agronomy

Article

Real-Time Lightweight Detection of Lychee Diseases with Enhanced YOLOv7 and Edge Computing

Jiayi Xiao [1], Gaobi Kang [1], Linhui Wang [2], Yongda Lin [3], Fanguo Zeng [1], Jianyu Zheng [1], Rong Zhang [1] and Xuejun Yue [1,*]

[1] College of Electronic Engineering (College of Artificial Intelligence), South China Agricultural University, Guangzhou 510642, China; 2021xiao@stu.scau.edu.cn (J.X.); kanggb@stu.scau.edu.cn (G.K.); tsvanco@stu.scau.edu.cn (F.Z.); 20223172035@stu.scau.edu.cn (J.Z.); r-zhang@scau.edu.cn (R.Z.)
[2] School of Intelligent Manufacturing, Hunan University of Science and Engineering, Yongzhou 425100, China; wlh3362@huse.edu.cn
[3] College of Science, China Agricultural University, Beijing 100193, China; b20233100855@cau.edu.cn
* Correspondence: yuexuejun@scau.edu.cn

Abstract: Lychee is an economically important crop with widespread popularity. However, lychee diseases significantly impact both the yield and fruit quality of lychee. Existing lychee disease detection models face challenges such as large parameter sizes, slow processing speeds, and deployment complexities. To address these challenges, this paper proposes an improved lightweight network, named YOLOv7-MGPC (YOLOv7-Mosaic-GhostNet-Pruning-CBAM), that enables real-time lychee disease detection. In this study, we collected datasets of lychee diseases, covering four types of leaf diseases, and employed Mosaic data augmentation for data preprocessing. Building upon the YOLOv7 framework, we replaced the original backbone network with the lightweight GhostNetV1 and applied channel pruning to effectively reduce the parameter overhead. Subsequently, an attention mechanism called CBAM was incorporated to enhance the detection accuracy. The resultant model was then deployed to edge devices (Nvidia Jetson Nano) for real-world applications. Our experiments showed that our enhanced YOLOv7 variant outperforms the original model by a large margin, achieving a speed increase from 120 frames/s to 217 frames/s while maintaining an accuracy of 88.6%. Furthermore, the parameter size was substantially reduced from 36.5 M to 7.8 M, which firmly demonstrates the effectiveness of our methods in enabling model deployment on edge devices for lychee disease detection.

Keywords: YOLOv7; GhostNetV1; channel pruning; disease identification; edge computing

check for updates

Citation: Xiao, J.; Kang, G.; Wang, L.; Lin, Y.; Zeng, F.; Zheng, J.; Zhang, R.; Yue, X. Real-Time Lightweight Detection of Lychee Diseases with Enhanced YOLOv7 and Edge Computing. *Agronomy* **2023**, *13*, 2866. https://doi.org/10.3390/agronomy13122866

Academic Editors: Imre J. Holb, Han Li and Ruicheng Qiu

Received: 7 September 2023
Revised: 28 October 2023
Accepted: 8 November 2023
Published: 22 November 2023

1. Introduction

Lychee (Litchi chinensis) is a tropical fruit tree belonging to the Sapindaceae family, known for its sweet and fragrant flavor [1]. China is the homeland of lychee and ranks first globally in terms of lychee cultivation area, production, and value, with 475,700 hectares of lychee cultivated area present in the country [2]. The development of lychee cultivation is crucial for improving economic benefits and meeting the market's demand for high-quality lychee, as it is one of China's most important economic crops. However, diseases on lychee leaves during the growth stage can significantly reduce both yield and quality, hindering the stable development of the lychee industry [3]. Therefore, the timely detection and early feedback of diseases on lychee leaves are key to resolving these issues and ensuring optimal lychee yields.

Traditional methods for the detection of diseases mainly rely on manual inspection, which is laborious, time-consuming, subjective, and highly damaging. Moreover, many growers lack sufficient scientific knowledge of these diseases, leading to the excessive and indiscriminate use of pesticides that delay optimal disease prevention and greatly reduce lychee quality.

Advances in computer technology have led researchers to employ machine learning and image processing methods for crop pest and disease detection. Deep learning techniques, in particular, have significantly contributed to the cultivation and increased yields of agricultural products [4]. Object detection algorithms based on deep learning [5] can rapidly and accurately recognize the target category and locate its position in input images, providing an effective method for the non-destructive identification of crop pests and diseases [6–8].

Recent research has been focused on improving existing deep learning models to enhance the accuracy and efficiency of crop pests and disease detection. For example, researchers have modified the YOLOV4 model for tomato pest detection [9], combined the YOLOV5 model with dual Feature Pyramid Networks (FPN) to improve feature representations [10], and incorporated GhostNetV1 into the SSD model to accelerate detection speed [11]. The integration of modern computer vision technology with cloud computing and edge computing has also provided increased flexibility for efficiently managing agricultural tasks [12]. Yang et al. combined autonomous scouting and lodged rice detection with edge computing [13]. Ajayi et al. introduced a sophisticated automated crop and weed classification system based on the YOLOv5 and UAV imagery [14]. These innovative approaches demonstrate the remarkable progress in this field and pave the way for more effective and efficient agricultural operations.

Despite the progress in this field, current detection algorithms still face challenges such as large network parameters and memory consumption, resulting in slow execution speeds and reduced efficiency. These limitations make them unsuitable for edge deployment and hinder their practical applications. Additionally, the large model parameter size makes it difficult to meet the lightweight requirements of mobile devices.

To address these issues, we introduce an improved lightweight lychee disease real-time detection model based on YOLOv7. Our model is deployed on lightweight edge devices for real-time detection, meeting the requirements of lightweight and easy deployment for crop disease detection. The main contributions of this study are as follows:

(1) We enhance the diversity of data through the mosaic augmentation technique, improving the generalization capability of the detection model;
(2) We replace the backbone network with GhostNetV1 and prune the redundant feature layers using channel pruning techniques to reduce computational overhead;
(3) We introduce the CBAM (Convolutional Block Attention Module) mechanism to make the model focus on crucial features, enhancing the model's feature extraction capabilities;
(4) We deploy the enhanced lightweight model on edge devices and verify that the model meets the deployment requirements within a computationally limited environment.

2. Related Works

2.1. Lychee Disease Detection with Deep Learning

In recent years, significant progress has been made in the intelligent detection of lychee diseases based on deep learning. Islam et al. utilized pre-trained Convolutional Neural Networks (CNNs) and a transfer learning-based approach to classify three major categories, namely "leaf necrosis", "leaf spot", and "stem canker diseases", from lychee-diseased leaf and stem images [15]. Xie et al. proposed an enhanced Fully Convolutional One-Stage Object Detection (FCOS) network, successfully identifying five common lychee leaf diseases in different orchards [16]. These diseases included lychee leaf mites, lychee downy mildew, lychee anthracnose, lychee leaf blotch, and lychee algal spot diseases. The detection network achieved a detection accuracy of 91.3% (IoU = 0.5), detection speed of 62.0 frames/s, and model parameter size of 17.65 million. Mahmud et al. conducted a comparative analysis of the VGG16, Inception-V3, and Xception algorithms for lychee disease detection and applied the Inception-V3 algorithm, which achieved an impressive accuracy of 92.67% [17]. Wang et al. investigated three deep Convolutional Neural Network (DCNN) models, namely SSD-MobileNetV2, Faster RCNN-ResNet50, and Faster RCNN-

Inception-ResNetV2, to explore the feasibility of the automated detection of defective surfaces in lychees [18].

While previous studies have achieved satisfactory results from employing deep learning models for lychee disease detection, the deployment of edge devices for lightweight lychee disease detection has not been extensively explored.

2.2. Lightweight Network Research

Currently, object detection algorithms for lychee diseases require significant computational resources and time due to their large model parameters and memory usage. This affects the detection speed and hinders the real-time monitoring of diseases, leaving room for improvement in terms of real-time monitoring. Model pruning and employing lightweight feature extraction networks are common methods for model slimming, and extensive research has been conducted in this area.

