agriculture

MDPI

Article

VLDNet: An Ultra-Lightweight Crop Disease Identification Network

Xiaopeng Li, Yichi Zhang, Yuhan Peng and Shuqin Li *

College of Information Engineering, Northwest A&F University, Xianyang 712100, China
* Correspondence: lsq_cie@nwsuaf.edu.cn

Abstract: Existing deep learning methods usually adopt deeper and wider network structures to achieve better performance. However, we found that this rule does not apply well to crop disease identification tasks, which inspired us to rethink the design paradigm of disease identification models. Crop diseases belong to fine-grained features and lack obvious patterns. Deeper and wider network structures will cause information loss of features, which will damage identification efficiency. Based on this, this paper designs a very lightweight disease identification network called VLDNet. The basic module VLDBlock of VLDNet extracts intrinsic features through 1×1 convolution, and uses cheap linear operations to supplement redundant features to improve feature extraction efficiency. In inference, reparameterization technology is used to further reduce the model size and improve inference speed. VLDNet achieves state-of-the-art model (SOTA) latency-accuracy trade-offs on self-built and public datasets, such as equivalent performance to Swin-Tiny with a parameter size of 0.097 MB and 0.04 G floating point operations (FLOPs), while reducing parameter size and FLOPs by 297 times and 111 times, respectively. In actual testing, VLDNet can recognize 221 images per second, which is far superior to similar accuracy models. This work is expected to further promote the application of deep learning-based crop disease identification methods in practical production.

Keywords: disease identification; lightweight model; reparameterization; CNN

Citation: Li, X.; Zhang, Y.; Peng, Y.; Li, S. VLDNet: An Ultra-Lightweight Crop Disease Identification Network. *Agriculture* **2023**, *13*, 1482. https://doi.org/10.3390/agriculture13081482

Academic Editor: Maciej Zaborowicz

Received: 13 June 2023
Revised: 18 July 2023
Accepted: 21 July 2023
Published: 26 July 2023

1. Introduction

Traditional crop disease identification relies mainly on the long-term accumulated experience of farmers, as the symptoms of diseases are complex and varied, requiring high levels of professional knowledge from agricultural producers. Manual observation and judgment of disease types may be subject to strong subjectivity and is also time-consuming. Therefore, using modern information technology to achieve efficient and accurate crop disease identification is of great importance. Traditional image processing methods require manual disease spot segmentation, feature extraction, and classifier construction, which consume a lot of time in terms of data preprocessing and are greatly affected by objective conditions, making feature extraction difficult [1,2].

In recent years, the rapid development of deep learning has provided new solutions for agriculture disease identification. Convolutional neural networks (CNN) have powerful feature extraction capabilities and have achieved successful applications in the fields of image classification [3], object detection [4], semantic segmentation [5], etc. In the field of agriculture, some scholars have successfully applied deep learning to crop disease identification. For example, ref. [6] used healthy and diseased leaf images to train CNN models for disease detection and diagnosis, achieving an identification accuracy of 99.53% when classifying 17,548 plant leaf images using VGGNet [7]. Ref. [8] used more than 40,000 images to train the GoogleNet model and obtained identification accuracies ranging from 75% to 100% on different plants. Ref. [9] proposed a new deep neural network structure consisting of two sub-models that separated the tree leaves from the background in the original image. Then, various popular pre-trained models were used to extract

features and classify diseases. They achieved an 87.45% identification accuracy in the 2019 AI Challenger competition. Early research in crop disease identification used relatively simple CNN structures, which had a large number of parameters and lower identification accuracy in complex environments. Some studies have also used the Vision Transformer [10] for disease identification, which has become popular recently and has achieved good results. For example, ref. [11] proposed a method that combines CNN and Transformer structures for kiwi disease identification, achieving an identification accuracy of 98.78% on a self-built dataset. Ref. [12] proposed PlantXViT based on traditional CNN and Visual Transformer for apple, corn, and rice disease identification, with average identification accuracies exceeding 93.55%, 92.59%, and 98.33%, respectively.

Through the analysis of the above research work of crop disease recognition based on deep learning, we found that these models usually directly or indirectly use models that perform well on the publicly available ImageNet dataset. These models achieve excellent identification performance on ImageNet by designing very deep and wide networks to learn different feature patterns for various objects, including cats and dogs. After the input image enters the network model, the detailed information gradually decreases, and the final network model uses advanced semantic information for the final decision. However, as the features of crop leaf surface diseases are usually discrete and similar, and often small, there is no obvious pattern, which means that models that achieve good identification performance on ImageNet may not necessarily be able to improve the performance of crop disease identification solely by stacking network layers and increasing model width. On the contrary, this may lead to the loss of detailed disease features and undoubtedly increase the model's parameters and FLOPs, ultimately leading to a decrease in identification accuracy.

The above disease recognition model also faces the problem of a large number of parameters and complex calculation. In order to reduce model parameters and FLOPs, some researchers have proposed lightweight deep learning algorithms that can help deep learning models to be applied to different edge devices. For example, ref. [13] combined the advantages of Transformer and CNN to propose lightweight apple disease recognition, and obtained competitive results. Ref. [14] developed WearNet, a lightweight convolutional neural network, to enable automatic scratch detection of contact sliding parts such as metal molding. Compared with the existing networks, WearNet can achieve 94.16% excellent classification accuracy with smaller model size and faster detection speed. Ref. [15] proposed a lightweight sheep face recognition model, SheepFaceNet, which achieved 97.75% recognition accuracy with 0.60 MB parameters. Ref. [16] used depthwise separable convolutions to construct a lightweight CNN model for plant disease leaf classification. Compared with traditional CNN models, the network has fewer parameters. Ref. [17] proposed a lightweight SimpleNet model that performs well in automatic wheat spike disease identification. These works greatly reduce the number of model parameters and FLOPs, contributing to the deployment of deep learning models on edge devices. However, related research shows that these indicators may not have a good correlation with the model's inference speed. Efficiency indicators such as FLOPs do not consider memory access costs and parallelism, which can have a significant impact on latency during inference. Therefore, ref. [18] implemented real-time disease identification using Faster-RCNN and Yolov4, which is highly applicable to edge devices, but the identification accuracy cannot meet the needs of actual production.

In this context, we need to rethink the design paradigm of crop disease identification models and study algorithms that are computationally efficient and have high identification accuracy to serve the development of smart agriculture. This paper proposes a relatively shallow and narrow lightweight disease identification network VLDNet. The basic structure of VLDNet, VLDBlock, extracts intrinsic features through 1×1 convolution and uses cheap linear operations to supplement redundant features, improving feature extraction efficiency. During inference, reparameterization techniques are used to reduce model parameters and computational costs and increase inference speed. VLDNet achieved SOTA model latency-accuracy trade-offs on self-built and publicly available datasets, achieving performance

comparable to Swin-Tiny with 0.097 MB parameters and 0.04 G FLOPs, respectively, while reducing parameters and FLOPs by 297 times and 111 times, respectively. In actual testing, VLDNet can recognize 221 images per second, far superior to similar-accuracy models. This paper's work is expected to further promote the application of crop disease identification in actual production.

