as the teacher model, which can effectively solve the problem of information loss caused by the structural difference between the teacher and the students, and ensure a good detection performance, while realizing making the model lightweight.

2. Experiment Materials

2.1. Data Collection

The main data collection locations are Man Village, Mosha Town, Xinping Yi Dai Autonomous County, Yunnan Province, Ode Village, Mosha Town, Xinping Yi Dai Autonomous County, Yunnan Province, and Nanbang Community, Gasa Town, Xinping Yi Dai Autonomous County, and Yunnan Province. The data collection period was the fruit setting period of citrus fruit trees. The data acquisition tool is the Intel D455 camera. Images were captured under various lighting conditions to ensure robustness against different environmental factors, and the selection of disease characteristics is guided by relevant farmers and experts. The experimental dataset contains a total of 2377 images in 12 categories, and the data types and number structures are shown in Table 1. Some of the datasets are shown in Figure 1.

Table 1. Data structure of citrus diseases.

Citrus Disease Categories	Number of Images	Collection Time
Brown spot disease	63	
Citrus black spot disease	372	
Citrus canker	277	
Citrus oil spot disease	121	
Citrus scab disease	366	
Citrus huanglongbing	222	9:00 a.m.~11:00 a.m.,
healthy citrus fruits	604	15:00 p.m.~17:00 p.m.
Citrus leprosy	187	
Orange scab	72	
Soot disease	160	
Thrips	350	
Citrus lignification	91	



Figure 1. Partial dataset.

2.2. Dataset Creation and Operating Environment

First, data cleaning and data augmentation were performed on the dataset; this was then divided into training, validation, and test sets in a ratio of 7:2:1. The validation and

test sets are composed of new images that are independent of the training set to ensure an unbiased evaluation of the model's performance. The labeling work is completed using the labeling software (version 1.8.6).

The experimental environment used a PC configuration, including an AMD Ryzen 9 5900X 12-core processor and an RTX 4090 24G GPU, running on Ubuntu 22.04. The types of citrus diseases in the dataset are classified sequentially, with each specific disease being categorized into one class.

3. Research Methods

3.1. Model Design

3.1.1. GYS-RT-DETR Network Structure

This paper improves upon the RT-DETR-r18 baseline model to develop the GYS-RT-DETR model for real-time detection of citrus diseases. The improvements to the model mainly involve significant changes to the Neck part, utilizing the GD mechanism and the SSFF scale sequence fusion mechanism to enhance the internal information exchange capability of the model and strengthen the network's ability to learn multi-scale feature information. The improved model's network structure is shown in Figure 2.

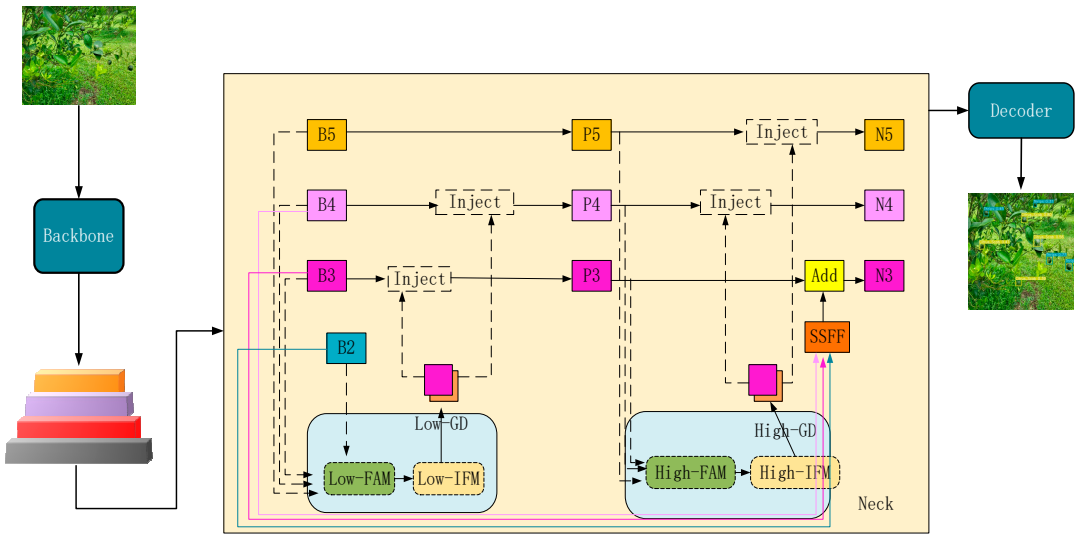


Figure 2. GYS-RT-DETR network structure (improved network structure).

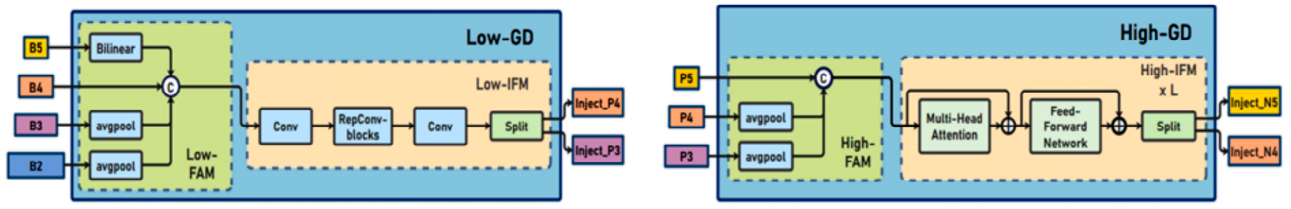
3.1.2. GD Collection and Distribution Mechanism

Wang et al. [17] proposed a collection and distribution mechanism for effective information exchange in YOLO by integrating multi-level features globally and injecting global information into higher levels. The GD mechanism includes two collection and distribution mechanisms—a low-level collection and distribution branch, and a high-level collection and distribution branch—and its basic principle is shown in Figure 3.

(1) As shown in Figure 3a, the low-level GD structure is mainly composed of two key modules: the low-level feature alignment module (Low-FAM) and the low-level information fusion module (Low-IFM). The input features are downsampled by using the average pooling operation in the low-level feature alignment module to unify the feature size to the minimum size of the group, so as to obtain accurate low-level feature alignment information. The low-level information fusion module consists of multi-layer reparametrized convolutional blocks (RepBlocks) and segmentation operations. In order to improve the effective integration of global information between

different levels, the attention mechanism is introduced, and the deep integration of information is realized.

(2) As shown in Figure 3b, the high-level GD structure is mainly composed of two key modules: the high-level feature alignment module (High-FAM) and the high-level information fusion module (High-IFM). The advanced feature alignment module consists of an average pooling layer that maps the dimensions of the input features to a uniform size. The advanced information fusion module mainly includes the Transformer module and the splitting operation. Each Transformer module includes a multi-head attention block, a feedforward network (FFN), and a residual connection, and since the Transformer module extracts high-level information, the pooling operation facilitates information aggregation and reduces the subsequent computational complexity in the Transformer module. This not only ensures that the model inference is not affected by the multi-head attention mechanism, which leads to performance degradation, but also improves the effective integration of global information between different levels.



a. Low-level collection and distribution branch b. High-level collection and distribution branch

Figure 3. The collection and distribution structure. (In (**a**), Low FAM and Low IFM are, respectively, the low-level feature alignment module and low-level information fusion module in the low-level branch. In (**b**), High FAM and High IFM are, respectively, the advanced feature alignment module and advanced information fusion module).

3.1.3. SSFF Module

On the basis of the YOLO segmentation framework, Kang et al. [18] enhanced the multi-scale information extraction capability of the network by using the Scale Sequence Feature Fusion (SSFF) module, which used the Triple Feature Encoder (TPE) module to fuse feature maps at different scales to increase detailed information. As shown in Figure 4, the SSFF module stacks the upsampled feature information as the input of the 3D convolutional network. At the same time, the SSFF module passes through the BN, SiLU activation function and MaxPool3d maximum pooling layer, respectively. Finally, the shape of the feature matrix is transformed by Squeeze and the filtered feature information is output.

It is worth noting that since the P_3-level feature information contains important information for small target detection, the input of the SSFF module only upsamples the feature information at the P_4 and P_5 levels. Such processing can not only expand the feature receptive field and increase the representation information of small targets, but also help to integrate multi-scale sequence features. At the same time, the maximum pooling layer is added at the tail end of the SSFF network structure to reduce redundant information and speed up network inference.

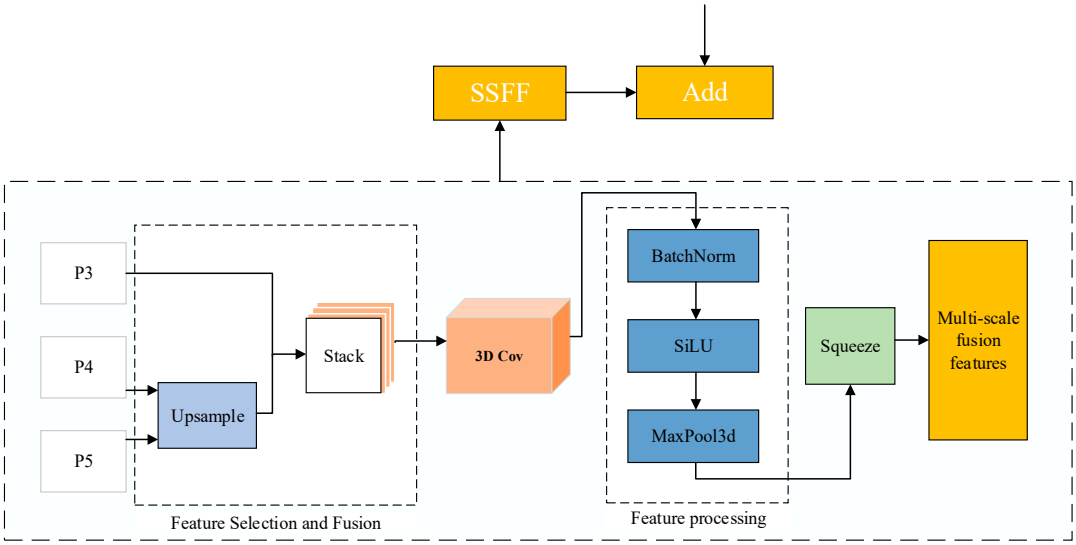


Figure 4. SSFF module.