Some classic models of lightweight Convolutional Neural Networks include MobileNet [19], GhostNet [20], and ShuffleNet [21]. The GhostNet model is specifically designed for hardware deployment and mobile devices. Han et al. conducted further validation by assessing the actual performance of GhostNet on an ARM-based mobile phone. The experiments demonstrate that GhostNet achieves about 0.5% higher accuracy than MobileNetV3 at the same latency, and it requires less runtime to achieve comparable performance [20]. Our research will further investigate and compare the strengths and weaknesses of these networks in our experiments.

Model pruning is another method for network lightweighting. It involves removing redundant parameters in deep learning network layers to reduce the spatial footprint of the network model and eliminate redundant computations. Shi et al. employed a layer-wise pruning approach in the channel dimension, selectively removing irrelevant convolutional kernels before performing fine-tuning [22]. Notably, this pruning technique resulted in a 68.7% reduction in network computational load, accompanied by a 0.4% improvement in accuracy.

Previous experiments have shown that replacing networks with lightweight counterparts can effectively reduce the model parameter count and enhance speed. These model pruning techniques contribute to the advancement in lightweight networks, making them more suitable for real-time applications.

2.3. Research into the Attention Mechanism

Generally, model pruning can lead to improved speed but decreased accuracy. Hence, the incorporation of attention mechanisms becomes pivotal to restoring high model performance. In typical situations, we perceive our surroundings by focusing on the key parts of a scene rather than its secondary components. Similarly, attention plays a similar role in deep learning architectures and has been widely applied to various tasks, including image classification [23], object detection [24], and scene segmentation [25]. The attention mechanism emphasizes the critical regions in the input feature layers, enhancing the model's feature extraction capability.

Xie et al. proposed a lightweight YOLOv4 model that integrated the ECA attention mechanism with the Mobilenetv2 backbone for defect detection on metal surfaces [26]. Experimental results showed that the algorithm's accuracy significantly surpassed algorithms incorporating the SE attention mechanism. In addition to channel-specific SE-NET and ECA attention mechanisms, there is a combined channel and spatial CBAM attention mechanism. Li et al. introduced an improved YOLOv5 algorithm for wheat spike detection and incorporated the CBAM attention mechanism into the network to address the problem of gradient vanishing during training and enhance the model's feature extraction capability [27]. In lightweight models, pruning operations may result in a decline in model detection accuracy. Therefore, introducing attention mechanisms can partially improve feature extraction capability and compensate for the decline in accuracy.

3. Materials and Methods

This paper proposes a method for detecting lychee leaf diseases that consists of three main steps: data collection and augmentation, model pruning and optimization, and deployment, as illustrated in Figure 1.

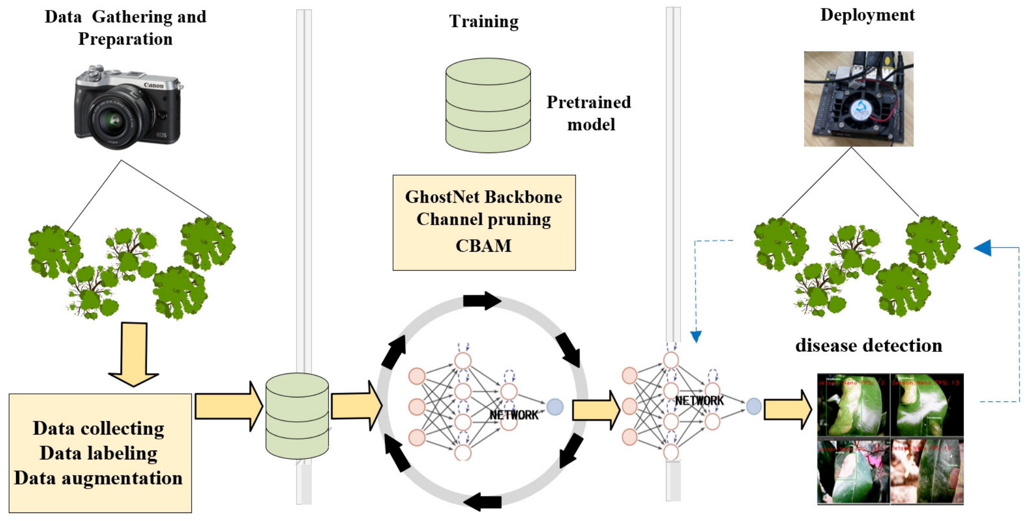


Figure 1. Pipeline of the lychee detection system.

The research methodology began with the collection and preprocessing of lychee orchard image data. The introduction of the Mosaic data augmentation technique aided the construction of a comprehensive dataset for lychee leaf diseases. Subsequently, a lightweight model based on YOLOv7 was developed. This entailed substituting the original backbone network with the lightweight GhostNetV1 for efficient feature extraction. Further parameter reduction was achieved by applying a sparse scaling factor γ and L1 regularization in Batch Normalization (BN) layers to prune the redundant model parameters. Additionally, the model's feature extraction capabilities were enhanced through the integration of the attention mechanism CBAM, resulting in improved accuracy. Finally, the refined model was deployed to edge devices for practical applications.

3.1. Data Acquisition

The lychee disease data were collected from the tree orchard of the College of Horticulture, South China Agricultural University, Guangzhou, China. The data collection took place from 2 March to 28 May 2023 during the time intervals of 8:30 AM to 11:30 AM and 2:30 PM to 5:30 PM. The data were captured using a high-resolution camera, and the distance between the image capturing device and the diseased leaves ranged from 0.25 m to 0.55 m. The camera used in the experiment and the test site are shown in Figure 2a,b. Figure 2c shows our method of capturing lychee disease images.

(**a**) Canon EOS M6 (**b**) Lychee orchard (**c**) Image capturing

Figure 2. Data collection.

This research focuses on four types of lychee diseases, including felt disease, leaf gall, bituminous, and anthrax, which are characterized by physiological features exhibited on the leaves under the influence of pest damage. The image resolution for each capture was set at 3024 pixels × 4032 pixels. In total, 1273 high-resolution images of lychee diseases were collected for this study, as illustrated in Figure 3.

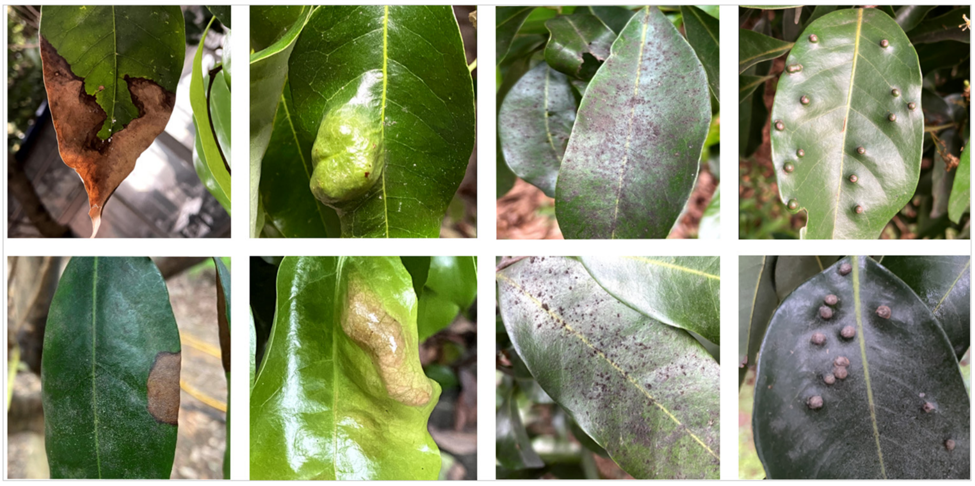

(**a**) Anthrax (**b**) Felt diseases (**c**) Bituminous (**d**) Leaf gall

Figure 3. Image of lychee diseases.

3.2. Data Preprocessing

To improve the robustness and accuracy of the disease detection model, a dataset of 1273 lychee disease images was utilized in this study. The dataset was then divided into a training set, testing set, and validation set with a ratio of 3:1:1. To augment the training set, the mosaic technique was applied. After augmentation, the dataset was expanded to 6295 images, as shown in Table 1. The mosaic method randomly combined four images and performed operations such as horizontal and vertical flipping, random scaling, and adjustments to brightness and contrast, as illustrated in Figure 4.

Table 1. Image number of the lychee disease dataset.