The contributions of this paper include:

1. We found no strict correlation between the depth, width design of the disease identification model, and the model performance.
2. We proposed a shallow, narrow crop disease identification model: VLDNet.
3. VLDNet achieved a good latency-accuracy tradeoff.

2. Materials and Methods

2.1. Datasets

2.1.1. PlantVillage Dataset

PlantVillage [19] is a publicly available dataset that contains 54,306 images of 38 classes of diseases and healthy crop leaves. These images are complete leaf images with a single background and without any interference from obstructions or background factors. This makes the PlantVillage dataset a good data foundation for research into disease identification. Information on the crop species and disease categories covered in this dataset can be found in the Table A1. In this paper, the dataset is divided into training, validation, and test sets in a ratio of 7:2:1. It should be noted that, due to the enormous size of the dataset, no data augmentation operations were performed. Some examples of this dataset are shown in Figure 1.

Figure 1. Examples of PlantVillage dataset.

2.1.2. Building Our Own Dataset

This dataset was collected from the apple and kiwi fruit experimental stations at the Northwest A&F University in Shaanxi Province, China. Healthy and diseased leaves were photographed using BM-500GE/BB-500GE (JAI Company, Copenhagen, Denmark) color digital cameras. The images have a resolution of 2456 × 2058 pixels, and a total of 4180 images were obtained. The dataset is divided into training, validation, and test sets in a ratio of 7:2:1. This includes six types of apple leaf disease images: spot, brown spot, flower leaf, gray spot, rust, and healthy, as well as four types of kiwi leaf disease images:

brown spot, flower leaf, anthracnose, and leaf blight. Examples of the images can be found in Figure 2.

Figure 2. Example of the self-built dataset.

To expand the dataset and improve the identification performance of the model, the necessary data augmentation strategies were applied to the training set, including random cropping, brightness adjustment, rotation, flipping, and adding salt and pepper noise and Gaussian noise to simulate the impact of the shooting equipment on the simulated results. A total of 14,600 images were obtained from the augmented training set. Detailed information on each type of image before and after enhancement is provided in Table 1. In order to reduce training time, the image sizes in the dataset were adjusted from 2456 × 2058 to 224 × 224.

Table 1. Statistics of self-built dataset.

Label ID	Type of Disease	Original Training Set	The Augmented Training Set
1	Apple Black rot	370	1850
2	Apple Brown spot	435	2175
3	Apple Healthy	475	2375
4	Kiwi Anthracnose	186	930
5	Kiwi Brown spot	70	350
6	Kiwi Leaf ulcer	99	495
7	Kiwi Mosaic leaf	61	305
8	Apple Mosaic leaf	375	1875
9	Apple Rust	438	2190
10	Apple Spotted leaf fall	411	2055
		2926	14,600

2.2. Methods

2.2.1. Do Deeper and Wider Networks Have Better Performance?

In this section, experiments were conducted on the PlantVillage dataset to preliminarily validate whether deeper and wider networks can achieve better performance. EfficientNet, PVTv2, and ResNet series models, each with different depths and widths, were selected for the experiment. The experimental results are shown in Figure 3. There is little difference in identification accuracy between the EfficientNet models, and their identification accuracy curves almost overlap. The loss curves of different sizes of EfficientNet models are also very consistent, with smaller models converging faster. For the PVTv2 model, the identification accuracy curves of models of different sizes are not stable, showing a fluctuating trend, but the final identification accuracy is almost the same. Moreover, the relatively smaller PVTv2-b1 obtained better identification results, which confirms our hypothesis. For the ResNet model, the identification accuracy of small-sized models is basically the same as or even higher than that of large-sized models. The convergence speed is also not significantly different, which further verifies our conjecture that deeper and wider models cannot achieve better results for crop disease identification tasks. Even if deeper and

wider models can achieve a slight identification accuracy advantage, this comes at the cost of increasing parameter size and computational complexity several times, which is unacceptable for edge devices.

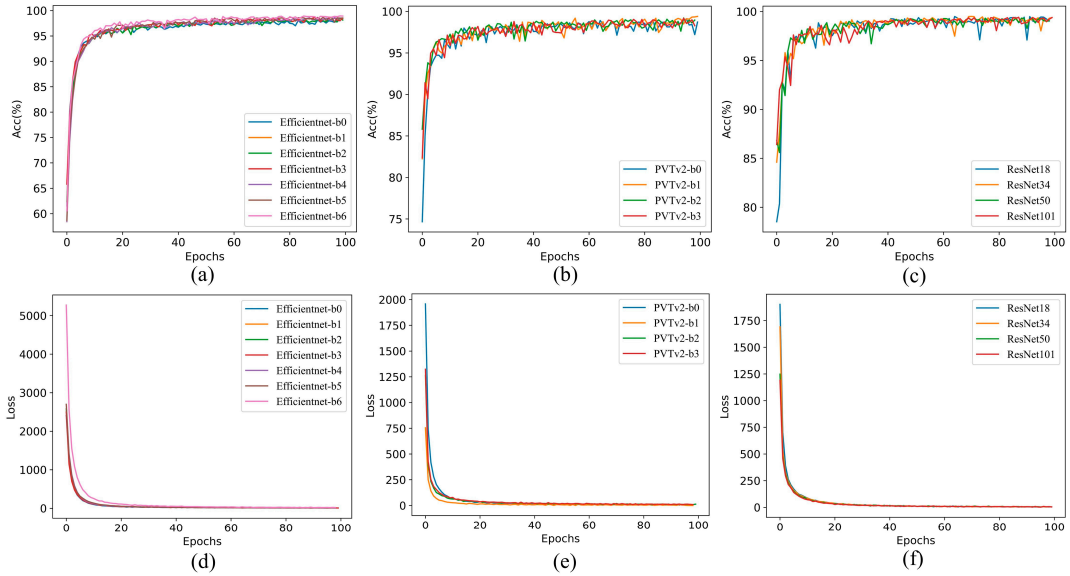


Figure 3. (**a**,**d**) are the experimental results of EfficientNet on the PlantVillage dataset, (**b**,**e**) are the experimental results of PVTv2 on the PlantVillage dataset, and (**c**,**f**) are the experimental results of ResNet on the PlantVillage dataset.

2.2.2. VLDNet

To further investigate the relationship between network model width, depth, and model performance, we propose a lightweight disease recognition model called VLDNet. The overall structure of VLDNet is shown in Figure 4. It includes a VLDBlock, four VLDBottlenecks, an average pooling layer, and a fully connected layer. The VLDBlock is built upon the MobileNet-V1 building block. It consists of a 3×3 depth convolution, followed by a 1×1 pointwise convolution. Each operation is repeated four times. It also introduces parameterized skip connections and Batch Normalization (BN), using the ReLU activation function. The VLDBlock has two different structures during training and testing. During training, it has a branch with 1×1 pointwise convolution and batch normalization. During inference, all parameterized branches are removed by parameterization. The VLDBlock forms the feature extraction network of VLDNet, enhancing the effectiveness and efficiency of feature extraction. Based on the VLDBlock, we propose VLDBottleneck following the idea of ResNet. VLDBottleneck consists of two stacked VLDBlocks. The first VLDBlock serves as an expansion layer to increase the number of channels. The second VLDBlock reduces the number of channels to match the residual connection. A residual connection is added between the input and output of the two VLDBlocks. Each VLDBlock is followed by a BN layer and ReLU activation function, except for the second VLDBlock. A stride = 2 depth convolution is inserted between the two VLDBlocks for downsampling. VLDBottleneck is used to extract disease features while reducing the model's parameter count. The average pooling layer mainly aims to reduce computation and extract essential features. The fully connected layer transforms the feature map into a vector representation of the disease and outputs the disease category.