3.1.4. Neck Network Structure Design

Figure 5 shows the difference between the network structure before and after the improvement. In the figure, B2, B3, B4, and B5 represent the feature maps extracted by the backbone network. The sizes are as follows: B2 $\in R^{N\times C\times H\times W}$, B3 $\in R^{N\times 2C\times \frac{H}{2}\times \frac{W}{2}}$, B4 $\in R^{N\times 4C\times \frac{H}{4}\times \frac{W}{4}}$, B5 $\in R^{N\times 8C\times \frac{H}{8}\times \frac{W}{8}}$. At the same time, N, C, H, and W represent the batch size, number of channels, height of the feature map, and width of the feature map, respectively. P3, P4, and P5 represent feature maps that contain feature information for small, medium, and large targets, respectively.

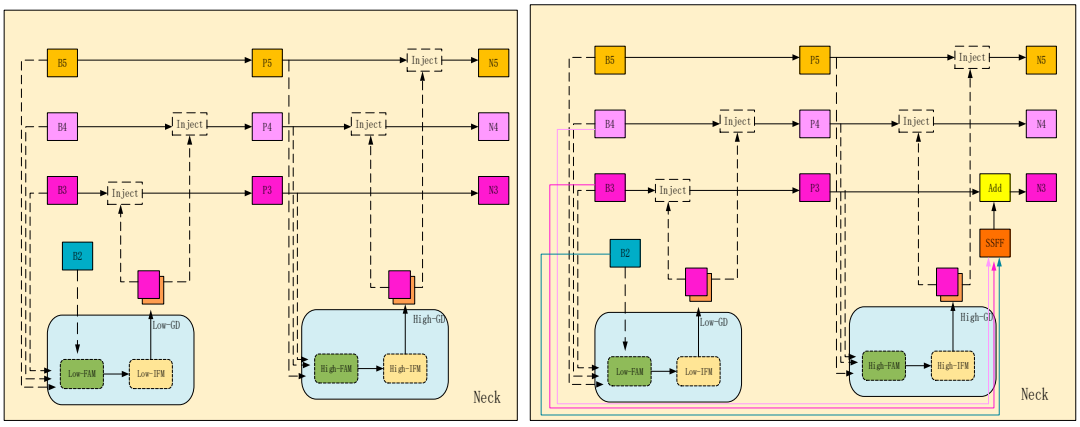


Figure 5. Comparison of Neck network structures (The (**left**) shows the Neck network structure of the RT-DETR-R18+GD network model, and the (**right**) shows the Neck network with the introduction of SSFF sequence feature fusion mechanism on the basis of RT-DETR-R18+GD).

3.2. Group_Taylor Pruning Strategy

Group_taylor pruning is a structured pruning method based on Taylor unfolding, which divides neural network parameters into groups according to specific rules, and then prunes them in groups, reducing model complexity by evaluating the impact of groups of neurons (such as convolutional kernels or channels) on the loss function and removing the groups with smaller contributions. The Group_taylor pruning strategy process can be shown in Figure 6.

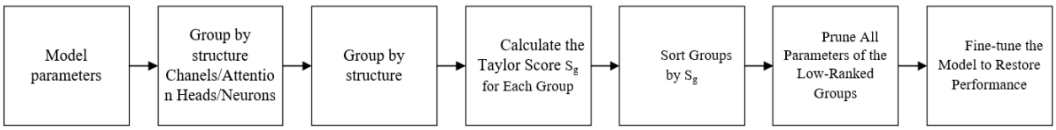


Figure 6. Group_taylor pruning strategy process.

Group_taylor pruning can reduce the scale of the model and balance computational efficiency and the model performance. Firstly, the Taylor expansion importance evaluation method preferentially prunes out the groups that have less impact on the output of the model, so as to compress the model while maintaining the accuracy of the model. Secondly, group-level pruning ensures the integrity of the filter, avoids the unstructured pruning affecting computational efficiency, and can be better used for hardware acceleration.

3.3. Feature-Logic Distillation Framework

As shown in Figure 7, the feature-knowledge distillation framework constructed in this paper uses a pruned model as the student model and RT-DETR-r50 as the teacher model. This framework integrates a dual strategy of feature distillation and logic distillation: on one hand, feature distillation calculates the loss between the intermediate network layers of the teacher model and the student model; on the other hand, logic distillation calculates the loss between the output layers of the student model and the teacher model. Ultimately, by integrating the output layer loss, intermediate layer loss, and the student model's own loss, a comprehensive loss function is formed. This ensures that the feature extraction and feature fusion capabilities of a certain layer in the student model's network structure are close to those of the corresponding layer in the teacher model. Additionally, by calculating the supervised loss between the teacher model's predictions and the true labels, the prediction accuracy of the student model is further constrained, ensuring that the distillation process enhances detection performance while maintaining model lightweightness. The feature-logic distillation loss function proposed in this article can be expressed mathematically as shown in Formula (1):

$$\text{Loss} = \alpha \times L_{\text{Feature}} + \beta \times L_{\text{Logit}} + \text{Hard_Loss} \quad (1)$$

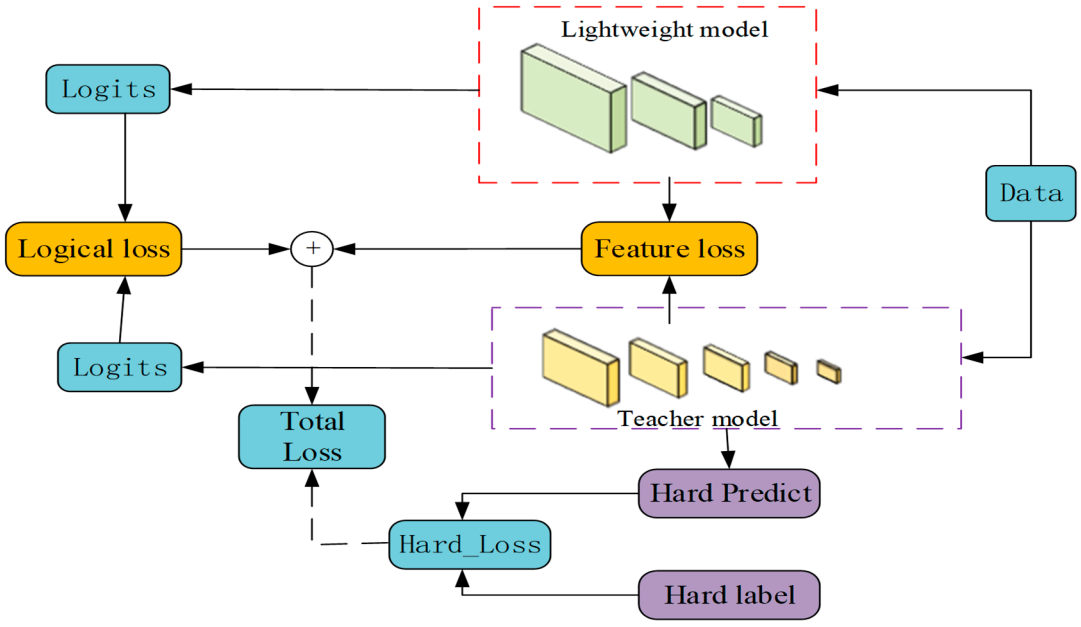


Figure 7. Knowledge distillation framework.

The meaning of each variable in the formula is as follows:

$L_{Feature}$: Measure the differences between the student model and the teacher model in the middle layer.

L_{Logit}: Measures the difference in raw scores between the student model and the teacher model at the final output layer, without passing through the Softmax layer.

Hard_Loss: Cross-entropy loss of student model on ground truth labels.

α and β: Hyperparameters that balance the weights of different loss items.

3.4. Focaler-ShapeIoU Loss Function

Existing bounding box regression methods typically consider the geometric relationship between the ground truth (GT) box and the predicted box, while neglecting the impact of inherent properties such as the shape and scale of the bounding boxes on the regression process [19]. In order to alleviate the problem of imbalance in the number of training samples and improve the effectiveness of the model for occluding targets and other difficult to detect positive samples, the Focaler-ShapeIoU loss function is used in this paper. In order to improve the detection performance of the model for occluding targets and other hard-to-detect positive samples, the ShapeIoU bounding box regression loss function was used to detect citrus diseases.

The formula for defining the ShapeIoU total regression loss function is as follows.

$$L_{ShapeIoU} = 1 - IoU + distance^{shape} + 0.5 \times \cap^{shape} \quad (2)$$

The meaning of each variable in the formula is as follows:

$distance^{shape}$: the similarity between the shape of the GT box and the predicted box.

IoU: the traditional regression loss function.

$\cap^{shape}$: the cost function of the shape of the GT box and the predicted box.