Diseases	Original Data	Training Set	Validation Set	Test Set
Bituminous	318	1570	79	79
Anthrax	326	1576	81	81
Felt diseases	314	1581	78	78
Leaf gall	315	1568	78	78

This data preprocessing approach ensured that the lychee disease image dataset was enriched and diverse, allowing more effective training of the model and enabling it to handle various scenarios during testing. The balanced distribution between the training and testing sets further guaranteed the reliable evaluation and validation of the proposed YOLOv7-MGPC network architecture.

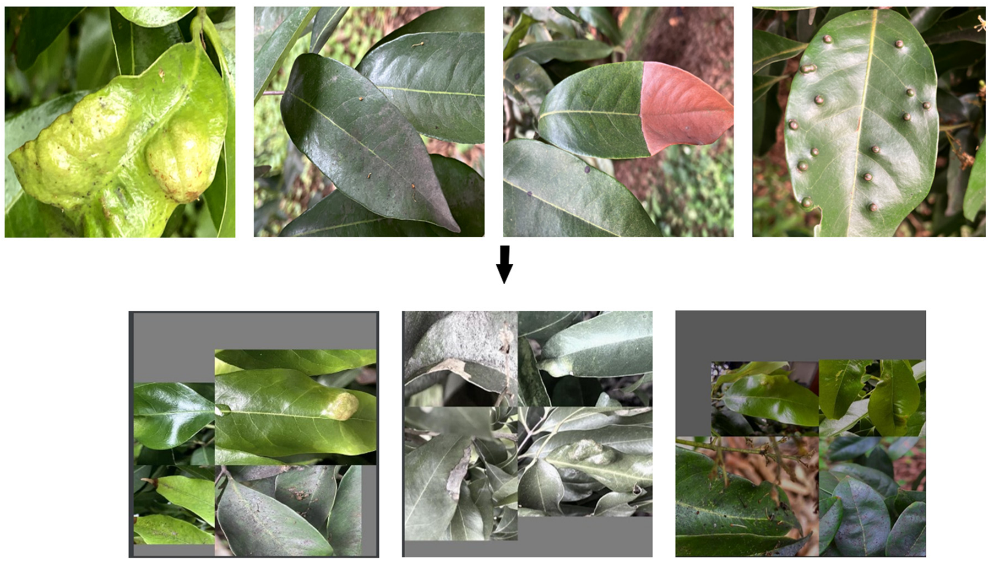


Figure 4. Data enhancement.

3.3. Model Architecture and Deployment

3.3.1. YOLOv7 Algorithm

YOLOv7 is the seventh-generation algorithm based on the YOLO series, which builds upon the optimization of YOLOv5. It showcases advantages in both accuracy and speed compared to its predecessors. Despite the improved accuracy, YOLOv7 still has a complex network structure and a large number of parameters. This demands the use of high-performance devices and makes it unsuitable for deployment on edge terminal devices. Therefore, a series of lightweight model techniques is required to make it more feasible for deployment on resource-constrained edge devices.

3.3.2. GhostNetV1 Lightweight Network

Conventional convolution operations in network structures like ResNet-50 typically generate a substantial number of redundant or similar feature maps. While this helps in achieving a comprehensive understanding of the input data, it also leads to increased computational demands for the model. To address this issue, an alternative approach that does not rely on traditional convolutions to generate redundant feature maps is adopted. The GhostNetV1 network utilizes a limited number of traditional convolutions to create a subset of feature maps. By applying simple linear transformations to this subset, high-level semantic information can be captured with limited computations

As shown in Figure 5, Φ_i represents a linear transformation. Assuming that there are n output feature maps, the Ghost module will obtain m feature maps through identity mapping, as well as $m \times (s-1)$ linear operations (where s represents the number of feature maps after linear operations). Here, $n = m \times s$. We define the kernel size for each linear operation and the traditional convolution as $k \times k$. When s is much smaller than c, it can be deduced that the acceleration ratio (r_s) and parameter compression ratio (r_c) for the Ghost module, compared to standard convolution, can be calculated as follows:

$$\text{r\_s} = (c \times k \times k \times n \times \mathrm{h} \times \mathrm{w}')/(m \times k \times k \times n \times \mathrm{h}' \times \mathrm{w}' + m \times (s-1) \times k \times k \times \mathrm{h}' \times \mathrm{w}') \approx (s \times cf)/(s + c - 1) \approx s, \quad (1)$$

$$\text{r\_c} = (n \times c \times k \times k)/(m \times c \times k \times k + m \times (s-1) \times k \times k) \approx (s \times c)/(s + c - 1) \approx s \quad (2)$$

The Ghost module is employed to reduce the model parameters and accelerate the inference speed by s times. This approach enhances the redundancy of feature maps and reduces computational overhead, aligning with the lightweight network requirements of this experiment.

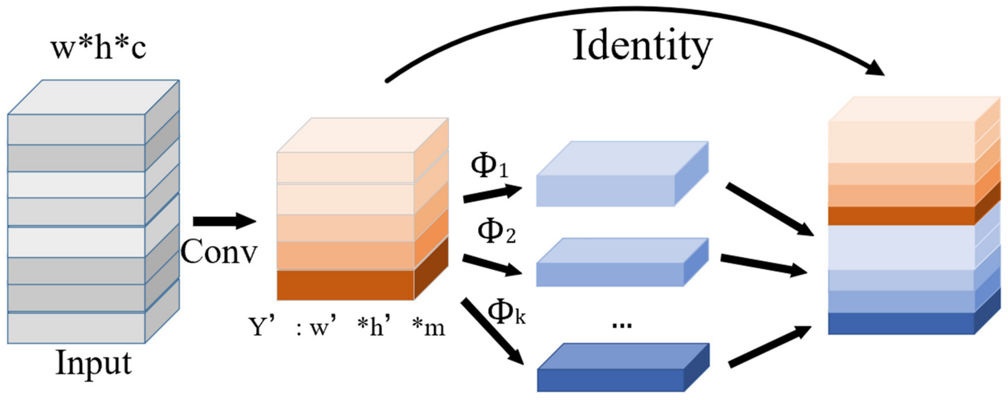


Figure 5. The principle of the Ghost module.

3.3.3. BN Layer-Channel Pruning

Among structured pruning methods, channel pruning is the most commonly used approach. In the BN layer, the scaling factor γ is inter-related to each channel in the convolutional layer, as shown in the Figure 6. During the training process, these scaling factors γ undergo sparse regularization, automatically filtering out unimportant channels. Channels with smaller scaling factors will be pruned, resulting in a compact model. This compact model is then fine-tuned to achieve comparable (or even higher) accuracy than that of the fully trained network.

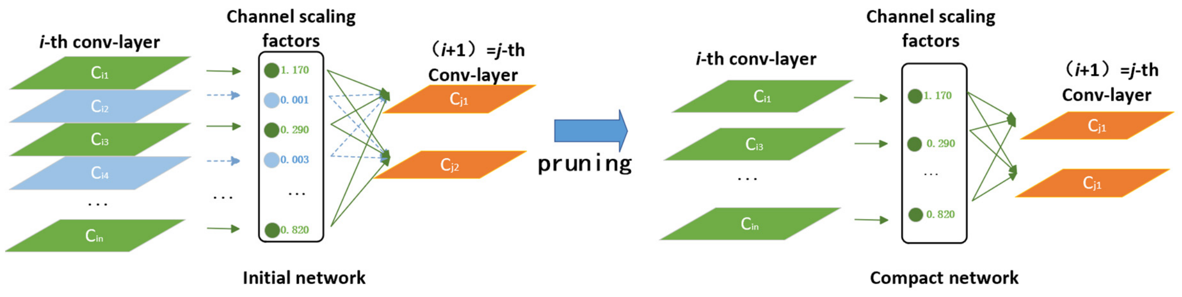


Figure 6. Channel pruning principle.