VLDBlock is a structural reparameterization of the Ghost module [20], shown in Figure 5b. Deep convolutional neural networks often consist of many convolutional layers, which leads to significant computational costs. Although recent works such as MobileNet [21] and ShuffleNet [22] have introduced depthwise convolutions or shuffle operations to construct efficient convolutional neural networks using smaller convolutional kernels, the remaining 1×1 convolutional layers still consume a considerable amount

of memory and parameters. This process is illustrated in Figure 5a and can be expressed mathematically as follows:

$$Y = X * F + B \tag{1}$$

The shape of the input data X is $X \in R^{c \times h \times w}$, where c, h, and w represent the number of channels, height, and width, respectively. The * symbol denotes the convolution operation. B represents the bias term. $Y \in R^{h' \times w' \times n}$ is the output feature map with n channels. $F \in R^{c \times k \times k \times n}$ represents the convolutional kernels in this layer, where h' and w' are the output height and width, respectively, and $k \times k$ is the size of the convolutional kernel F. In this process, the FLOPs are typically very large.

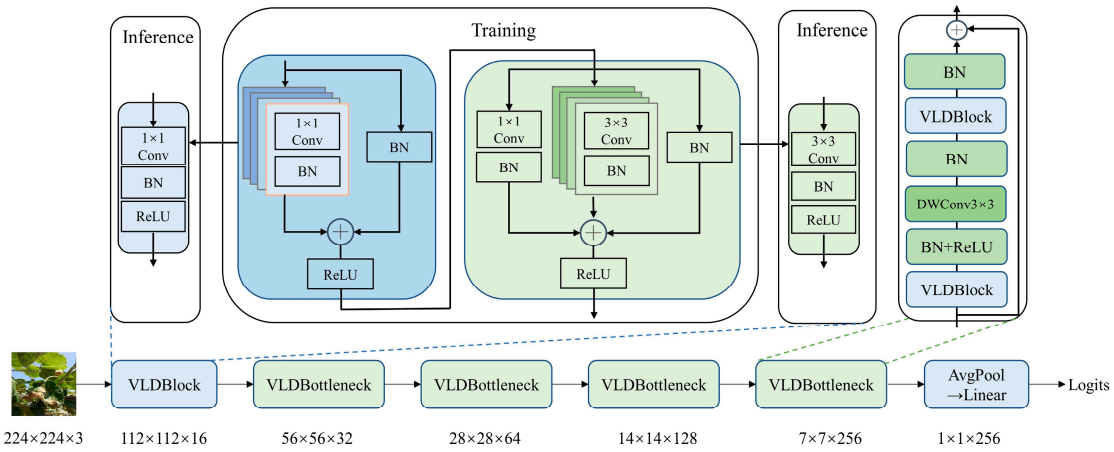


Figure 4. Overall architecture of VLDNet.

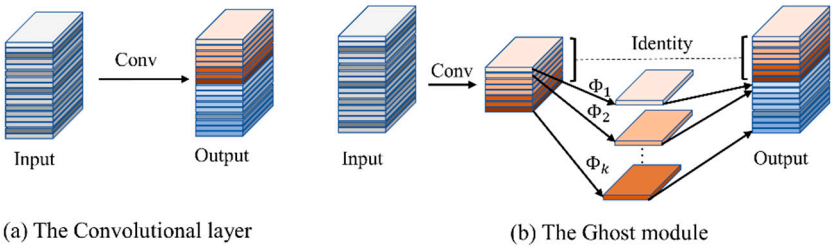


Figure 5. Traditional convolution and Ghost module [20].

In fact, generating redundancy during the calculation of feature maps is necessary for the performance of the network, as seen in Figure 6, where there are many similar feature maps. However, it is not necessary to generate these redundant feature maps one by one using a large number of parameters and FLOPs. Therefore, this paper uses a simple linear transformation to achieve mutual conversion between redundant feature maps, as shown in Figure 5b.

The intrinsic feature maps and redundant feature maps together constitute the feature maps. The process of generating m inherent feature maps $Y' \in R^{h' \times w' \times m}$ is shown in Equation (2). The inherent feature maps are the remaining feature maps after subtracting redundant feature maps from the total feature maps.

$$Y' = X * F' \tag{2}$$

This process is generated by a primary convolution, with convolution kernel parameters identical to those in Equation (2). Here, $F' \in R^{c \times k \times k \times m}$ represents the convolution kernel used. The process of generating n redundant feature maps from m inherent feature maps using an inexpensive linear transformation is described by Equation (3):

$$y_{ij} = \Phi_{i,j}(y'_i), \ \forall \ i = 1, \ldots, m, \ j = 1, \ldots s. \tag{3}$$

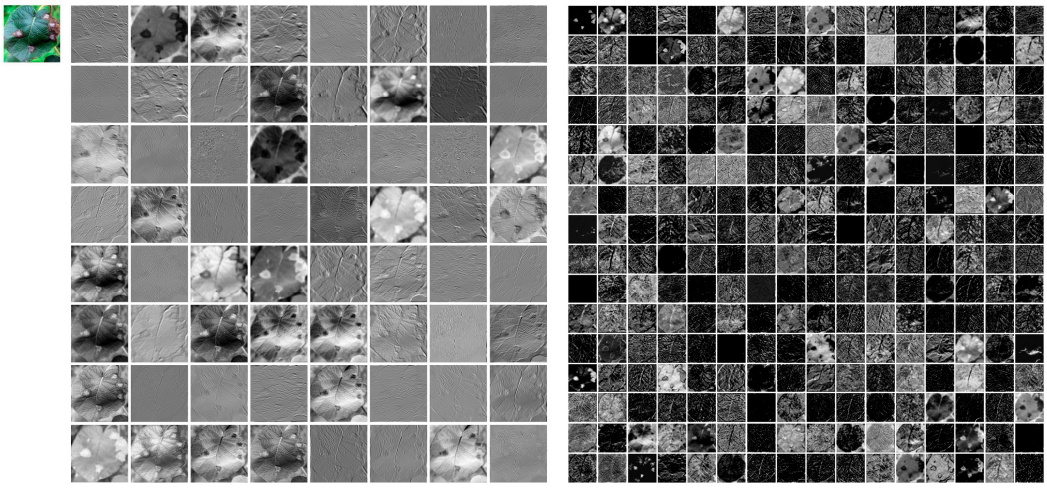


Figure 6. Redundant feature maps generated during the convolution operation.

Here, y'_i represents the i-th inherent feature map in Y', and $\Phi_{i,j}$ is the j-th linear transformation used to generate the j-th feature map y_{ij} in Equation (3). Using Equation (3), we can obtain $n = m * s$ feature maps $Y = [y_{11}, y_{12}, \ldots y_{ms}]$. Linear transformations operate on each channel, and their computational cost is much lower than that of normal convolutions. The structure of the linear transformations is shown in Figure 5b.