The problem of sample imbalance exists in various object detection tasks. In order to focus on different detection tasks in different regression samples, Zhang et al. [20] used a linear interval mapping method to reconstruct the IoU loss, which helps to improve edge regression. The formula is defined as follows:

$$IoU^{Focaler} = \begin{cases} 0, & IoU < d \\ \frac{IoU-d}{u-d}, & d \leq IoU \leq u \\ 1, & IoU > u \end{cases} \quad (3)$$

In the formula, $IoU^{Focaler}$ is the reconstructed Focaler-IoU function, where IoU is the Intersection over Union. Here, u is set to 0 and d is set to 0.95, which allows the model to focus on training samples of varying difficulty levels.

Combined with the advantages of Focaler-IoU to alleviate the equalization of the number of sample types and the advantages of ShapeIoU to improve the occlusion target, the Focaler-ShapeIoU loss function can solve the problem of imbalance in the number of training samples and improve the ability of the model to detect occluding targets, and the loss function formula is defined as follows:

$$L_{Focaler-ShapeIoU} = L_{ShapeIoU} + IoU - IoU^{Focaler} \quad (4)$$

3.5. Comparison with Existing Models

To clearly demonstrate the innovations and improvements of the proposed GYS-RT-DETR model, we conducted a detailed comparison with the existing Gold-YOLO and ASF-YOLO models, with the results shown in Table 2. Unlike Gold-YOLO, which incorporates only the Gather-and-Distribute (GD) mechanism but lacks the Scale Sequence

Feature Fusion (SSFF) mechanism, and ASF-YOLO, which includes the SSFF mechanism but not the GD mechanism, the GYS-RT-DETR model uniquely integrates both GD and SSFF mechanisms. In addition, GYS-RT-DETR employs the Focaler-ShapeIoU loss function to address issues of sample imbalance and inaccurate localization, which are not fully resolved in Gold-YOLO or ASF-YOLO. Meanwhile, GYS-RT-DETR utilizes Group_taylor pruning and feature-logic distillation to achieve lightweight performance.

Table 2. Comparison of GYS-RT-DETR with Gold-YOLO and ASF-YOLO.

Model	Base Model	Mechanisms	Loss Function
GYS-RT-DETR	RT-DETR-r18	Gather-and-Distribute (GD) Mechanism Scale Sequence Feature Fusion (SSFF)	Focaler-ShapeIoU
Gold-YOLO	YOLOv5	Gather-and-Distribute (GD) Mechanism Lightweight Adjacent Layer Fusion (LAF) Global Context Fusion Module (GCFM)	Traditional IoU loss + classification loss (cross-entropy)
ASF-YOLO	YOLOv5	Scale Sequence Feature Fusion (SSFF) Module Channel and Position Attention Mechanism (CPAM)	EIoU loss function + classification loss (cross-entropy)

3.6. Model Training and Model Evaluation Metrics

The improved GYS-RT-DETR model was trained, the training rounds were 150 rounds, the input image size was 640 × 640, the Batchsize was set to 16, the scaling factor in the ShapeIoU regression loss function was set to 0.6, and the optimizer was selected AdamW.

In this paper, precision (P), recall (R), F_1-score, mean average precision (mAP), FLOPs, model size, FPS, and model parameters are selected as model evaluation indicators. Table 3 shows the definitions of the formulas.

Table 3. Definition of formulas for various indicators.

Evaluation Indicators	Evaluation Formula
Precision (P)	$\frac{TP}{TP+FP}$
Recall (R)	$\frac{TP}{TP+FN}$
F_1-Scores	$\frac{2\times R\times P}{R+P}$
Average Precision (AP)	$\int_0^1 P(R)dR$
Mean Average Precision (mAP)	$\frac{\sum_{i=1}^{N} AP_i}{N}$
Accuracy (Acc)	$\frac{\mathrm{TP+TN}}{\mathrm{TP+TN+FN+TF}}$

The variables in the table have the following meanings:

1. TP is a true positive, predicted to be a positive sample, and the actual sample is also a positive sample.
2. FP is a false positive, predicted to be a positive sample, and actually a negative sample.
3. FN is a false negative, predicted to be a negative sample, and the actual sample is positive.
4. TN is a true negative, predicted to be a negative sample, and actually a negative sample.

4. Experiment Analysis

4.1. Experiment Environment Configuration

4.1.1. Conventional Experimental Parameter Settings

In the conventional experimental process, to ensure a fair comparison among different models, we set consistent experimental parameters: the Epochs were set to 250, the Batch size was set to 16, and the Optimizer was chosen as AdamW. The Image size was set to 640 × 640.

4.1.2. Model Pruning Experiment Parameter Settings

The channel pruning algorithm used in this paper can be divided into two stages, which are the pruning stage and the fine-tuning stage. In the pruning stage, different pruning strategies were used to prune, and different pruning rates were set to find the optimal pruning model. Due to the sparse parameters of the pruned model, it is important to fine-tune the parameters to ensure network performance. After fine-tuning, the pruned network parameters are re-optimized to the best state. Table 4 shows the fine-tuning of the pruning model and the setting of the pruning rate parameters.

Table 4. Parameter settings for different stages of model pruning.

Stage	Parameter Settings	
Pruning	Pruning rate/%	50.0~66.7%
	Global_pruning	False
Fine-tuning	Image size	640 × 640
	Epochs	250
	Batch size	16
	Optimizer	AdamW

4.1.3. Knowledge Distillation Experiment Parameter Settings

The parameters of the distillation test in this paper are shown in Table 5 below. The number of layers set by the teacher network layer and the student network layer in Table 4 is inconsistent, which is to better verify the feature extraction ability, output decision-making ability and feature fusion ability of the corresponding network layer in the student model network layer of the learning teacher model.

Table 5. Distillation test parameter settings.

Parameter Type	Parameter Value
Image size	640 × 640
Epochs	200
Batch size	16
Teacher Network Layer	3,5,6,7,25,22,19
Student Network Layer	3,5,6,7,28,25,30

4.2. Ablation Test

By observing Experiment 2 in Table 6, it can be seen that the model accuracy increased by 0.5 percentage points and the model size decreased by 3 MB when the GD collection and distribution mechanism was introduced into the RT-DETR model. However, the other indicators are reduced, which proves that the Neck network improved by the GD mechanism cannot effectively extract the feature information in the network. At the same time, the SSFF mechanism is introduced into the RT-DETR model, and it can be seen that compared with the RT-DETR-r18 model, the precision, recall, mAP value and F_1-score are reduced

compared with the RT-DETR-r18 model. This proves that the monotonous introduction of the SSFF mechanism cannot effectively enhance the performance of the model.

Table 6. Comparison of model improvement results. ("√" represents the fact that the experimental model adopts this mechanism, while "×" represents that the experimental model cancels this mechanism).

Baseline Model	GD	SSFF	P/%	R/%	mAP/%	F_1/%	FLOPs/G	Params/M	Model Size/MB	FPS(f/s)
RT-DETR-r18	×	×	75.6	72.6	72.5	73.0	57.0	19.8	48.5	344.5
RT-DETR-r18	√	×	76.1	71.9	72.1	73.0	60.0	22.2	45.5	275.1
RT-DETR-r18	×	√	73.1	72.1	70.2	72.0	61.5	20.1	39.2	267.2
RT-DETR-r18	√	√	78.4	73.7	73.9	75.0	63.2	22.5	45.9	239.5

After the GD mechanism is improved to the Neck network part of the RT-DETR model, the SSFF scale sequence feature fusion module is introduced again. By observing Experiment 4 in Table 6, it can be seen that the precision, recall, mean average precision and F_1-score of the improved model combined with SSFF and GD mechanisms are increased by 2.8%, 1.1%, 1.4% and 2.0%, respectively. Based on these indicators, this fully proves that the overall model performance of the RT-DETR model has been significantly improved after the introduction of the GD mechanism and the SSFF module, and fully proves that the GD mechanism and the SSFF mechanism have good adaptability, and the improvement of the model is effective.

In order to visually verify the effectiveness of the model, this paper selects two challenging images to test its detection effect in complex backgrounds. As shown in Figure 8, the GYS-RT-DETR model successfully detected healthy citrus and citrus with crust disease in a strongly interfering environment, depicted in green, as shown in Figure 8a. However, the original RT-DETR-r18 model could not accurately identify and locate the location and disease type of citrus in the face of such strong interference, as shown in Figure 8b.



a.GYS-RT-DETR b.RT-DETR-r18

Figure 8. Comparison of detection effects ((**a**) on the left shows the detection effect of GYS-RT-DETR model, and (**b**) on the right shows the detection effect of RT-DETR-R18 original model).

4.3. Comparison of Different Loss Functions

The Focaler-ShapeIoU, WIoU, CIoU and GIoU loss functions were used to validate the models. As shown in Table 7, the accuracy of the Focaler-ShapeIoU loss function is increased by 0~4.2% compared with other loss functions, which indicates that the Focaler-ShapeIoU loss function is more accurate in the detection of citrus diseases. The recall rate of Focaler-ShapeIoU loss function is increased by 0.1~4.8%, which proves that the missed detection rate of Focaler-ShapeIoU loss function in real scenarios is greatly reduced compared with other loss functions in complex backgrounds. The $mAP_{0.5}$ and $mAP_{0.5:0.95}$ of the Focaler-ShapeIoU loss function were increased by 2.6~5.1% and 2.9~4.1%, respectively. This proves that the average precision of the Focaler-ShapeIoU loss function is improved under different detection thresholds. The F_1 -score of the Focaler-ShapeIoU loss function is increased by 2~3%, which proves that it has the best overall performance compared with other loss functions. This also proves that it can effectively alleviate the imbalance of training samples and improve the detection ability of masked objects.

Table 7. Comparison results of different loss functions. ("√" represents that the experimental model adopts this Loss function, while "×" represents that the experimental model cancels this loss function).