The specific operations of the BN layer pruning consist of three parts:

$$\hat{z} = \frac{z_{in} - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}, \tag{3}$$

$$z_{out} = \gamma \hat{z} + \beta, \tag{4}$$

$$L = \sum_{(x,y)} l(f(x,W),y) + \lambda \sum_{\gamma \epsilon \Gamma} g(\gamma) \tag{5}$$

As shown in Equations (3) and (4), z_{in} and z_{out} represent the input and output of the BN layer, respectively. μ_B and σ_B represent the mean and standard deviation of each batch. The parameter ϵ is a small positive number used to avoid division by zero. Additionally, the BN layer introduces scaling factors γ and biases β for each channel, which normalize the channel data [28]. During the training process, when the scaling factor γ is small, the corresponding activation z_{out} will also be small. Output with γ approaching zero contributes minimally to the model, allowing the pruning of the channel output in the BN layer. Furthermore, when the scaling factor γ is relatively uniform, making it challenging to prune, as shown in Equation (5), we introduce the L1 regularization constraint of γ in the loss function to induce sparsity. The regularization term $g(\gamma) = |\gamma|$ is added to the function, where λ is the regularization coefficient.

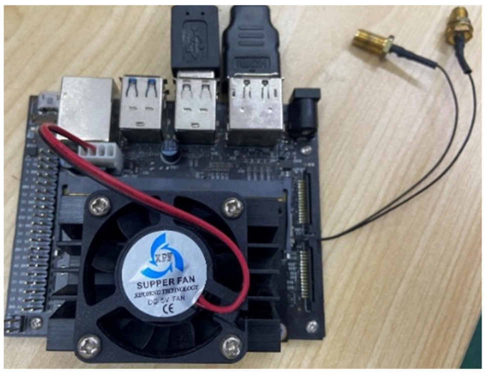
Based on the YOLOv7-GhostNetV1 structure used in this study, we set the sparsity level of γ to 0.005 and pruned the model by 50% after inducing sparsity. The aim was to reduce the number of parameters, improve speed, and meet the performance requirements for edge deployment while causing minimal changes to accuracy.

3.3.4. CBAM Attention Mechanism

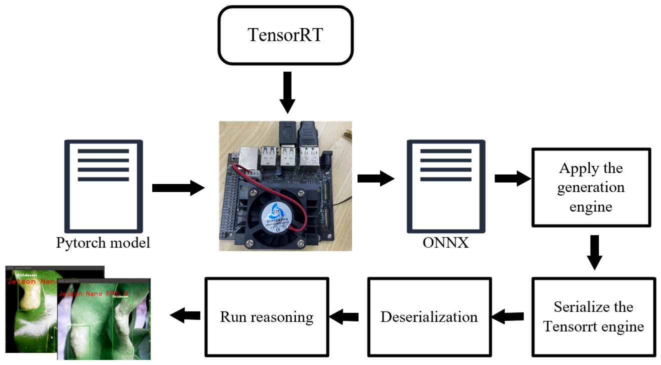
To ensure minimal loss in accuracy, this study introduced a lightweight attention mechanism, namely the Convolutional Block Attention Module (CBAM). CBAM operates on both spatial and channel dimensions. In our model, CBAM processing is applied to the output of feature layers in the backbone. It redistributes weights for different channel feature maps, thereby enhancing the extraction of deep-layer information in the network. This enables the network to focus on relevant features and ignore less important ones.

3.3.5. TensorRT Deployment on Jetson Nano

In this study, Nvidia's Jetson Nano development board was used to leverage TensorRT8.0 for accelerating inference on PyTorch's Pth models. The Jetson Nano is a product developed by NVIDIA, a technology company based in Santa Clara, CA, USA, which is a small, affordable, and energy-efficient single-board computer designed for various AI and machine learning applications at the edge. It is part of NVIDIA's Jetson family of products, known for their GPU acceleration and performance in AI tasks, as shown in Figure 7a.



(**a**) Jetson Nano



(**b**) Deployment process

Figure 7. Edge equipment deployment process.

TensorRT is a deep learning inference optimizer that brings reduced latency and high throughput deployment to deep learning applications. Applications running TensorRT on Nvidia GPUs during inference can achieve significantly improved execution speeds compared to pure CPU platforms.

The specific deployment process involves using a pre-trained PyTorch model and transferring it to the Jetson Nano. Subsequently, it undergoes conversion into the ONNX model format, followed by importation into TensorRT for the generation of an accelerated network inference engine. The resulting engine file can be permanently stored through serialization, enabling its subsequent loading through deserialization to facilitate inference. The overall structure is illustrated in Figure 7b.

3.3.6. Evaluation of the Model Performance

To conduct a comprehensive analysis of the lychee disease detection performance, this study utilized various evaluation metrics, including detection accuracy (P), recall rate (R), mean average precision (mAP), frames per second (FPS), and model parameter count (params/M). These metrics were used to compare and assess the effectiveness of different models and their detection results.

$$P = \frac{TP}{TP + FP}, \tag{6}$$

$$R = \frac{TP}{TP + FN}, \tag{7}$$

$$mAP = \frac{\int_0^1 PRdR}{n} \tag{8}$$

In Equations (6)–(8), TP represents the number of targets correctly detected via the algorithm, FP represents the number of targets incorrectly detected via the algorithm, FN represents the number of targets missed via the algorithm, and 'n' represents the total number of images in the dataset.

4. Result and Discussion

4.1. Ablation Experiments

To validate the effectiveness of the proposed methods in this paper, two sets of experiments were designed for comparative analysis. To ensure the experiment's accuracy, the same parameters were used during the training process. The test equipment uses the Windows 10 operating system, an Intel Core I9-10900K processor, Pytorch1.9.0, an Nvidia GeForce RTX3090 graphics card, and Jetson Nano, as shown in Table 2.

Table 2. Test system configuration.

Computer Configuration	Specific Parameters
Operating system	Windows10
CPU	Intel(R) Core I9-10900K
Training framework	Pytorch1.9.0
GPU	Nvidia GeForce RTX3090
Edge deployment devices	Jetson Nano (Quad-core ARMCortex-A57MPCore processor)
Edge device accelerator	TensorRT8.0

Table 3 demonstrates the impact of different techniques on the accuracy of the YOLOv7 model. The techniques evaluated include Mosaic data augmentation (M), using GhostNetV1 as the feature extraction network (G), BN layer pruning (P), and the CBAM attention mechanism (C). It is observed that applying data augmentation leads to a significant improvement in accuracy. The YOLOv7 model achieved a mAP of 77.3% using the original dataset, while it achieved a mAP of 92.8% after applying data augmentation, representing an increase of 15.5%. This substantial increase in model accuracy illustrates that data augmentation techniques, such as random flipping, rotation, and translation, enhance the diversity of the data, thereby improving the model's robustness and generalization performance while mitigating overfitting.

Furthermore, reducing the parameter count while maintaining reasonable performance is essential. The GhostNetV1 improved network (YOLOv7-MG) in Table 3, row 3, exhibits a slight decrease in accuracy compared to the original model (YOLOv7-M), but the parameter count reduces from 36.5M to 26M, validating the effectiveness of replacing it with a lightweight GhostNetV1 network. To meet the computation requirements for edge deployment, we prune the network (YOLOv7-M) to achieve a balance between performance and parameter reduction. After channel pruning, the model parameters reduce from 36.5 M to 9.5 M in row 4 (YOLOv7-MP), which is almost one-fourth of the original count, and the speed increases by 136 frames/s, with only a slight decrease in accuracy.

Table 3. Comparison of ablation results.

Model	mAP@50 (%)	Params/M	FPS (Frames/s)	P	R
YOLOv7	77.3	36	120	83.2	81.5
YOLOv7 + M	92.8	36.5	120	96.3	90.1
YOLOv7 + M + G	89.5	26	131	95.4	84.4
YOLOv7 + M + P	87	9.5	256	94.8	83.2
YOLOv7 + M + G + P	86.5	6.6	222	94.0	78.9
YOLOv7 + M + G + P + C	88.6	7.8	217	95.0	80.7

Combining the GhostNetV1 network with channel pruning, as shown in Table 3, row 5 (YOLOv7-MGP), the improved network model achieves a nearly 30M reduction in parameter count compared to the original model (YOLOv7-M), with a 6.3% decrease in accuracy. Therefore, the combination of lightweight networks and pruning operations results in a significant reduction in parameters but may lead to some information loss and a decrease in accuracy. To restore the model performance, the CBAM attention mechanism is introduced to enhance its detection capability. The model with CBAM (YOLOv7-MGPC) shows a 2.1% increase in accuracy with only a slight increase in the parameter size and inference speed. This performance improvement is practical for model deployment where computational resources are limited.