Although reducing FLOPs and parameters may lower the computational complexity of the model, ref. [23] has shown that these metrics are not well correlated with the efficiency of the model. This is because metrics like FLOPs do not take into account memory access costs and parallelism, which can have a significant impact on latency during inference [24]. Therefore, this paper proposes the reparameterization of structure Figure 5b to build the basic structure VLDBlock of VLDNet, in order to further reduce the cost and inference time of the model.

The structural reparameterization technique [25,26] is an effective neural network technique that decouples training and inference, greatly facilitating the deployment of deep neural networks in practical applications. During training, for a given backbone network, the structural reparameterization technique increases the model's representational power by adding multiple branches or specific layers with various neural network components to the backbone network. During inference, the added branches or layers can be merged into the backbone network's parameters through equivalent transformations, significantly reducing the number of parameters or computational costs without affecting performance and accelerating inference.

In this paper, during training, for a convolution layer with kernel size K = {1,3}, input channels C_{in}, output channels C_{out}, the weight matrix can be represented as $W' \in \mathbb{R}^{C_{out} \times C_{in} \times K \times K}$, and the bias represented as $B' \in \mathbb{R}^{D}$. The BatchNorm (BN) layer includes accumulated mean μ, variance σ, and bias β. As convolution layers and BN are linear operations during inference, they can be merged, and the corresponding weights are $\hat{W} = W' * \frac{\gamma}{\sigma}$, bias is $\hat{B} = (B' - \mu) * \frac{\gamma}{\sigma} + \beta$, where γ is the scaling factor. For skip connections, BN is merged into identity 1×1 kernel with 0-padding. After merging BN into each branch, the corresponding weight matrix is $W = \sum_i^M \hat{W}_i$, and the bias is $B = \sum_i^M \hat{B}_i$, where M is the number of network branches, as shown in Figure 7. This way, the number of parameters and computational cost of the model are significantly reduced, and the inference speed is also increased.

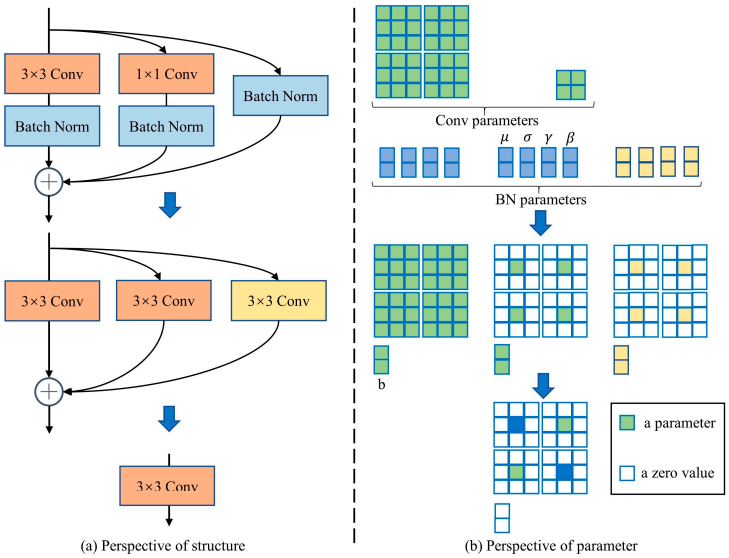


Figure 7. Reparameterization process.

3. Results

3.1. Evaluation Indicators and Experimental Parameter Settings

3.1.1. Evaluation Indicators

The evaluation metrics used in this paper include *accuracy*, *balanced accuracy*, *recall*, *precision*, *F1 score*, *geometric mean*, *parameters*, and *FLOPs*. This is calculated as follows:

$$Accuracy = (TP + TN)/(TP + TN + FP + FN) \tag{4}$$

$$Precision = TP/(TP + FP) \tag{5}$$

$$Recall = TP/(TP + FN) \tag{6}$$

$$F1\ Score = 2/((1/precision) + (1/recall)) \tag{7}$$

$$FLOPs = 2hw \times \left(C_{in} \times K^2 + 1\right) \times C_{out} \tag{8}$$

$$Specificity = TN/(TN + FP) \tag{9}$$

$$Balanced\ Accuracy = \frac{1}{n}\sum_{1}^{n} Accuracy_i, i\epsilon[1, n] \tag{10}$$

$$Geometric\ Mean = \sqrt{Specificity \times Recall} \tag{11}$$

TP represents true positives, *FP* represents false positives, *TN* represents true negatives, and *FN* represents false negatives. Accuracy represents the proportion of correctly classified samples. *Balanced accuracy* is a measure of the proportion of correctly classified data on imbalanced datasets. n is the number of classes. *Recall* represents the proportion of all positive samples that are correctly identified by the classifier. *Precision* represents the proportion of samples that the classifier correctly identifies as positive out of all the samples it classifies as positive. The *F1 Score* is a measure that takes into account both precision and recall. *Geometric mean* calculates the geometric mean of the sensitivity of each class to take into account the predictive power of the model on different classes. Parameters refer to the number of adjustable parameters in a model, including weights and biases. Smaller numbers of parameters mean less requirements for hardware. *FLOPs* represent the computational complexity of a model, with lower values indicating simpler calculations. Among the formula variables, h, w, and C_{in} represent the height, width, and number of channels of input feature maps, respectively, while C_{out} represents the number of output feature map channels, and K represents the convolution kernel width. By selectively

controlling *FLOPs* and *parameters*, we can reduce the size of a model while maintaining its performance.

3.1.2. Experimental Parameter Setting

The experiment was conducted on Ubuntu 20.04 with an Intel Core i9 10900X processor, 48GB RAM (Dell T5820 graphics workstation, Round Rock, TX, USA), and two GeForce RTX 3090 GPUs (NVIDIA, Santa Clara, CA, USA). The deep learning framework used was PyTorch, and training was carried out using Cuda 11.1. Please refer to Table 2 for information on the other settings.

Table 2. Experimental parameter configuration.

Config	Value
Optimizer	Adam
Loss function	CrossEntropyLoss
Initial learning rate	0.0001
Momentum	0.0005
Weight decay	0.05
Dropout	0.6
Batch size	64
Learning rate schedule	cosine decay
Training epochs	250
Image resolution	224 × 224

3.2. Experimental Results on PlantVillage Dataset

To validate the effectiveness of our proposed VLDNet model, we conducted experiments on the PlantVillage dataset and compared the results with those of recent studies. As shown in Table 3, our model achieved an impressive identification accuracy of 99.26%, far surpassing a range of large-scale models. For instance, compared to VGG16, reported by [27], VLDNet achieved an accuracy that was 17.43% higher, with an almost negligible parameter count. Compared to DECA-ResNet18 [28], which achieved the highest identification accuracy on this dataset, our model's identification accuracy was only 0.64% lower, but our model's parameter count was reduced by 499 times. This demonstrates the advantage of VLDNet in model lightweighting. VLDNet achieves high identification accuracy and has fewer parameters, making it ideal for deployment on resource-limited edge devices.

Table 3. Comparison of results between VLDNet and other studies on PlantVillage dataset.