Baseline Model	Focaler-ShapeIoU	WIoU	CIoU	GIoU	P/%	R/%	$mAP_{0.5}$/%	$mAP_{0.5:0.95}$/%	F_1/%
RT-DETR-r18 + GD+SSFF	√	×	×	×	78.0	73.8	76.5	49.8	75.0
RT-DETR-r18 + GD+SSFF	×	√	×	×	78.0	73.7	73.9	46.9	75.0
RT-DETR-r18 + GD+SSFF	×	×	√	×	73.8	72.2	72.4	46.8	73.0
RT-DETR-r18 + GD+SSFF	×	×	×	√	75.9	69.0	71.4	45.7	72.0

To intuitively verify the comparison of the Focaler-ShapeIoU loss function with other loss functions in terms of model performance, as shown in Figure 9, panels a, b, c, and d represent the Precision–Recall (P-R) curves obtained when the RT-DETR-r18 model, augmented with the GD mechanism and SSFF mechanism, employs CIoU, GIoU, WIoU, and Focaler-ShapeIoU loss functions, respectively. By comparison, it is evident that Focaler-ShapeIoU achieves performance improvements in the detection of all disease categories, particularly for those with fewer samples, such as brown-spot and orange scab, where there is a significant enhancement. In contrast to the other three P-R curves, they still exhibit limitations when it comes to categories with fewer samples. This indicates that Focaler-ShapeIoU can effectively address the issue of data imbalance and enhance the model's detection capability for categories with small sample sizes.

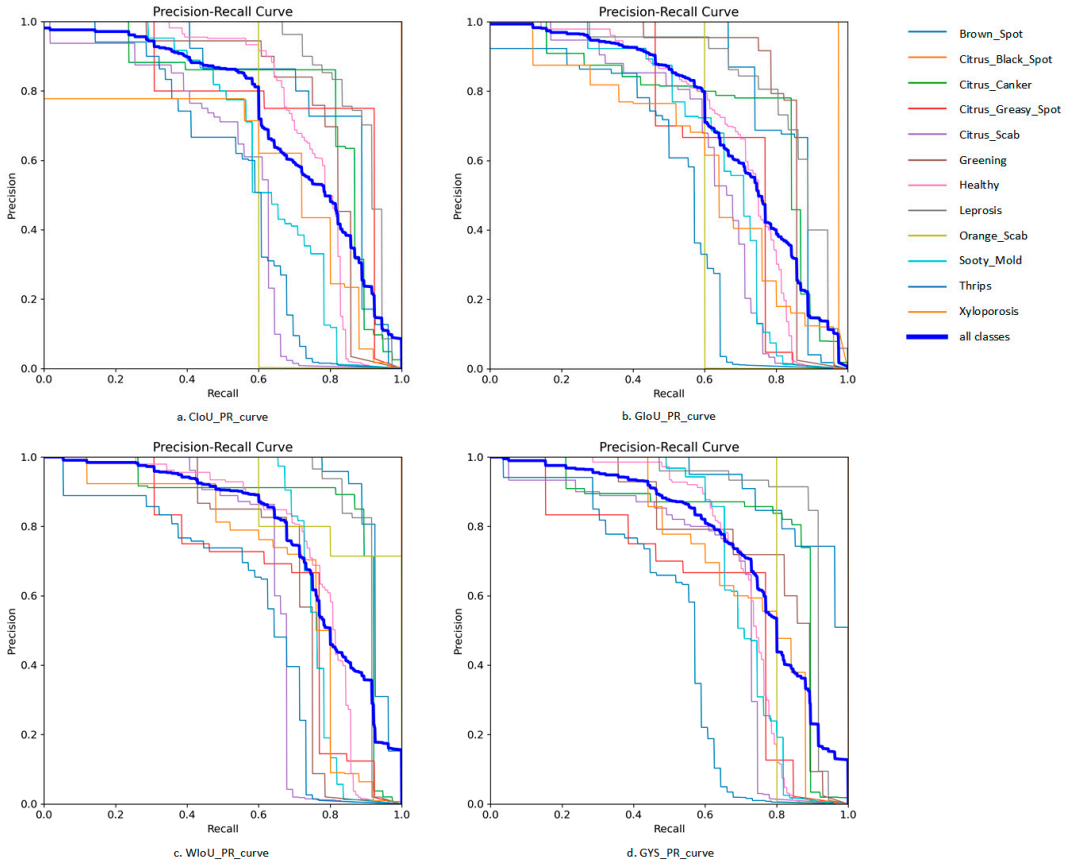


Figure 9. Precision–recall curves of the model under different loss functions.

4.4. Comparison and Analysis of Different Models

By replacing different mainstream backbone networks and mainstream YOLO series models, the superiority of the detection performance of the GYS-RT-DETR model was compared and analyzed.

As shown in Table 8, the precision of the GYS-RT-DETR model is improved by 2.4~16.7% compared with other models, which proves that the improved model has higher accuracy in detecting citrus diseases. Compared with other models, the recall rate of the GYS-RT-DETR model is increased by 1.2~12.6%, which proves that the improved model is less likely to miss difficult samples in complex environments. Compared with other models, the average precision of the GYS-RT-DETR model is increased by 4.0~16.7%, which proves that it has good detection performance under different judgment thresholds. Compared with other models, the F_1-score of the GYS-RT-DETR model was increased by 2.0~14%, which proved that it had better robustness and detection performance than other models. After a comprehensive comparative analysis of the key performance indicators of the improved model and other models, including FLOPs, Params, model size, and frame rate, the overall performance of the improved model is still among the highest.

By comparing the training curves of different models (Figure 10), it can be intuitively observed that the GYS-RT-DETR model has the highest average precision and converges more quickly. This fully demonstrates that the GYS-RT-DETR model exhibits better generalization and detection performance compared to other models.

Table 8. Performance comparison results of different models.

Model	P/%	R/%	mAP/%	F_1/%	FLOPs/G	Params/M	Model Size/MB	FPS (f/s)
GYS-RT-DETR	78.0	73.8	76.5	75.0	63.2	22.5	43.8	239.5
rtdetr-EfficientViT	70.3	63.9	64.3	64.3	23.7	10.7	22.7	255.8
rtdetr-VanillaNet	74.5	65.4	68.8	69.0	110.2	21.7	55.8	182.8
rtdetr-unireplknet	65.7	65.7	67.4	65.0	35.3	13.1	12.7	247.0
rtdetr-SwinTransformer	66.4	64.6	63.4	64.0	98.1	36.6	26.8	129.7
YOLOv8n	68.3	71.4	73.9	69.0	8.1	3.0	6.3	1916.8
YOLO v5n	68.3	65.3	71.2	66.0	7.1	2.5	5.3	1920.9
YOLO v10m	74.2	63.6	73.3	67.0	58.9	15.3	33.5	444.9
RT-DETR-r18	75.6	72.6	72.5	73.0	57.0	19.8	48.5	344.5

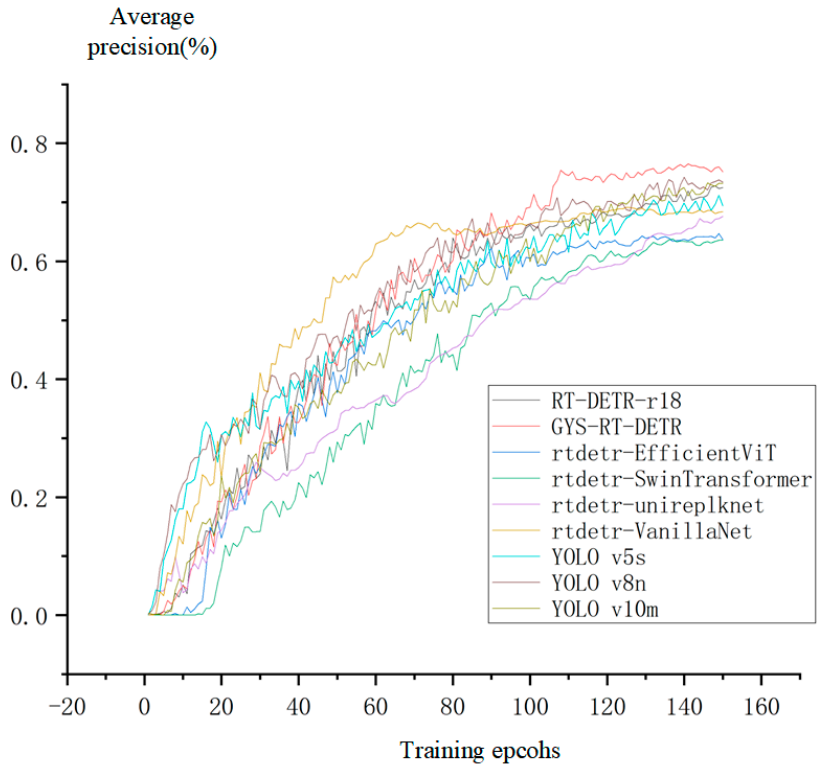


Figure 10. Comparison results of training curves for different models.

4.5. Global Contextual Information Utilization Capability Analysis

Gradient-weighted Class Activation Mapping (GradCAM) is a tool for visualizing the decision-making process of deep learning models, which helps to understand the decision-making basis of the model by generating heat maps to show the key areas of the image that the model focuses on during classification. The red areas in the heatmap indicate regions of higher importance, which correspond to the parts of the image that the model considers more relevant for its decision-making process. In contrast, the blue areas represent regions of lower importance. In order to verify the detection ability of the GYS-RT-DETR model, two challenging images were selected for analysis. As shown in Figure 11, comparing the heat maps of Figure 11b,c, it can be found that the red areas in Figure 11b are more widely distributed, indicating that the GYS-RT-DETR model can capture more characteristic information and thus more precisely determine the type of citrus disease. The results show that the proposed structure can significantly improve the multi-scale feature fusion ability of the model, so as to make full use of the global context information.