Overall, the comprehensive use of lightweight networks, pruning operations, and the CBAM attention mechanism allows the model's accuracy to remain above 88%, while the parameter count decreases significantly from 36.5 M to 7.8 M, and the speed increases from 120 frames/s to 217 frames/s. These improvements make the modified model more suitable for deployment on mobile devices, as it not only reduces the parameter size but also accelerates inference speed.

4.2. Performance of the Attention Mechanism

Figure 8 compares the performance of YOLOv7-MGPC with that of YOLOv7-MGP. It is observed that YOLOv7-MGPC learns a little slower due to the process of the adaptation of the attention mechanism to the specific pattern of the lychee data and extraction of informative features. After training convergence, the CBAM attention mechanism results in a 2% increase in mAP with negligible computational costs.

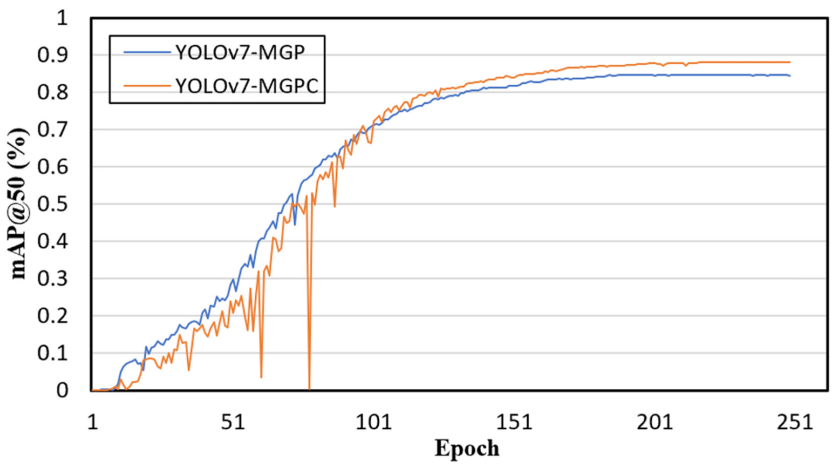


Figure 8. Curve of network accuracy changes with number of epoch.

To further demonstrate the effectiveness of the improved YOLOv7-MGPC model, Class Activation Maps (CAM) [29] were used for the visual analysis of disease detection. The heatmaps in Figure 9 depict the model's weight and focus transfer during training, demonstrating which part of the features contribute to the classification decision. It is evident that before the introduction of the attention mechanism, the network did not give sufficient attention to less prominent disease features. However, after incorporating the attention mechanism, the network becomes capable of detecting finer disease features and accurately identifying the affected areas.

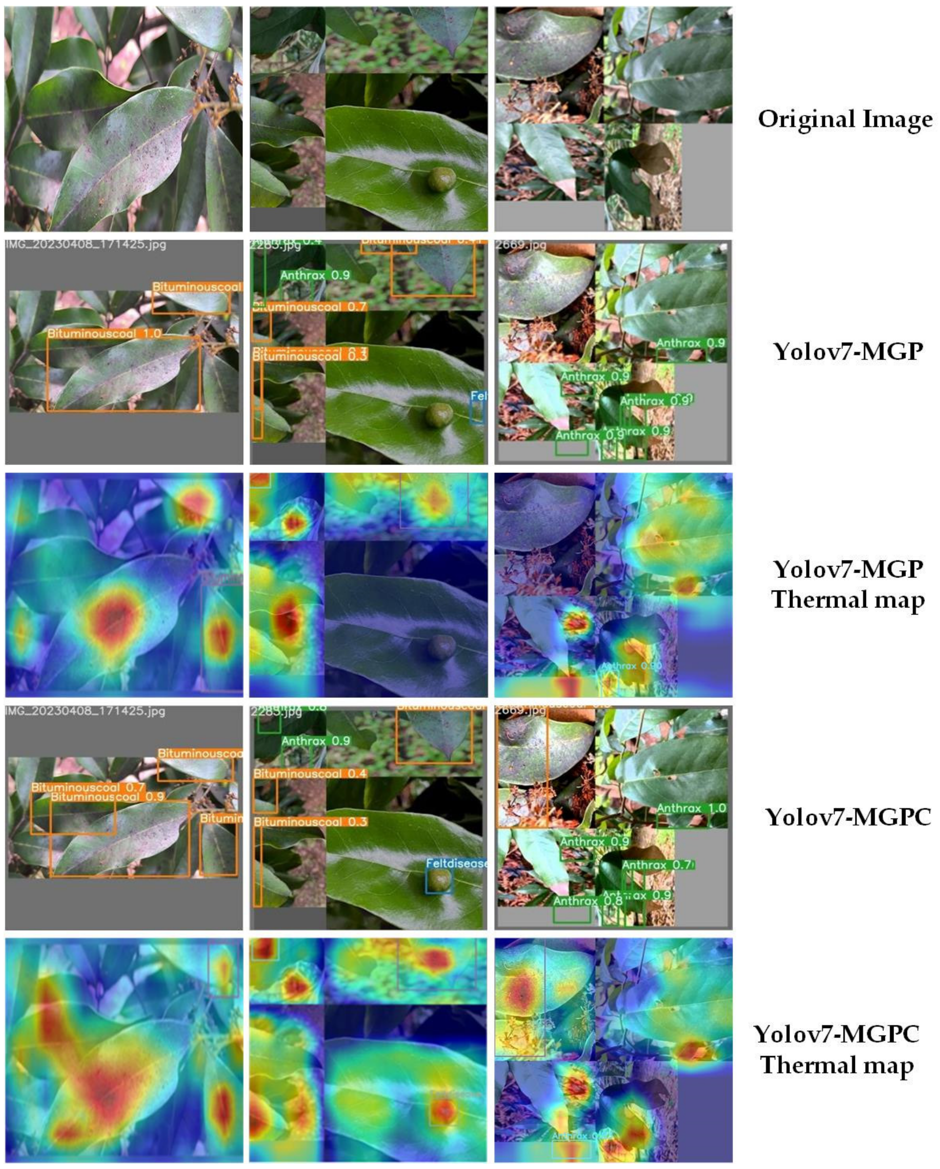


Figure 9. Class activation feature heatmap visualization results.

4.3. Different Network Experiment Contrast

In order to further validate the effectiveness of the proposed model, comparative experiments were conducted by replacing the original YOLOv7 backbone network with current main-stream lightweight feature extraction models, including MobileNetV3, ShuffleNetV2, and ResNet50. As shown in Figure 10 and Table 4, it is evident that the improved YOLOv7-MGPC outperforms other lightweight networks in terms of accuracy. It achieves a higher accuracy, ranging from 4.9% to 9.4%, compared to the other models. Regarding detection speed, when compared to YOLOv7-ShuffleNetV2, YOLOv7-MGPC is only 16 frames/s slower, but its accuracy is 5.5% higher. Compared to YOLOv7-MobileNetV3 and YOLOv7-ResNet50, YOLOv7-MGPC has a speed advantage of 10 frames/s to 35 frames/s while maintaining comparable parameter sizes. Figure 11 shows the confusion matrix for the recognition results of different disease classes obtained via various network models.

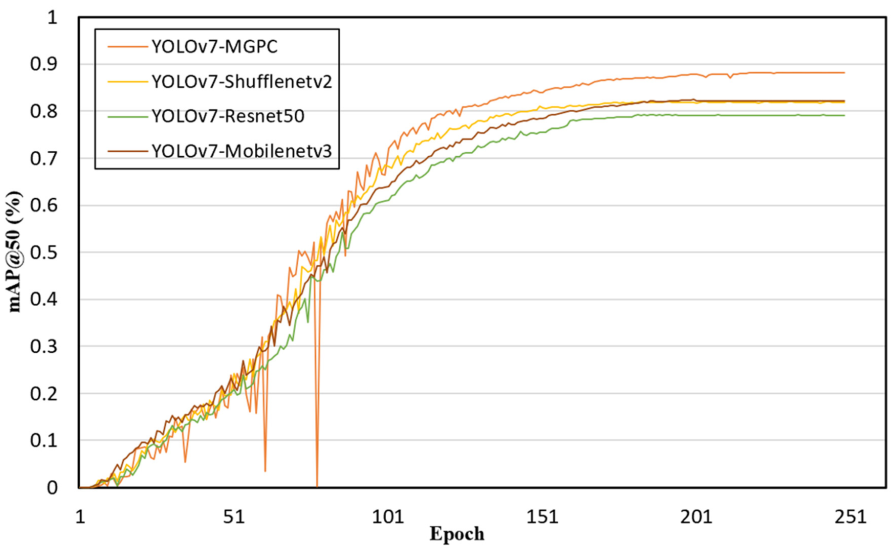


Figure 10. The network accuracy of each models varies with the number of epochs.