Study	Year	Dataset	Method	Accuracy (%)	Parameters (MB)
[29]	2017	PlantVillage	Inception-V3	80	23.83
[30]	2018	PlantVillage	MobileNet	92	3.3
[27]	2019	PlantVillage	VGG16	81.83	138.3
[31]	2019	Tomato leaf disease	MobileNet	88.4	3.3
[32]	2020	PlantVillage	INC-VGGN	91.83	-
[33]	2020	PlantVillage	VGG16	97.82	138.3
[33]	2020	PlantVillage	GoogleNet	95.3	6.62
[33]	2020	PlantVillage	Resnet50	95.38	25.5
[28]	2021	PlantVillage	DECA-ResNet18	99.74	48.6
[34]	2022	Part of PlantVillage	RIC-Net	99.55	19.1
[35]	2022	PlantVillage	VGG-ICNN	99.16	6
[36]	2023	PlantVillage	MobileNetV2	99.10	2.3
[36]	2023	PlantVillage	MDCDenseNet	99.40	7.3
Ours	2023	PlantVillage	VLDNet	99.26	0.097

3.3. Experimental Results on Self-Built Dataset

3.3.1. Comparison between the Proposed Model and Lightweight SOTA

To further validate the performance of the VLDNet model, we conducted experiments on a self-built dataset. As shown in Table 4, VLDNet achieved an identification accuracy of 98.32%, outperforming a range of widely used lightweight CNNs such as MobilenetV2 (96.17%), MobilenetV3 (96.70%) [37], etc., with identification accuracy that was higher than them by 2.15% and 1.62%, respectively, while having a smaller model size. Compared to Swin-Tiny, which has the best identification accuracy (98.77%), VLDNet's identification accuracy was only 0.45% lower, while its parameter count, and FLOPs were reduced by 297 and 111 times, respectively. This, once again, demonstrates the advantage of VLDNet in lightweighting. Compared to the smallest MobileViT-XXS [38] (97.97%), VLDNet achieved an identification accuracy that was 0.35% higher, but its parameter count and FLOPs were only 7.6% and 16% of it, respectively. It can be seen that VLDNet achieves a good balance between computational efficiency and identification accuracy of the model.

Table 4. Comparison of identification results with other lightweight models on self-built datasets.

Model	Parameters (MB)	FLOPs (G)	Acc (%)	Balanced Acc	Precision (%)	Recall (%)	F1 Score	G-Mean
Resnet18	11.5	1.71	98.32	0.9823	98.29	98.32	0.9831	0.9833
MobilenetV2	2.23	0.32	96.17	0.9608	96.15	96.17	0.9616	0.9617
MobilenetV3	5.4	0.22	96.70	0.9661	96.68	96.72	0.9670	0.9671
EfficientNet-B0	5.3	0.41	96.30	0.9621	96.57	96.30	0.9643	0.9631
EfficientNet-B1	7.73	0.56	96.82	0.9673	96.65	96.82	0.9673	0.9683
DeiT-Tiny	5.68	1.05	96.47	0.9638	96.36	96.47	0.9641	0.9648
ViT-Tiny	9.70	1.06	96.91	0.9682	96.78	96.90	0.9684	0.9691
Swin-Tiny	29	4.5	98.77	0.9848	98.80	98.71	0.9875	0.9875
PVT-Tiny	12.33	1.82	97.71	0.9752	97.65	97.71	0.9768	0.9772
PVTV2-b0	3.67	0.52	98.41	0.9812	98.39	98.46	0.9842	0.9847
PVTV2-b1	14.01	1.99	98.73	0.9864	98.76	98.79	0.9878	0.9879
MobileViT-XXS	1.27	0.25	97.97	0.9774	98.01	98.01	0.9801	0.9802
MobileViT-XS	2.31	0.69	98.68	0.9849	98.63	98.70	0.9866	0.9870
MobileViT-S	5.57	1.39	98.73	0.9834	98.74	98.73	0.9874	0.9873
MobileViTV2-50	1.36	0.35	97.18	0.9707	97.19	97.22	0.9721	0.9723
MobileViTV2-75	2.85	0.78	97.71	0.9732	97.63	97.67	0.9765	0.9768
MobileViTV2-100	4.88	1.38	98.68	0.9839	98.65	98.67	0.9866	0.9868
ConViT-Ti	9.5	0.98	96.91	0.9622	96.96	96.91	0.9693	0.9691
Ours	0.097	0.04	98.32	0.9813	98.30	98.32	0.9831	0.9833

3.3.2. Comparison between the Proposed Model and Heavyweight SOTA

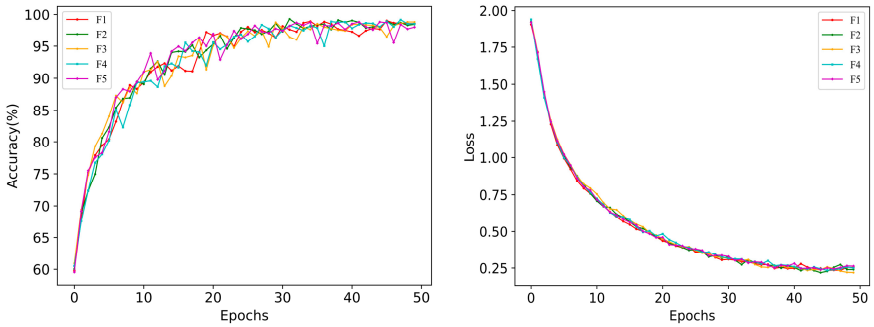
Table 5 shows the comparative experimental results of VLDNet with other heavyweight SOTA models on our self-built dataset. We can see that, even when facing models with much larger parameters and computational complexity, VLDNet's identification performance is still competitive. For example, it achieved higher identification accuracy than a range of large-scale models, such as ViT-base, PVTV2-b5 [39], ConViT-Base [40], etc. This validates our hypothesis that deeper and wider network structures may not necessarily achieve better results in crop disease identification tasks. Compared to the best performing model VGG16, VLDNet's identification accuracy was only 1.06% lower, but at a very high cost of resource consumption and computation, which is unacceptable for edge devices. However, VLDNet achieves a high identification accuracy with very little resource consumption and consumption, which is very friendly to edge devices.

Table 5. Comparison of identification results with other heavyweight models on self-built dataset.

Model	Parameters (MB)	FLOPs (G)	Acc (%)	Balanced Acc	Precision (%)	Recall (%)	F1 Score	G-Mean
ResNet-101	44.55	7.68	98.78	0.9859	98.86	98.71	0.9875	0.9874
VGG16	138	15.5	99.38	0.9919	99.35	99.42	0.9938	0.9943
Densenet121	7.9	2.77	97.35	0.9706	97.38	97.26	0.9732	0.9728
DeiTBase	86.56	16.47	96.21	0.9612	96.27	96.08	0.9617	0.9673
ViT-base	86.56	16.47	96.38	0.9619	96.37	96.36	0.9638	0.9637
PVT-Large	61.4	9.8	98.41	0.9822	98.48	98.32	0.9844	0.9843
PVTV2-b4	62.56	9.59	98.94	0.9856	98.91	98.93	0.9892	0.9893
PVTV2-b5	81.96	11.12	98.24	0.9815	98.25	98.24	0.9825	0.9825
Swin-Base	87.7	14.81	98.79	0.9830	98.81	98.73	0.9877	0.9875
ConViT-Base	86.39	16.42	97.35	0.9726	97.38	97.26	0.9732	0.9729
Ours	0.097	0.04	98.32	0.9813	98.30	98.32	0.9831	0.9833

3.4. Five-Fold Cross-Validation on Self-Built Dataset

In order to further validate the performance of VLDNet, we conducted a five-fold cross-validation on our self-built dataset. The experimental parameters were set to the default values provided in Table 2. A total of 50 epochs of experiments were performed, and the results of the test set are shown in Figure 8. F1–F5 represent models trained using different validation sets. It can be observed that the models trained with different validation sets converge quickly. The final identification accuracy rate is consistently above 98%, which is essentially consistent with the previous experimental results. This demonstrates the excellent performance of VLDNet.