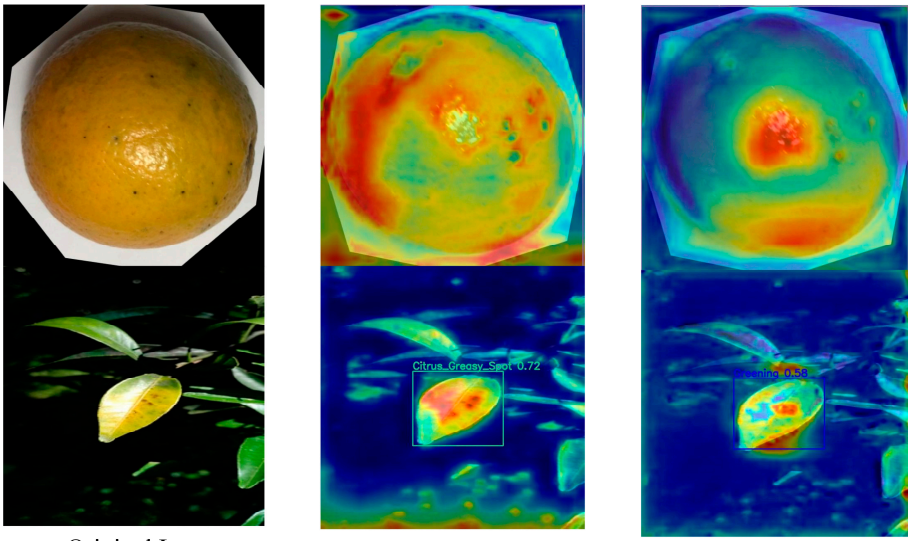


Figure 11. Visual comparison results of GYS-RT-DETR and RT-DETR-R18 characteristic heatmaps ((**a**) on the left is the original image, (**b**) in the middle is the heatmap after GYS-RT-DETR model detection, and (**c**) on the right is the heatmap after RT-DETR-R18 model detection).

4.6. Comparison of Accuracy in Identifying Different Diseases

By comparing the average accuracy of the twelve types of diseases in Figure 12, it can be seen that the GYS-RT-DETR model has achieved significant improvement in the detection of citrus diseases compared with the RT-DETR-r18 model, and most of its average precision has been improved. Although the training samples for citrus brown spot and orange scab were only 63 and 72, respectively, which were far less than the sample size of other diseases, the mean average precision of the GYS-RT-DETR model for these two diseases was 9% and 13.2% higher than that of the original RT-DETR-r18 model, respectively. This significant improvement fully proves that the GYS-RT-DETR model has a strong ability to solve the problem of sample size imbalance.

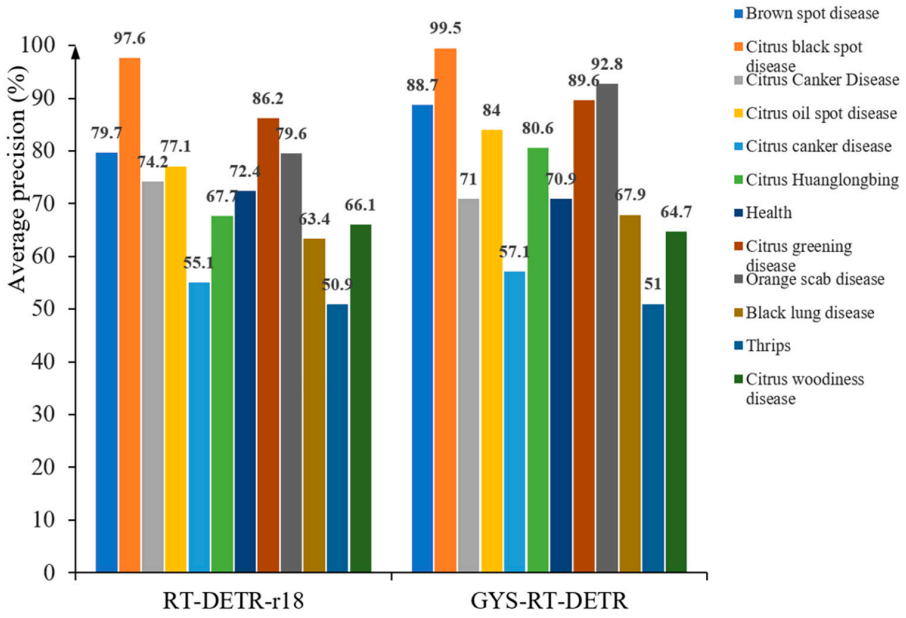


Figure 12. Comparison results of average precision of different citrus diseases (the (**left**) shows the average precision of RT-DETR-R18 for different diseases, and the (**right**) shows the average precision of GYS-RT-DETR for different diseases).

4.7. Comparison of Receptive Fields with Different Network Feature Extraction Capabilities

In order to clearly demonstrate the receptive field sizes for feature extraction of small, medium, and large targets in the Neck network of different network models, Figure 13 visualizes the feature receptive fields of the P3, P4, and P5 feature maps in different networks.

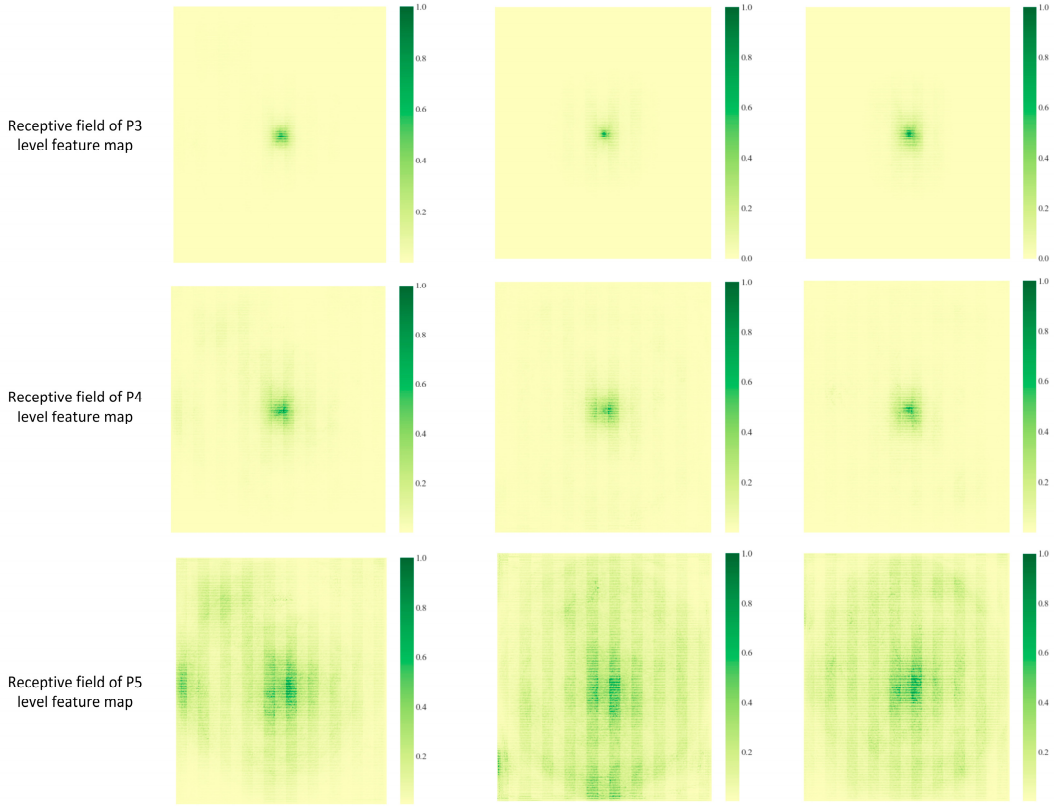


Figure 13. Comparison of receptive fields of different models (the first column in the figure shows the visualization results of the receptive field of RT-DETR-R18, the second column shows the visualization results of the receptive field of RT-DETR-R18+GD, and the third column shows the visualization results of the receptive field of RT-DETR-R18+GD+SFF, where the first row shows the receptive field of the P3 level feature map, the second row shows the receptive field of the P4 level feature map, and the third row shows the receptive field of the P5 level feature map).

By observing the first row in Figure 13, it can be seen that the receptive field of feature extraction for small targets is significantly reduced after the introduction of GD mechanism in the RT-DETR-r18 network, but after the introduction of SSFF mechanism in the Neck network, the receptive field of feature extraction for small targets is expanded. The results show that the receptive field of the P3 feature map and the small target feature information contained in the feature map are effectively increased by introducing the SSFF mechanism.

By observing the second and third rows in Figure 13, it can be seen that the receptive field of feature extraction for large and medium-sized targets has been significantly expanded after the introduction of the GD mechanism of the original model RT-DETR-r18. This result shows that after the GD mechanism is introduced into the Neck network, the feature extraction ability of P4 and P5 level feature maps for medium and large targets is effectively improved, and the information richness is significantly increased compared with the original model. Since the SSFF network module is only used to solve the loss of GD detection performance for small targets, it can be seen from the third column in Figure 13 that the RT-DETR-r18 GD SSFF network and RT-DETR-r18 GD network do not change much from the feature extraction receptive field for large and medium targets.

Through the above analysis, it can be seen that after the introduction of the GD mechanism, the detection ability of the Neck network for medium and large targets has been effectively improved, but its detection performance for small targets has decreased. The introduction of the SSFF mechanism can effectively make up for the shortcomings caused by the GD mechanism. Therefore, by analyzing the receptive field visualization of the P3, P4 and P5 level feature maps of the Neck network, it can be seen that the detection performance of the GYS-RT-DETR network model has been comprehensively improved compared with the detection performance of the original model.