Table 4. The comparative experimental results.

Model	mAP@50 (%)	Params/M	FPS (Frames/s)	P	R
YOLOv7-MobilenetV3	83.7	7.1	207	94.2	75.7
YOLOv7-ShufflenetV2	83.1	6.6	233	93.4	74.6
YOLOv7-Resnet50	79.2	10.2	182	89.8	71.2
YOLOv7-MGPC	88.6	7.8	217	95.0	80.7

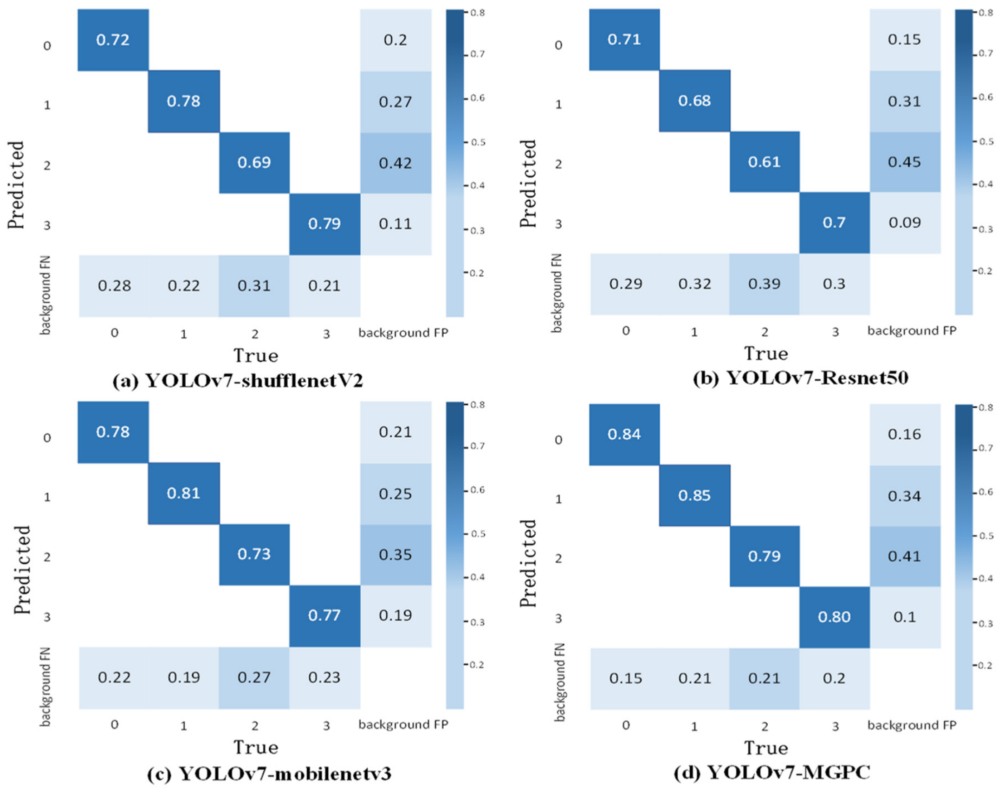


Figure 11. Confusion matrixes of different network models used to identify results.

To comprehensively evaluate the performance of our model in lychee disease detection, we conducted a comparative analysis of the YOLOv7 model to compare it to other prominent object detection algorithms, including RetinaNet [30], SSD [31], YOLOv5s [32], and YOLOv8s [33]. Among them, RetinaNet uses the ResNet50 backbone network, and the SSD model uses the VGG16 backbone network. As demonstrated in Table 5, it is evident that the enhanced YOLOv7 + MGPC surpasses mainstream object detection networks like RetinaNet and SSD in terms of both accuracy and speed while maintaining a minimal number of parameters. Compared to its sibling YOLO models, YOLOv7-MGPC exhibits similar accuracy and parameter efficiency as YOLOv5s but outperforms it in terms of FPS by 68 frames/s. Although YOLOv8s achieves higher accuracy compared to YOLOv7-MGPC, it lags in terms of inference speed and parameter efficiency. When considering the balance of accuracy, speed, model size, and practical applicability for deployment, we favor the well-rounded improvement achieved via the YOLOv7-MGPC model.

Table 5. Comparison of mainstream object detection capabilities.

Model	mAP@50 (%)	Params/M	FPS (Frames/s)	P	R
RetinaNet	82.5	36.5	67	85.3	80.1
SSD	72.7	24.9	74	76.5	70.3
YOLOv5s	85.4	7.6	149	88.0	82.4
YOLOv8s	89.1	11.4	178	93.6	83.2
YOLOv7-MGPC	88.6	7.8	217	95.0	80.7

As demonstrated in Table 6, we have conducted a comparative analysis of our model and contemporary deep learning models for crop disease detection. Evaluating the data presented in Table 6, we can conclude that our model achieves a better balance in terms of accuracy, parameter count, and speed, which makes it a more optimal choice for real-world applications.

Table 6. Findings of existing related studies.

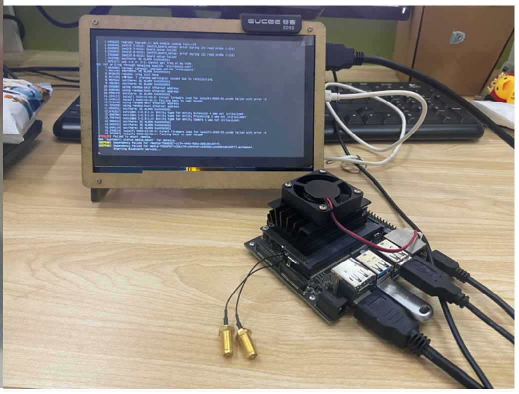
Ref.	Dataset Name	Model Name	Number of Image	Model Size	Performance	Inference Time (ms)
Zhang et al. [34]	Soybean leaf	Multi-feature fusion Faster RCNN	1000	-	mAP = 83.34%	0.859
Su et al. [35]	Wheat	Mask + ResNet 101	3000	-	mAP = 77.85%	0.68
Liu et al. [36]	Tomato leaf	YOLOX-MobileNetV3	18,834	44.31 M	mAP = 98.56%	-
Shao et al. [37]	Cotton Leaf	ResNet	5903	21.55 M	Accuracy = 96.61%	-
Ours	Litchi leaf	YOLOv7 + GhostNetV1	6295	18.37 M	mAP = 88.6%	4.60