(**a**) (**b**)

Figure 8. (**a**) is the accuracy on the test set. (**b**) is the loss on the test set.

3.5. Do Deeper and Wider Networks Lead to Better Identification Results?

The experimental results in Sections 2.2.1 and 3.3.2 have already shown that deeper and wider networks may not necessarily improve disease identification accuracy, and sometimes may even lead to a decrease in accuracy. For example, on our self-built dataset, PVTV2-b5 had a lower identification accuracy than PVTV2-b0. To further validate this issue, we conducted additional experiments in this section. The experiments were conducted on our self-built dataset using VLDNet, with the network width controlled by the parameter $\alpha = \{1,2,4\}$, which increased the network width by multiplying the number of channels in each layer by α, while keeping the network depth constant. The experimental results are shown in Figure 9, which shows that there is no difference in identification accuracy between models of different widths, and that shallower networks actually converge faster. This once again confirms our hypothesis and demonstrates the effectiveness of VLDNet's design.

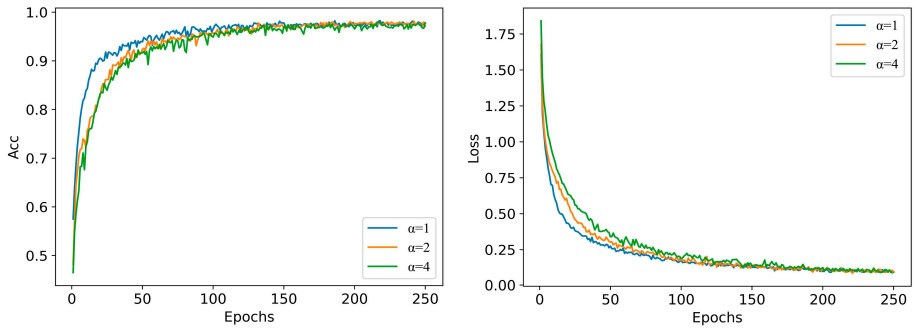


Figure 9. Identification accuracy and loss of VLDNet models with different widths.

3.6. Model Inference Time Testing

In this section, we tested the inference time of different models on an NVIDIA GTX 1650Intel (R), Core (TM) i7-10700 CPU @ 2.90 GHz. As shown in Figure 10, the red arrow points to the VLDNet model, which has the fastest inference speed while maintaining a high identification accuracy of 98.32%. In actual measurements, VLDNet can recognize 221 images per second, with an average of 4.52 ms per image, which is more than 37% faster than Resnet18 (6.2 ms), which has the closest inference speed. Compared to PVTV2-b5 (98.24%) with a similar identification accuracy, the inference speed of VLDNet is increased by 11.6 times. Compared to VGG16 (99.38%), which has the best identification performance, although the identification accuracy of VLDNet is 1.06% lower, its inference speed is increased by 8.2 times. VLDNet not only has fewer model parameters and lower FLOPs, but also has a fast inference speed, thanks to the use of inexpensive operations to supplement redundant features and the operation of reparameterization.

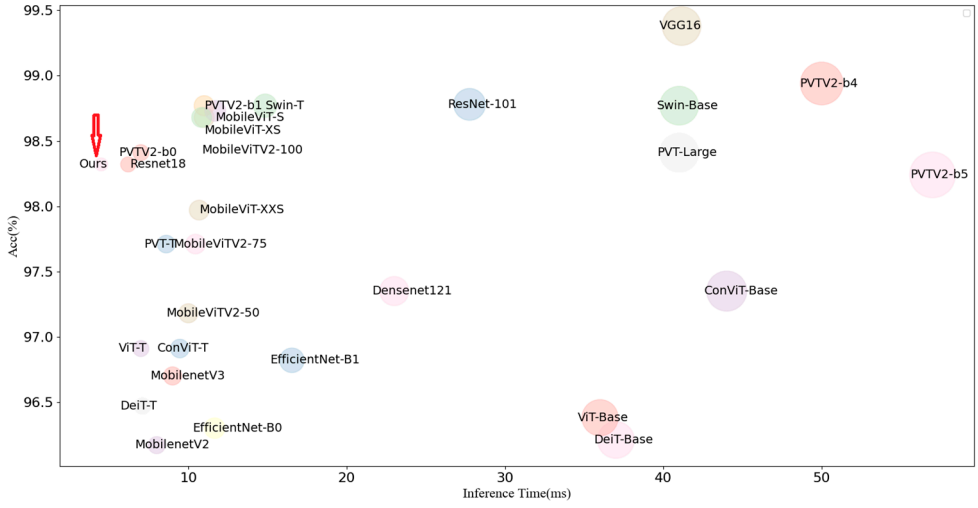


Figure 10. Inference time—identification accuracy for different models.

3.7. Ablation Experiments

In order to validate the necessity and effectiveness of each design in VLDNet, this section conducted ablation experiments on our self-built dataset. The experimental results are shown in Table 6. Even with a shallow and narrow ordinary model without using linear operations to supplement redundant features and reparameterization, an identification accuracy as high as 98.33% can be achieved while keeping the model parameters and FLOPs low. This result again validates our hypothesis about the relationship between model depth, width, and performance. When linear operations are used to supplement redundant features, the number of model parameters and FLOPs is reduced to half of the original, while the identification accuracy is not significantly affected, indicating the effectiveness of

this operation. Furthermore, based on this, using structural reparameterization reduces the number of model parameters and FLOPs to 12% and 11% of the original, respectively. The identification accuracy of the model also did not change significantly. The ablation experiment verifies that each design in VLDNet is effective and necessary.

Table 6. Results of ablation experiments.

Linear Operation	Reparameterization	Parameters (MB)	FLOPs (G)	Accuracy (%)
		0.747	0.354	98.35
√		0.373	0.176	98.33
√	√	0.097	0.040	98.32

3.8. Visual Display of Identification Results

In this section, the Grad-CAM method was used for visualization to observe the classification basis of the VLDNet model. The experimental results are shown in Figure 11. Grad-CAM [41] is a deep neural network visualization method based on gradient localization. It calculates the weight of each feature map in the last convolutional layer for the image category and obtains the weighted sum of each feature map. Then, it maps the weighted feature maps to the original image in the form of a heatmap to explain the classification basis of the deep neural network model. Figure 10 shows that the VLDNet model accurately focuses on the area where the disease occurs in each disease image, which is consistent with our judgment basis. This indicates that the VLDNet model has good performance in crop disease classification.