4.8. Experiments with Different Pruning Rates Based on the Group_Taylor Pruning Strategy

In order to achieve model compression and reduce the memory usage and computational complexity of the model, this paper uses different pruning rates to prune the model based on Group_taylor pruning strategies, and the experimental results are analyzed as follows:

As shown in Table 9, by comparing Experiment 1 with Experiment 0, at a pruning rate of 66.7%, the mAP0.5 and mAP0.5:0.9 values of the pruned model show significant improvement, and both the computational parameters and model size have significantly decreased. Comparing the data from Experiment 1 and Experiment 2 shows that when the pruning rate is reduced to 55.6%, the mAP0.5 and mAP0.5:0.9 values of the model continue to improve, and at this point, the computational parameters and model size have also decreased compared to the previous one, which indicating that the model still has the potential to increase the pruning rate. Observing the results of Experiment 3 shows that when the pruning rate is reduced to 50.0%, the mAP value of the pruned model only experiences a slight decline, suggesting that under the current conditions, the pruning rate may be close to or has reached an optimal balance point, and further pruning may lead to a decrease in model performance. Since the decline in mAP value in Experiment 3 compared to Experiment 2 is very small, it is determined that a pruning rate of 55.6% is the optimal pruned model.

Table 9. Experimental results of different pruning rates based on the Group_taylor pruning strategy. ("×" indicates that no pruning rate was used in this experiment).

Experiment Serial Number	Model	Params/M	FLOPs/G	Model Size/MB	$mAP_{0.5}$/%	$mAP_{0.5:0.95}$/%	FPS (f/s)	Pruning Rate/%
0	GYS-RT-DETR	22.5	63.2	43.8	76.5	49.8	239.5	×
1	Group_taylor exp1	15.1 (67.1%)	39.6 (62.7%)	31.3 (71.5%)	77.9 (+1.4)	51.3 (+1.5)	294.9 (+55.4)	66.7
2	Group_taylor exp2	13.3 (59.1%)	34.2 (54.1%)	27.6 (63.0%)	78.2 (+1.7)	51.4 (+1.6)	327.6 (+88.1)	55.6
3	Group_taylor exp3	11.0 (48.9%)	27.7 (43.8%)	23.0 (52.5%)	75.7 (−0.8)	49.4 (−0.4)	354.8 (+115.3)	50.0

To further clarify this point, we have plotted the Pareto frontier of mAP and FPS at different pruning rates. As shown in Figure 14, the Pareto frontier demonstrates the trade-off between mAP values and FPS at various pruning rates. The optimized pruning rate of 55.6% has been determined as the point at which the model achieves the highest mAP while maintaining a high FPS. In contrast, although a 50% pruning rate can yield the highest FPS, it results in a significant decrease in mAP, which also highlights the importance of avoiding excessive pruning.

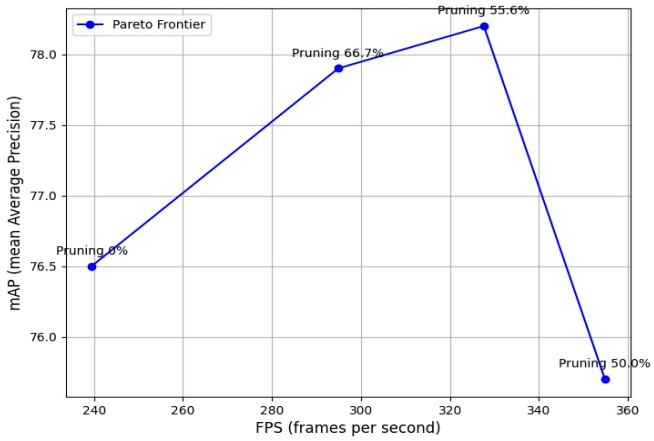


Figure 14. Pareto frontier of mAP vs. FPS for different pruning rates.

In order to more intuitively demonstrate the pruning effect of the compression model and the GYS-RT-DETR model on the number of channel parameters under the Group_taylor pruning strategy, the comparison between the two can be revealed in Figure 15. As can be seen from the figure, the optimal pruning model has the most significant reduction in the channel parameters in the left network layer, while the reduction in the number of parameters in the other network layers is relatively small. This phenomenon shows that the optimal pruning model effectively removes the redundant or small contribution channel parameters in the network through the local pruning strategy, rather than simply reducing the number of parameters in all layers.

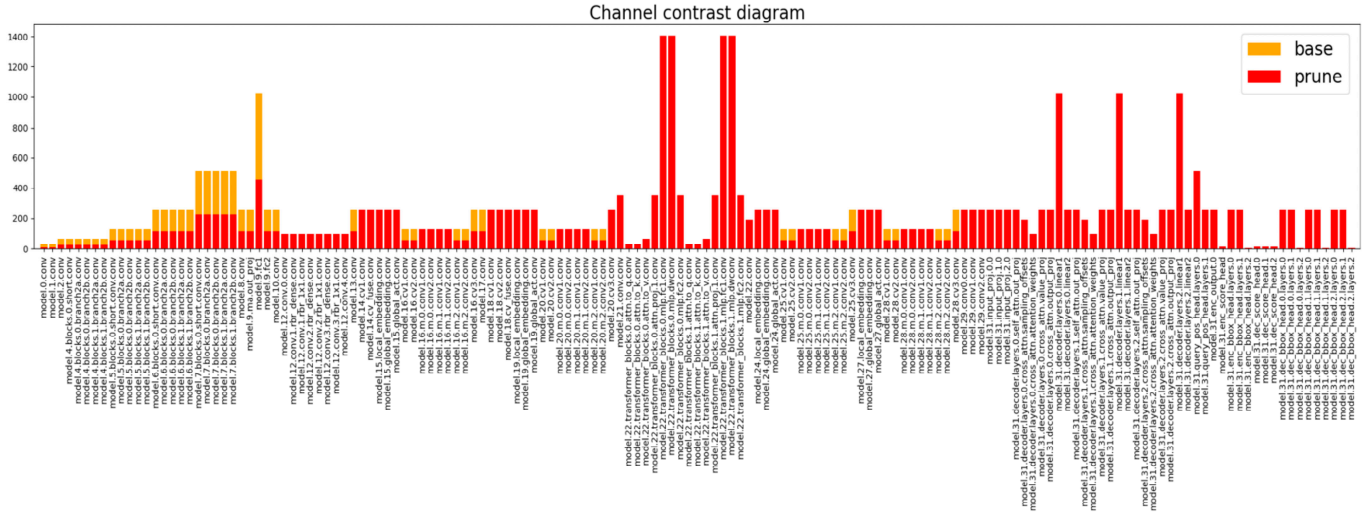


Figure 15. Comparison results of channel parameter quantities between the GYS-RT-DETR model and the pruning model (yellow represents different channel parameter quantities of the GYS-RT-DETR model, and red represents different channel parameter quantities of the pruning model).

4.9. Comparative Analysis of Different Feature Loss Functions

This experiment explores the impact of different feature distillation methods on the detection performance of the student model and the model's lightweighting effects, based on the knowledge distillation framework that combines feature distillation and logic distillation proposed in this paper. The optimal pruned model is selected as the lightweight student model and paired with a teacher model that has more parameters and stronger performance. Experiments are conducted by setting different teacher models, various

distillation evaluation score strategies, and different learning rate decay algorithms for comparative analysis to obtain the optimal student model.

The mainstream algorithms for the loss function of feature distillation include cwd, mimic, mgd, chsim, and sp. However, in this paper, the logical distillation loss function is consistently set to the logical algorithm. The constant parameter settings for different feature distillation methods are shown in Table 10.

Table 10. Distillation constant parameter settings.

Parameter Type	Parameter Value
kd_loss_decay	Linear
kd_loss_epoch	1.0
logical_loss_type	logical
kd_loss_type	all

From Table 11, it can be seen that compared with other loss functions, the characteristic distillation loss function SP has the highest $mAP_{0.5}$ and $mAP_{0.5:0.95}$ values. This indicates that the comprehensive detection performance of the student model based on the SP loss function is the best. In addition, the student model based on the SP loss function has the highest F_1-score compared with other loss functions, which further proves that its comprehensive performance evaluation is the best. Therefore, by comparing the experiments of different eigen-loss functions, it can be concluded that the student model based on the SP loss function has the best comprehensive performance.

Table 11. Comparison results of distillation loss functions for different features of student teacher models.

Student Model	Feature Loss Function	$mAP_{0.5}$/%	$mAP_{0.5:0.95}$/%	FLOPs/G	Params/M	P/%	R/%	F_1/%
Pruning model	x	75.7	49.4	27.7	11.0	75.1	75.5	76.0
	cwd	77.2 (+1.5)	50.8 (+1.4)	27.7	11.0	75.9 (+0.8)	75.9 (+0.4)	77.0 (+1.0)
	mimic	76.4 (+0.7)	50.4 (+1.0)	27.7	11.0	80.6 (+5.5)	72.3 (−3.2)	76.0 (+0.0)
	mgd	76.8 (+1.1)	50.1 (+0.7)	27.7	11.0	80.9 (+5.8)	74.1 (−1.4)	77.0 (+1.0)
	chsim	76.5 (+0.8)	50.3 (+0.9)	27.7	11.0	76.0 (+0.9)	74.4 (−1.1)	75.0 (−1.0)
	sp	77.7 (+2.0)	51.1 (+1.7)	27.7	11.0	78.6 (+3.5)	76.3 (+0.8)	77.0 (+1.0)

4.10. Comparative Analysis of Model Performance Evaluation Indexes at Different Stages

In order to objectively assess the performance differences and improvement effects of the optimization model at various stages, this paper conducts a systematic comparative analysis of the performance metrics data for RT-DETR-r18, GYS-RT-DETR, the pruned model, and the distilled model. Parameter settings for the experimental content in this subsection are all configured according to the parameters in Section 4.1.3.