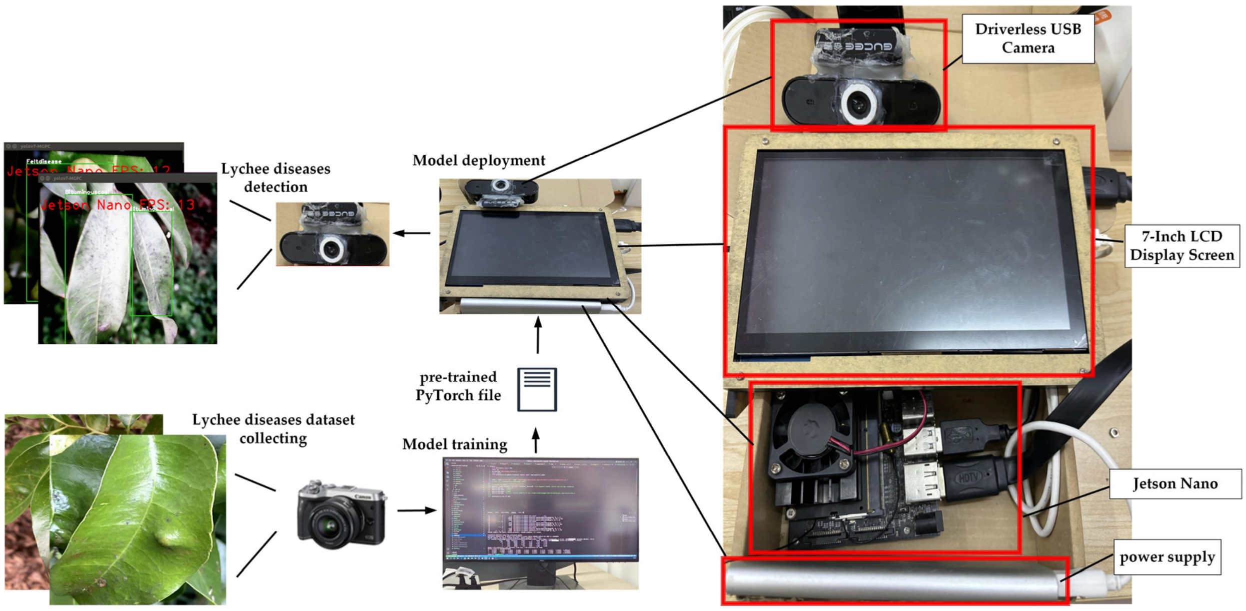
4.4. Platform Deployment and Verification

To further validate the effectiveness of the improved lightweight algorithm on mobile devices, this study deployed the YOLOv7-MGPC model on edge computing devices. Real-time testing was conducted within the South China Agricultural University's arboretum.

As illustrated in Figure 12, the system configuration included a GUCEEHD98 camera and a Nvidia Jetson Nano edge computing device, with the Jetson Nano running the Linux operating system and utilizing TensorRT acceleration capabilities. This deployment allowed for practical, on-site testing and demonstrated the practicality of the YOLOv7-MGPC model in real-world agricultural scenarios thanks to its lightweight design and optimized inference process using TensorRT on the Nvidia Jetson Nano.

(**a**) UVC Camera


(**b**) Jetson Nano detection system



(**c**) Deployment process

Figure 12. Deployment units.

As illustrated in Figure 13, it is evident that our easily constructed Jetson Nano detection platform excels in real-time scene detection, accurately identifying lychee diseases at a remarkable speed range of 8–13 frames/s. The real-time detection demonstrated in the field underscores the model's efficiency, precise target recognition, and ability to meet the accuracy and efficiency demands of practical applications. This achievement holds promising implications for the advancement of agricultural intelligence and automation projects, providing both the technical prowess and theoretical foundation necessary for their success.

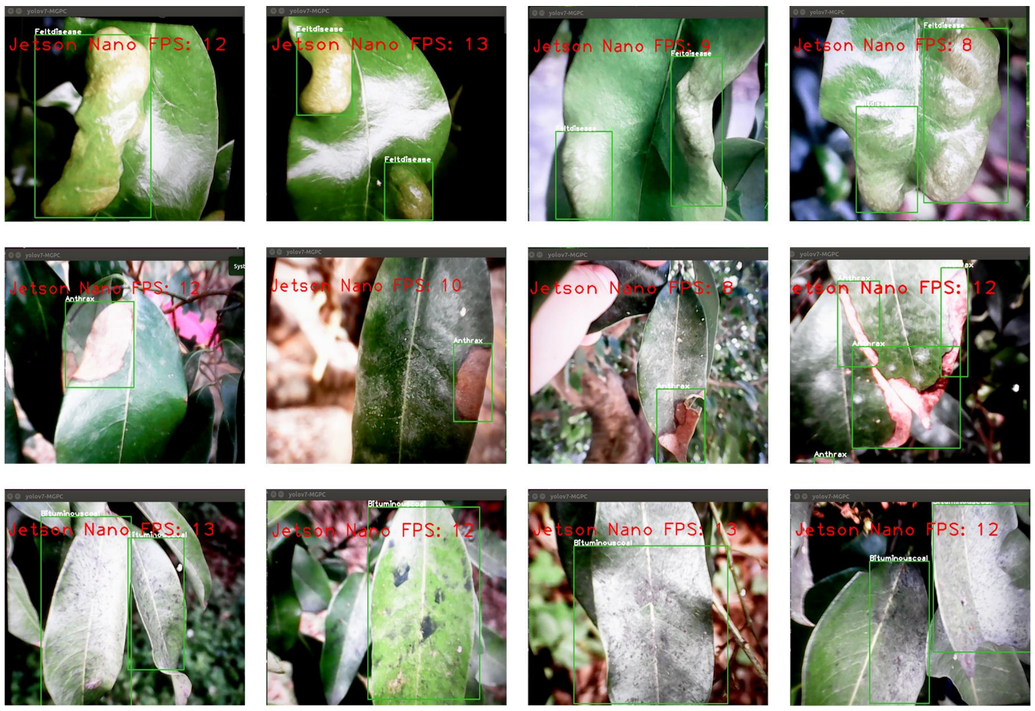


Figure 13. Real-time inspection effect drawing.

5. Conclusions

In this study, we replaced the main feature extraction network of the YOLOv7 algorithm with the lightweight neural network GhostNetV1, designed for mobile devices. Furthermore, we employed BN layer network sparsification to conduct channel pruning and applied the CBAM mechanism to enhance detection accuracy, resulting in an improved version of the lychee disease detection model, namely YOLOv7-MGPC. The YOLOv7-MGPC model was trained on a dataset containing four types of lychee disease images.

By conducting comparative experiments and evaluating the model's performance on metrics such as mean average accuracy (mAP), parameter count, and frames per second (FPS), the enhanced YOLOV7-MGPC network demonstrates significant advancements. Notably, the network achieves a mAP of 88.6% while reducing the parameter count from 36.5 million to 7.8 million and enhancing speed from 120 FPS to an impressive 217 FPS. By deploying our model on the Jetson Nano platform, we have demonstrated its capability to achieve real-time detection, offering immediate feedback to farmers and decision makers. Our system, consisting of a camera mounted on a Jetson Nano platform, continuously captures images of the lychee orchard. The images are then processed using our real-time detection model, which instantly identifies disease symptoms. These symptoms can be transmitted to the farmer's mobile device or computer, enabling them to make timely decisions regarding disease management. This real-time functionality enables early intervention and targeted treatment, reducing the impacts of lychee diseases and increasing crop yield.

The lightweight algorithm proposed in this study offers a new approach for the accurate and real-time detection of lychee diseases, and we are committed to continuously optimizing our model. Our goal is to create a robust and versatile system that can be widely deployed in various agricultural settings to assist farmers in effectively identifying and managing lychee diseases. We hope our research findings can provide technical support for future into on lychee disease detection.

Author Contributions: Conceptualization, J.X. and X.Y.; methodology, J.X., G.K. and L.W.; software, J.X. and L.W.; validation, J.X. and L.W.; formal analysis, J.X. and L.W.; investigation, J.X., Y.L. and R.Z.; resources, J.X. and X.Y.; data curation, J.X., Y.L., R.Z. and J.Z.; writing—original draft preparation, J.X. and L.W.; writing—review and editing, J.X., G.K., L.W. and F.Z.; visualization, J.X.; supervision, X.Y. and L.W.; project administration, X.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the Science and Technology Planning Project of Guangzhou, grant number 202206010088; the Scientific Research Project of Hunan Provincial Department of Education, grant number 21B0741; and the Yongzhou Guiding Science and Technology Plan Project, grant number 2021-YZKJZD-007.

Data Availability Statement: The data presented in this study are available from the corresponding author upon request. The data are not publicly available due to the privacy policy of the organization.