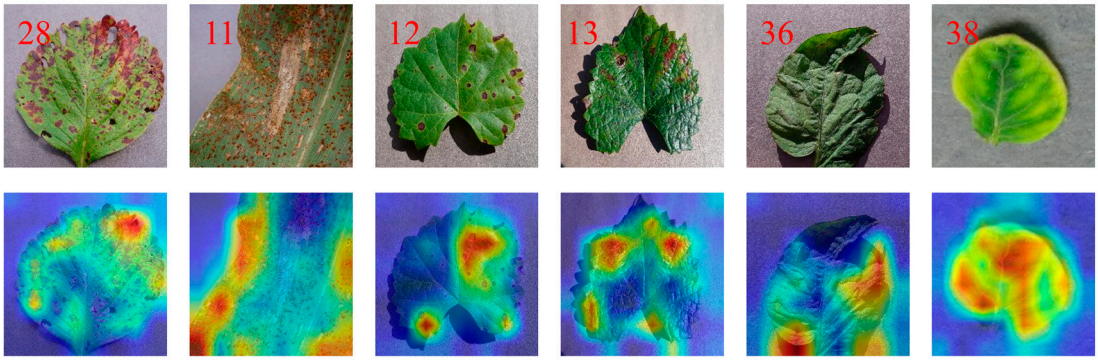


Figure 11. Visual display of identification results.

4. Discussion

This paper rethinks the relationship between the depth and width design of disease identification models and model performance. It found that the commonly used paradigm of designing models with wide and deep structure to improve identification accuracy is not applicable in disease identification tasks. Based on this finding, this paper proposes the lightweight disease identification model VLDNet, which achieves a good balance between efficiency and accuracy. The paper first experiments with ResNet and other models on the public dataset PlantVillage to verify their hypothesis about the non-strict correlation between the depth and width design of disease identification models and their performance. Based on this, the paper proposes the VLDNet, which uses the basic module VLDBlock to extract inherent features using 1×1 convolutional operations, and then supplements redundant features using inexpensive linear operations, thereby improving the efficiency of feature extraction. The paper also uses structural reparameterization during inference to reduce the number of model parameters and FLOPs and accelerate model inference. VLDNet achieves good identification results on both the PlantVillage and self-built datasets, with fewer model parameters, simpler computation, and faster inference speed. In addition, the paper conducts ablation experiments to verify the necessity and effectiveness of each

design of VLDNet, and the visualization graphs also prove that VLDNet can accurately focus on diseased areas.

Compared to large-scale models such as [28,33,34], our model performs better or on par with them in terms of identification accuracy. Although the differences in identification accuracy are not significant, our model has much fewer parameters and FLOPs than theirs, and its inference speed is faster. Some works choose to use strategies such as pruning, quantization, and distillation to obtain lightweight models [27,42]. Although these methods can reduce the number of model parameters and computational complexity, the identification accuracy of the model may decrease. In addition, even if large-scale models are lightweighted through these strategies, it is still difficult to achieve complete lightweighting. In contrast, our model has the same identification accuracy as large-scale models, while having much fewer parameters and FLOPs than them. There are also some works that specialize in designing lightweight models to reduce params and FLOPs, such as [16,17], and have achieved high identification accuracy, but their actual inference speed needs to be verified. In addition, the lightweight network designed by [16] achieved high recognition accuracy in complex backgrounds, but the number of model parameters and FLOPs still cannot be compared with ours. Ref. [18] and our model achieved real-time identification effects, but empirical evidence suggests that our model has higher inference efficiency, which means that our model has lower device requirements and a wider range of applications. Our model uses structural reparameterization for optimization, but this requires redesigning and adjusting the structure and parameters in the neural network, which is more complex than traditional neural network models. Additionally, further improvement is needed in our model's support for large-scale data.

This paper discovers that the paradigm of designing models with wide and deep structure to improve identification accuracy is not applicable in disease identification tasks. The authors hope that researchers can further verify this discovery and apply it to guide the design of disease models. The proposed lightweight identification model VLDNet has very low requirements for deployment devices, which will help accelerate the deployment process of deep learning-based disease identification models on edge devices and promote the development of smart agriculture.

5. Conclusions

This paper proposes a rethinking of the design paradigm for crop disease identification models based on deep learning. The experiment verifies that there is a non-strict correlation between the depth and width design of disease identification models and their performance. Based on this, the paper designs the VLDNet, a lightweight disease identification model that achieves a good balance between efficiency and performance. VLDNet performed well on both public and self-built datasets. This discovery is crucial for the efficient design of models, especially for the deployment of deep learning-based identification methods on edge devices in smart agriculture. Our future research direction is to apply structure reparameterization techniques to Vision Transformers. This is because Vision Transformers have stronger expressive power, which has the potential to further enhance the performance of disease recognition tasks based on deep learning. We will continue to explore efficient model design methods, improve and optimize VLDNet, and better serve the development of smart agriculture.

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

Funding: This work was supported by National Key Research and Development Program of China (Grants number 2020YFD1100600, Grants number 2020YFD1100601).

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

Acknowledgments: We thank all of the funders.

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

Appendix A

Table A1. The PlantVillage dataset Label ID and corresponding Label Name.

Label ID	Label Name	Label ID	Label Name
1	Apple_Apple_scab	21	Potato_Early_blight
2	Apple_Black_rot	22	Potato_healthy
3	Apple_Cedar_apple_rust	23	Potato_Late_blight
4	Apple_healthy	24	Raspberry_healthy
5	Blueberry_healthy	25	Soybean_healthy
6	Cherry_(including_sour)_healthy	26	Squash_Powdery_mildew
7	Cherry_(including_sour)_Powdery_mildew	27	Strawberry_healthy
8	Corn_(maize)_Cercospora_leaf_spot Gray_leaf_spot	28	Strawberry_Leaf_scorch
9	Corn_(maize)_Common_rust_	29	Tomato_Bacterial_spot
10	Corn_(maize)_healthy	30	Tomato_Bacterial_spot
11	Corn_(maize)_Northern_Leaf_Blight	31	Tomato_healthy
12	Grape_Black_rot	32	Tomato_Late_blight
13	Grape_Esca_(Black_Measles)	33	Tomato_Leaf_Mold
14	Grape_healthy	34	Tomato_Septoria_leaf_spot
15	Grape_Leaf_blight_(Isariopsis_Leaf_Spot)	35	Tomato_Spider_mites Two-spotted_spider_mite
16	Orange_Haunglongbing_(Citrus_greening)	36	Tomato_Target_Spot
17	Peach_Bacterial_spot	37	Tomato_Tomato_mosaic_virus
18	Peach_healthy	38	Tomato_Tomato_Yellow_Leaf_Curl_Virus
19	Pepper_bell_Bacterial_spot		
20	Pepper_bell_healthy		