The comparison results of the models at different stages are shown in Table 12:

Table 12. Comparison results of models at different stages.

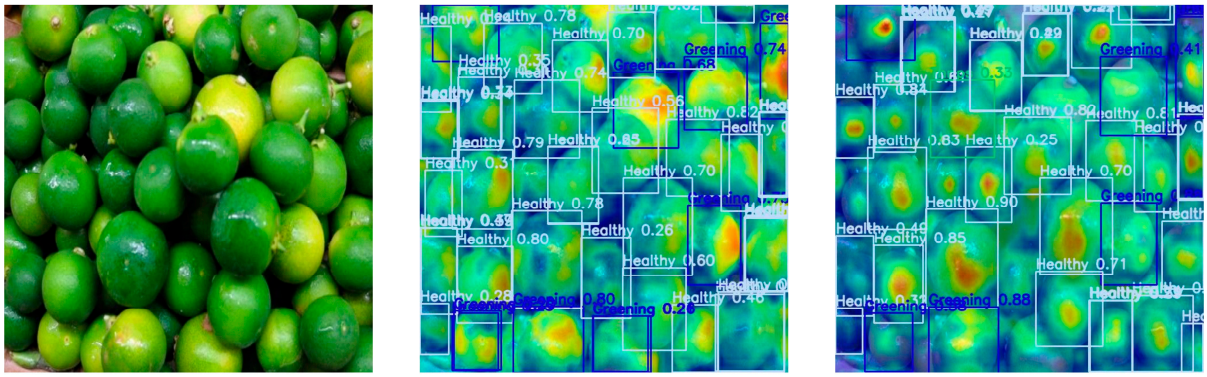
Model	$mAP_{0.5}$/%	$mAP_{0.5:0.95}$/%	FLOPs/G	Params/M	P/%	R/%	F_1/%
RT-DETR-r18	72.5	46.1	57.0	19.8	75.6	72.6	73.0
GYS-RT-DETR	76.5	49.8	63.2	22.5	78.0	73.8	75.0
Pruning model	75.7	49.4	27.7	11.0	75.1	75.5	76.0
Distillation optimization model	77.7	51.1	27.7	11.0	78.6	76.3	77.0

(1) Compared with the original model RT-DETR-r18, the improved GYS-RT-DETR model has an improvement of 4%, 3.7%, 2.4%, 1.2% of F_1-scores, $mAP_{0.5}$, $mAP_{0.5:0.95}$, precision, recall and F_1-score, respectively, which proves that the improved model has stronger generalization ability. At the same time, compared with RT-DETR-r18, the number of parameters and FLOPs of the GYS-RT-DETR model increased by 2.7 M and 6.2 G, respectively, which proved that the model memory occupation and model inference prediction speed of the improved GYS-RT-DETR model decreased significantly. This result shows that although the detection performance of GYS-RT-DETR is better than that of the original RT-DETR-r18 model, its deployment difficulty on edge devices is significantly increased.

(2) After multiple experiments with different channel pruning strategies and pruning rates, the pruned model showed a decrease of 0.8% in $mAP_{0.5}$ and 0.4% in $mAP_{0.5:0.95}$ compared to the GYS-RT-DETR model. This demonstrates that models based on coarse-grained channel pruning strategies often experience a decline in detection performance due to excessively high pruning rates. Although the precision of the pruned model decreased by 2.9% compared to the GYS-RT-DETR model, the former's F_1-score and recall rate increased by 1% and 1.7%, respectively. This improvement in recall rate led to an increase in the F1-score. Finally, since the model pruning algorithm primarily focuses on the model's memory usage and inference prediction speed, a comparison between the pruned model and the GYS-RT-DETR model shows that the FLOPs and parameter count decreased by 35.5 G and 11.5 M, respectively, indicating that the model's computational complexity and memory usage were nearly reduced by 50%. This indicates that the pruned model addresses the deployment challenges of the GYS-RT-DETR model on edge devices, while the overall performance evaluation metrics of the pruned model only show a slight loss compared to the GYS-RT-DETR model.

(3) Compared with the pruning model, the FLOPs and parameters of the student model KDP-GYS-RT-DETR model after feature-logic distillation remain unchanged, which proves that the attention of the KD distillation algorithm mainly focuses on the deep feature learning of the network layer between the student-teacher model, without changing the number of model parameters and the computational complexity of the model. At the same time, the distillation optimization model increased the $mAP_{0.5}$, $mAP_{0.5:0.95}$, precision, recall and F_1-score by 2.0%, 1.7%, 3.5%, 0.8% and 1.0%, respectively, compared with the pruning model. This proves that the feature-logic distillation framework proposed in this paper makes the pruning model effectively learn the deep features and output decision-making ability of the RT-DETR-r50 model.

Previous analysis shows that the improvement strategies at each stage can specifically enhance the model's performance. To further verify whether the detection performance gap between the distilled optimized model and the RT-DETR-r50 model in real scenarios has narrowed or surpassed that of the teacher model, the comparison of Figure 16b,c reveals that the teacher model has instances of missed detections, while the student model does not have any missed detections. This result indicates that the student model, after distillation optimization, outperforms the teacher model RT-DETR-r50 in terms of disease recognition performance in real scenarios.



a.Original image b.Student model detection thermal map c.Teacher model detection thermal map

Figure 16. Thermal map results of student model and teacher model in actual scene recognition. (**a**) is the original image, (**b**) is the thermal map detection result of the student model, and (**c**) is the thermal map detection result of the teacher model).

5. Discussion

(1) Network Improvement: After improvements to the network structure and loss function, the GYS-RT-DETR model shows an increase of 4%, 3.7%, 2.4%, 1.2%, and 2% in $mAP_{0.5}$, $mAP_{0.5:0.95}$, precision, recall, and F_1-score, respectively, compared to the original model RT-DETR-r18. At the same time, the parameter count and FLOPs of the GYS-RT-DETR model increased by 2.7 M and 6.2 G, respectively, compared to RT-DETR-r18, indicating that the memory usage and inference prediction speed of the GYS-RT-DETR improved model have significantly decreased. This result suggests that the increase in computational parameters for GYS-RT-DETR is not conducive to the model's lightweight goals.

(2) Model Pruning: The pruned model obtained using the Group_taylor pruning strategy shows a reduction of 35.5 G in FLOPs and 11.5 M in parameter count compared to the GYS-RT-DETR model, resulting in nearly a 50% decrease in computational complexity and memory usage. This result demonstrates that the pruned model addresses the deployment challenges of the GYS-RT-DETR model on edge devices, while the overall performance evaluation parameters of the pruned model only incur a slight loss compared to the GYS-RT-DETR model.

(3) Knowledge Distillation: This article optimizes lightweight pruned models through a feature-logic distillation framework. Compared to the pruned models before distillation, the pruned models after distillation show improvements of 2.0% in $mAP_{0.5}$, 1.7% in $mAP_{0.5:0.95}$, 3.5% in precision, 0.8% in recall, and 1.0% in F_1-score. Additionally, the knowledge distillation framework proposed in this article shows differences in the feature learning network layers between the student model and the teacher model. This demonstrates that the feature-logic distillation framework enables the pruned model to effectively learn the deep features and output decision-making capabilities of the RT-DETR-r50 model.

(4) Comparison with Other Studies: Compared with the ASF-YOLO model proposed by Kang et al. [18], the GYS-RT-DETR model, with the addition of the GD mechanism, outperforms ASF-YOLO in the detection capability of medium and large targets. When compared with the Gold-YOLO model proposed by Wang et al. [17], both Gold-YOLO and GYS-RT-DETR enhance the detection of multi-scale targets through the introduction of the global feature fusion mechanism (GD). However, GYS-RT-DETR further improves the detection performance of small targets by incorporating the Scale

Sequence Feature Fusion (SSFF) mechanism. Compared with the Focaler-IoU method proposed in reference [20], the GYS-RT-DETR model adopts a different strategy to address the issues of sample imbalance and localization accuracy in disease detection tasks. It introduces the Focaler-ShapeIoU loss function, which combines shape and scale information to optimize the detection capability for occluded and small targets. Moreover, the GYS-RT-DETR model achieves model lightweighting and is more suitable for deployment on edge devices compared to the two aforementioned models.

6. Conclusions

In summary, the precision, recall, F_1-score, model size, and mAP value of the GYS-RT-DETR model were 79.1%, 77.9%, 78.0%, 23.0 MB, and 77.8%, respectively. Compared with the original model, the accuracy, recall, F_1-score, mAP value, and FPS value are increased by 3.5%, 5.3%, 5.0%, 5.3%, and 10.3 f/s, respectively. At the same time, the memory usage of the GYS-RT-DETR model is reduced by 25.5 MB compared with the original model. This result shows that the GYS-RT-DETR model proposed in this paper is optimized compared with the original model in terms of recognition performance and light weight of the model. These indicators are sufficient to provide reliable citrus detection and diagnostic results, while also meeting the demand for low computational resource usage. The GYS-RT-DETR model is designed to address the real-world challenges faced by citrus farmers and agricultural practitioners. The complete implementation details of the GYS-RT-DETR model, including its architecture and optimization strategies, are provided in Appendix A. Finally, in future work, the richness of datasets will be expanded, and the disease detection models will be continuously optimized in close integration with mobile apps, edge devices, and drone deployments to enhance actual detection capabilities. We will also strengthen our research on the issue of multiple diseases that may affect a single citrus fruit. As a result, farmers will be able to obtain real-time disease detection results. Additionally, we plan to conduct field trials to verify the effectiveness of the models in real agricultural scenarios.