Acknowledgments: The authors would like to thank the anonymous reviewers for their critical comments and suggestions for improving the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Xiang, X. Bottleneck of litchi industry development and progress of technology research and development. *Guangdong Agric. Sci.* **2020**, *47*, 32–41.
2. Zhuang, L.; Qiu, Z. Development characteristics and policy recommendations of China's lychee industry in 2019. *South China Fruits* **2021**, *50*, 184–188.
3. Ling, J.F.; Song, X.B.; Xi, P.G.; Cheng, B.P.; Cui, Y.P.; Chen, X.; Peng, A.T.; Jiang, Z.D.; Zhang, L.H. Identification of Colletotrichum Siamense Causing Litchi Pepper Spot Diseases in Mainland China. *Plant Pathol.* **2019**, *68*, 1533–1542. [CrossRef]
4. Sun, H.; Xu, H.; Liu, B.; He, D.; He, J.; Zhang, H.; Geng, N. MEAN-SSD: A novel real-time detector for apple leaf diseases using improved light-weight convo-lutional neural networks. *Comput. Electron. Agric.* **2021**, *189*, 106379. [CrossRef]
5. Lecun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef] [PubMed]
6. Dargan, S.; Kumar, M.; Ayyagari, M.R.; Kumar, G. Asurvey of deep learning and its applications: A new paradigm to machine learning. *Arch. Comput. Methods Eng.* **2020**, *27*, 1071–1092. [CrossRef]
7. Too, E.C.; Yujian, L.; Njuki, S.; Yingchun, L. A comparative study of fine-tuning deep learning models for plant diseases identification. *Comput. Electron. Agric.* **2019**, *161*, 272–279. [CrossRef]
8. Jia, S.; Gao, H.; Hang, X. Research Progress on Crop Diseases Image Recognition Technology Based on Deep Learning. *Trans. Chin. Soc. Agric. Mach.* **2019**, *50* (Suppl. S1), 313–317.
9. Liu, J.; Wang, X.; Liu, G. Tomato Pests Recognition Algorithm Based on Improved YOLOv4. *Front. Plant Sci.* **2022**, *13*, 814681. [CrossRef]
10. Zhang, Y.; Yang, G.; Liu, Y.; Wang, C.; Yin, Y. An Improved YOLO Network for Unopened Cotton Boll Detection in the Field. *J. Intell. Fuzzy Syst.* **2022**, *42*, 2193–2206. [CrossRef]
11. Liu, J.; Cong, W.; Li, H. Vehicle Detection Method Based on GhostNetV1-SSD. In Proceedings of the 2020 International Conference on Virtual Reality and Intelligent Systems (ICVRIS), Zhangjiajie, China, 18–19 July 2020; pp. 200–203. [CrossRef]
12. Mannanuddin, K.; Aluvala, S.; Sneha, Y.; Kumaraswamy, E.; Sudarshan, E.; Mahender, K. Confluence of Machine Learning with Edge Computing for IoT Accession. In *IOP Conference Series: Materials Science and Engineering, International Conference on Recent Advancements in Engineering and Management (ICRAEM-2020), Warangal, India, 9–10 October 2020*; IOP Publishing Ltd.: Bristol, UK, 2020; Volume 981, p. 042003. [CrossRef]
13. Yang, M.-D.; Boubin, J.G.; Tsai, H.P.; Tseng, H.H.; Hsu, Y.C.; Stewart, C.C. Adaptive autonomous UAV scouting for rice lodging assessment using edge computing with deep learning EDANet. *Comput. Electron. Agric.* **2020**, *179*, 105817. [CrossRef]
14. Ajayi, O.G.; Ashi, J.; Guda, B. Performance evaluation of YOLOv5 model for automatic crop and weed classification on UAV images. *Smart Agric. Technol.* **2023**, *5*, 100231. [CrossRef]
15. Islam, S.; Akter, S.; Islam, M.; Rahman, M. DCNN Based Diseases Prediction of Lychee Tree. In *Key Digital Trends in Artificial Intelligence and Robotics. ICDLAIR 2022. Lecture Notes in Networks and Systems*; Troiano, L., Vaccaro, A., Kesswani, N., Díaz Rodriguez, I., Brigui, I., Pastor-Escuredo, D., Eds.; Springer: Cham, Switzerland, 2023; Volume 670. [CrossRef]
16. Xie, J.; Zhang, X.; Liu, Z.; Liao, F.; Wang, W.; Li, J. Detection of Litchi Leaf Diseases and Insect Pests Based on Im-proved FCOS. *Agronomy* **2023**, *13*, 1314. [CrossRef]
17. Mahmud, M.P.; Ali, M.A.; Akter, S.; Bijoy, M.H.I. Lychee Tree Diseases Classification and Prediction using Transfer Learning. In Proceedings of the13th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kharagpur, India, 3–5 October 2022; pp. 1–7. [CrossRef]
18. Wang, C.; Xiao, Z. Lychee Surface Defect Detection Based on Deep Convolutional Neural Networks with GAN-Based Data Augmentation. *Agronomy* **2021**, *11*, 1500. [CrossRef]

19. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
20. Han, K.; Wang, Y.; Tian, Q.; Guo, J.; Xu, C.; Xu, C. Ghostnet: More features from cheap operations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 1580–1589.
21. Zhang, X.; Zhou, X.; Lin, M.; Sun, J. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6848–6856.
22. Shi, R.; Li, T.; Yamaguchi, Y. An attribution-based pruning method for real-time mango detection with YOLO network. *Comput. Electron. Agric.* **2020**, *169*, 105214. [CrossRef]
23. Wang, F.; Jiang, M.; Qian, C.; Yang, S.; Li, C.; Zhang, H.; Wang, X.; Tang, X. Residual attention network for image classification. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, 21–26 July 2017; pp. 3156–3164.
24. Zhang, X.; Wang, T.; Qi, J.; Lu, H.; Wang, G. Progressive Attention Guided Recurrent Network for Salient Object Detection. In Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, 18–22 June 2018; pp. 714–722.
25. Fu, J.; Liu, J.; Tian, H.; Li, Y.; Bao, Y.; Fang, Z.; Lu, H. Dual Attention Network for Scene Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, 16–20 June 2019; pp. 3146–3154.
26. Xikun, X.; Changjiang, L.; Meng, X. Application of attention YOLOV4 algorithm in metal defect detection. In Proceedings of the IEEE International Conference on Emergency Science and Information Technology (ICESIT), Chongqing, China, 22–24 November 2021; pp. 465–468. [CrossRef]
27. Li, H.; Deng, L.; Yang, C.; Liu, J.; Gu, Z. Enhanced YOLO v3 Tiny Network for Real-Time Ship Detection from Visual Image. *IEEE Access* **2021**, *9*, 16692–16706. [CrossRef]
28. Liu, Z.; Li, J.; Shen, Z.; Huang, G.; Yan, S.; Zhang, C. Learning efficient convolutional networks through network slimming. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2736–2744.
29. Zhou, B.; Khosla, A.; Lapedriza, A.; Oliva, A.; Torralba, A. Learning deep features for discriminative localization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2921–2929.
30. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
31. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Springer International Publishing: Berlin/Heidelberg, Germany; pp. 21–37.
32. Jocher, G. YOLOv5 by Ultralytics, version 7.0.; 2020. Available online: https://github.com/ultralytics/yolov5 (accessed on 6 September 2023).
33. Jocher, G.; Chaurasia, A.; Qiu, J. YOLO by Ultralytics 2023. Available online: https://github.com/ultralytics/ultralytics (accessed on 6 September 2023).
34. Zhang, K.; Wu, Q.; Chen, Y. Detecting soybean leaf disease from synthetic image using multi-feature fusion Faster R-CNN. *Comput. Electron. Agric.* **2021**, *183*, 106064. [CrossRef]
35. Su, W.-H.; Zhang, J.; Yang, C.; Page, R.; Szinyei, T.; Hirsch, C.D.; Steffenson, B.J. Automatic evaluation of wheat resistance to fusarium head blight using dualmask-RCNN deep learning frameworks in computer vision. *Remote Sens.* **2021**, *13*, 26. [CrossRef]
36. Liu, W.; Zhai, Y.; Xia, Y. Tomato Leaf Disease Identification Method Based on Improved YOLOX. *Agronomy* **2023**, *13*, 1455. [CrossRef]
37. Shao, M.; He, P.; Zhang, Y.; Zhou, S.; Zhang, N.; Zhang, J. Identification Method of Cotton Leaf Diseases Based on Bilinear Coordinate Attention Enhancement Module. *Agronomy* **2023**, *13*, 88. [CrossRef]