Author Contributions: L.Y.: Writing—review and editing, Writing—original draft, Methodology, Funding acquisition, Conceptualization. Z.H.: Writing—review and editing, Writing—original draft, Validation, Methodology, Conceptualization, Software. Y.H.: Writing—review and editing, Validation, Software, Resources. R.L.: Writing—review and editing, Data Curation. X.W.: Writing—review and editing, Data Curation. Z.P.: Writing—review and editing, Data Curation. J.S.: Writing—review and editing, Validation, Supervision, Investigation. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the China Central Fund for Guiding Development of Local Science and Technology under Grant No. 202407AB110010, in part by the Yunnan Science and Technology Major Project under Grant No. 202302AE090020, in part by Yunnan Science and Technology Major Project under Grant No. 202303AP140014.

Data Availability Statement: To ensure the transparency and reproducibility of our research, the dataset used in this study is now publicly available. The dataset includes all the images and related information used for the analyses. It can be accessed via the following link: https://www.kaggle.com/datasets/jxgahs/citrus-disease-dataset159 (accessed on 14 June 2025). The data may be freely used, distributed, and reproduced under the condition of appropriate attribution.

Conflicts of Interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A

Below is the complete code of the GYS-RT-DETR model

```
# Parameters
nc: 80  # number of classes
scales: # model compound scaling constants, i.e. 'model=yolov8n-cls.yaml' will call
yolov8-cls.yaml with scale 'n'
  # [depth, width, max_channels]
  l: [1.00, 1.00, 1024]

backbone:
  # [from, repeats, module, args]
  - [-1, 1, ConvNormLayer, [32, 3, 2, None, False, 'relu']] # 0-P1/2
  - [-1, 1, ConvNormLayer, [32, 3, 1, None, False, 'relu']] # 1
  - [-1, 1, ConvNormLayer, [64, 3, 1, None, False, 'relu']] # 2
  - [-1, 1, nn.MaxPool2d, [3, 2, 1]] # 3-P2/4

  # [ch_out, block_type, block_nums, stage_num, act, variant]
  - [-1, 1, Blocks, [64, BasicBlock, 2, 2, 'relu']] # 4
  - [-1, 1, Blocks, [128, BasicBlock, 2, 3, 'relu']] # 5-P3/8
  - [-1, 1, Blocks, [256, BasicBlock, 2, 4, 'relu']] # 6-P4/16
  - [-1, 1, Blocks, [512, BasicBlock, 2, 5, 'relu']] # 7-P5/32

head:
  - [-1, 1, Conv, [256, 1, 1, None, 1, 1, False]]  # 8 input_proj.2
  - [-1, 1, AIFI, [1024, 8]] # 9
  - [-1, 1, Conv, [256, 1, 1]]  # 10, Y5, lateral_convs.0

  - [[4, 5, 6, 7], 1, SimFusion_4in, []] # 11
  - [-1, 1, IFM, [[64, 32]]] # 12

  - [10, 1, Conv, [256, 1, 1]] # 13
  - [[5, 6, -1], 1, SimFusion_3in, [256]] # 14
  - [[-1, 12], 1, InjectionMultiSum_Auto_pool, [256, [64, 32], 0]] # 15
  - [-1, 3, RepC3, [256, 0.5]] # 16

  - [6, 1, Conv, [256, 1, 1]] # 17
  - [[4, 5, -1], 1, SimFusion_3in, [256]] # 18
  - [[-1, 12], 1, InjectionMultiSum_Auto_pool, [256, [64, 32], 1]] # 19
  - [-1, 3, RepC3, [256, 0.5]] # 20

  - [[20, 16, 10], 1, PyramidPoolAgg, [352, 2]] # 21
  - [-1, 1, TopBasicLayer, [352, [64, 128]]] # 22

  - [[20, 17], 1, AdvPoolFusion, []] # 23
  - [[-1, 22], 1, InjectionMultiSum_Auto_pool, [256, [64, 128], 0]] # 24
  - [-1, 3, RepC3, [256, 0.5]] # 25
```

```
- [[-1, 13], 1, AdvPoolFusion, []] # 26
- [[-1, 22], 1, InjectionMultiSum_Auto_pool, [256, [64, 128], 1]] # 27
- [-1, 3, RepC3, [256, 0.5]] # 28

- [[5, 6, 8], 1, ScalSeq, [256]] # 29
- [[20, -1], 1, Add, []] # 30
# - [[20, -1], 1, asf_attention_model, []] # 30

- [[30, 25, 28], 1, RTDETRDecoder, [nc, 256, 300, 4, 8, 3]] # 29
```

References

1. Deng, X. A Review and Perspective for Citrus Breeding in China During the Last SixDecades. *Acta Hortic. Sin.* **2022**, *49*, 2063–2074. [CrossRef]
2. Zhao, J.; Liu, Y.; Feng, Z.; Li, R.; Li, M.; Li, Y. Analysis of Competitive Advantages in the Development of the Citrus Industry in Yunnan Province. *Yunnan Agric.* **2018**, 40–41.
3. Nie, R.; Sun, M.; Feng, D.; Song, X.; Gao, J.; Guo, L.; Deng, X.; Chai, L.; Xie, Z.; Ye, J. An Investigation on Occurrence and Distribution Patterns of Five Common Citrus Diseases in Yunnan. *Acta Hortic. Sin.* **2024**, *51*, 2685–2700. [CrossRef]
4. Tan, B. Research on Citrus Disease Identification Based on Convolutional Neural Networks. Master's Thesis, Jiangsu University, Zhenjiang, China, 2023.
5. Zeng, W.; Chen, Y.; Hu, G.; Bao, W.; Liang, D. Detection of citrus Huanglongbing in natural background by SMS and two-way feature fusion. *Trans. Chin. Soc. Agric. Mach.* **2022**, *53*, 1.
6. He, C. Rapid Detection Citrus HLB by Developing a Handheld Device Based on Spectral Imaging Technology. Master's Thesis, Fujian Agriculture and Forestry University, Fuzhou, China, 2022.
7. Dai, Z. Research on Citrus Huanglongbing Diagnosis System Based on Edge Computing. Master's Thesis, Southwestern University, Georgetown, TX, USA, 2022.
8. Lian, B. Research on Online Diagnostic Technology and System of Citrus Huanglongbing Based on MobileNet. Master's Thesis, South China Agricultural University, Guangzhou, China, 2019.
9. Liu, Y.; Xiao, H.; Sun, X.; Zhu, D.; Han, R.; Ye, L.; Wang, J.; Ma, K. Spectral feature selection and discriminant model building for citrus leaf Huanglongbing. *Trans. Chin. Soc. Agric. Eng.* **2018**, *34*, 180–187.
10. Qing, D. Rapid Nondestructive Detection of Citrus Greening (HLB) Using Hyperspectral Imaging Technology. Ph.D. Dissertation, East China Jiaotong University, Nanchang, China, 2016.
11. Bové, J.; Chau, N.M.; Trung, H.M.; Bourdeaut, J.; Garnier, M. Huanglongbing (greening) in Vietnam: Detection of Liberobacter asiaticum by DNA-hybridization with probe in 2.6 and PCR-amplification of 16S ribosomal DNA. In Proceedings of the International Organization of Citrus Virologists Conference Proceedings (1957–2010), Fuzhou, China, 16–23 November 1996.
12. Weng, H.; Liu, Y.; Captoline, I.; Li, X.; Ye, D.; Wu, R. Citrus Huanglongbing detection based on polyphasic chlorophyll a fluorescence coupled with machine learning and model transfer in two citrus cultivars. *Comput. Electron. Agric.* **2021**, *187*, 106289. [CrossRef]
13. He, C.; Li, X.; Liu, Y.; Yang, B.; Wu, Z.; Tan, S.; Ye, D.; Weng, H. Combining multicolor fluorescence imaging with multispectral reflectance imaging for rapid citrus Huanglongbing detection based on lightweight convolutional neural network using a handheld device. *Comput. Electron. Agric.* **2022**, *194*, 106808. [CrossRef]
14. Zhao, Y.; Lv, W.; Xu, S.; Wei, J.; Wang, G.; Dang, Q.; Liu, Y.; Chen, J. Detrs beat yolos on real-time object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 17–21 June 2024; pp. 16965–16974.
15. Linlin, L.; Hailong, S. Forest Pedestrian Detection Based on Improved YOLOv8. *For. Eng.* **2025**, *41*, 138–150.
16. Molchanov, P.; Tyree, S.; Karras, T.; Aila, T.; Kautz, J. Pruning convolutional neural networks for resource efficient transfer learning. *arXiv* **2016**, arXiv:1611.06440. [CrossRef]
17. Wang, C.; He, W.; Nie, Y.; Guo, J.; Liu, C.; Wang, Y.; Han, K. Gold-YOLO: Efficient object detector via gather-and-distribute mechanism. *Adv. Neural Inf. Process. Syst.* **2023**, *36*, 51094–51112.
18. Kang, M.; Ting, C.-M.; Ting, F.F.; Phan, R. ASF-YOLO: A novel YOLO model with attentional scale sequence fusion for cell instance segmentation. *Image Vis. Comput.* **2024**, *147*, 105057. [CrossRef]

19. Zhang, H.; Zhang, S. Shape-iou: More accurate metric considering bounding box shape and scale. *arXiv* **2023**, arXiv:2312.17663.
20. Zhang, H.; Zhang, S. Focaler-iou: More focused intersection over union loss. *arXiv* **2024**, arXiv:2401.10525.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.