References

1. Sharif, M.; Khan, M.A.; Iqbal, Z.; Azam, M.F.; Lali, M.I.U.; Javed, M.Y. Detection and classification of citrus diseases in agriculture based on optimized weighted segmentation and feature selection. *Comput. Electron. Agric.* **2018**, *150*, 220–234. [CrossRef]
2. Patil, J.K.; Kumar, R. Analysis of content based image retrieval for plant leaf diseases using color, shape and texture features. *Eng. Agric. Environ. Food* **2017**, *10*, 69–78. [CrossRef]
3. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
4. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
5. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, 5–9 October 2015.
6. Ferentinos, K.P. Deep learning models for plant disease detection and diagnosis. *Comput. Electron. Agric.* **2018**, *145*, 311–318. [CrossRef]
7. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
8. Barbedo, J.G.A. Plant disease identification from individual lesions and spots using deep learning. *Biosyst Eng.* **2019**, *180*, 96–107. [CrossRef]
9. Huang, S.; Liu, W.; Qi, F.; Yang, K. Development and validation of a deep learning algorithm for the recognition of plant disease. In Proceedings of the 2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), Changsha, China, 10–12 August 2019.
10. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
11. Li, X.; Chen, X.; Yang, J.; Li, S. Transformer helps identify kiwifruit diseases in complex natural environments. *Comput. Electron. Agric.* **2022**, *200*, 107258. [CrossRef]

12. Thakur, P.S.; Khanna, P.; Sheorey, T.; Ojha, A. Explainable vision transformer enabled convolutional neural network for plant disease identification: PlantXViT. *arXiv* **2022**, arXiv:2207.07919.
13. Li, X.; Li, S. Transformer Help CNN See Better: A Lightweight Hybrid Apple Disease Identification Model Based on Transformers. *Agriculture* **2022**, *12*, 884. [CrossRef]
14. Li, W.; Zhang, L.; Wu, C.; Cui, Z.; Niu, C. A new lightweight deep neural network for surface scratch detection. *Int. J. Adv. Manuf. Technol.* **2022**, *123*, 1999–2015. [CrossRef]
15. Li, X.; Zhang, Y.; Li, S. SheepFaceNet: A Speed–Accuracy Balanced Model for Sheep Face Recognition. *Animals* **2023**, *13*, 1930. [CrossRef] [PubMed]
16. Kamal, K.C.; Yin, Z.; Wu, M.; Wu, Z. Depthwise separable convolution architectures for plant disease classification. *Comput. Electron. Agric.* **2019**, *165*, 104948.
17. Bao, W.; Yang, X.; Liang, D.; Hu, G.; Yang, X. Lightweight convolutional neural network model for field wheat ear disease identification. *Comput. Electron. Agric.* **2021**, *189*, 106367. [CrossRef]
18. Khan, A.I.; Quadri, S.M.K.; Banday, S.; Shah, J.L. Deep diagnosis: A real-time apple leaf disease detection system based on deep learning. *Comput. Electron. Agric.* **2022**, *198*, 107093. [CrossRef]
19. Hughes, D.; Salathé, M. An open access repository of images on plant health to enable the development of mobile disease diagnostics. *arXiv* **2015**, arXiv:1511.08060.
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 (CVPR), Seattle, DC, USA, 14–19 June 2020.
21. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
22. 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 (CVPR), Salt Lake City, UT, USA, 18–22 June 2018.
23. Dehghani, M.; Arnab, A.; Beyer, L.; Vaswani, A.; Tay, Y. The efficiency misnomer. *arXiv* **2021**, arXiv:2110.12894.
24. Ma, N.; Zhang, X.; Zheng, H.T.; Sun, J. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018.
25. Ding, X.; Guo, Y.; Ding, G.; Han, J. Acnet: Strengthening the kernel skeletons for powerful cnn via asymmetric convolution blocks. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019.
26. Ding, X.; Zhang, X.; Ma, N.; Han, J.; Ding, G.; Sun, J. Repvgg: Making vgg-style convnets great again. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 19–25 June 2021.
27. Too, E.C.; Yujian, L.; Njuki, S.; Yingchun, L. A comparative study of fine-tuning deep learning models for plant disease identification. *Comput. Electron. Agric.* **2019**, *161*, 272–279. [CrossRef]
28. Gao, R.; Wang, R.; Feng, L.; Li, Q.; Wu, H. Dual-branch, efficient, channel attention-based crop disease identification. *Comput. Electron. Agric.* **2021**, *190*, 106410. [CrossRef]
29. Wang, G.; Sun, Y.; Wang, J. Automatic image-based plant disease severity estimation using deep learning. *Comput. Intell. Neurosci.* **2017**, *2017*, 2917536. [CrossRef]
30. Gandhi, R.; Nimbalkar, S.; Yelamanchili, N.; Ponkshe, S. Plant disease detection using CNNs and GANs as an augmentative approach. In Proceedings of the 2018 IEEE International Conference on Innovative Research and Development (ICIRD), Bangkok, Thailand, 11–12 May 2018.
31. Elhassouny, A.; Smarandache, F. Smart mobile application to recognize tomato leaf diseases using Convolutional Neural Networks. In Proceedings of the 2019 International Conference of Computer Science and Renewable Energies (ICCSRE), Agadir, Morocco, 22–24 July 2019.
32. Chen, J.; Chen, J.; Zhang, D.; Sun, Y.; Nanehkaran, Y.A. Using deep transfer learning for image-based plant disease identification. *Comput. Electron. Agric.* **2020**, *173*, 105393. [CrossRef]
33. Mohameth, F.; Bingcai, C.; Sada, K.A. Plant disease detection with deep learning and feature extraction using plant village. *J. Comput. Commun.* **2020**, *8*, 10–22. [CrossRef]
34. Zhao, Y.; Sun, C.; Xu, X.; Chen, J. RIC-Net: A plant disease classification model based on the fusion of Inception and residual structure and embedded attention mechanism. *Comput. Electron. Agric.* **2022**, *193*, 106644. [CrossRef]
35. Thakur, P.S.; Sheorey, T.; Ojha, A. VGG-ICNN: A Lightweight CNN model for crop disease identification. *Multimed. Tools Appl.* **2023**, *82*, 497–520. [CrossRef]
36. Li, E.; Wang, L.; Xie, Q.; Gao, R.; Su, Z.; Li, Y. A novel deep learning method for maize disease identification based on small sample-size and complex background datasets. *Ecol. Inform.* **2023**, *75*, 102011. [CrossRef]
37. Howard, A.; Sandler, M.; Chu, G.; Chen, L.C.; Chen, B.; Tan, M.; Adam, H. Searching for mobilenetv3. In Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR), Long Beach, CA, USA, 16–20 June 2019.
38. Mehta, S.; Rastegari, M. Mobilevit: Light-weight, general-purpose, and mobile-friendly vision transformer. *arXiv* **2021**, arXiv:2110.02178.
39. Wang, W.; Xie, E.; Li, X.; Fan, D.P.; Song, K.; Liang, D.; Shao, L. Pvt v2: Improved baselines with pyramid vision transformer. *Comput. Vis. Media* **2022**, *8*, 415–424. [CrossRef]

40. D'ascoli, S.; Touvron, H.; Leavitt, M.L.; Morcos, A.S.; Biroli, G.; Sagun, L. Convit: Improving vision transformers with soft convolutional inductive biases. In Proceedings of the International Conference on Machine Learning (ICML), Online, 18–24 July 2021.
41. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017.
42. Arun, R.A.; Umamaheswari, S. Effective multi-crop disease detection using pruned complete concatenated deep learning model. *Expert Syst. Appl.* **2023**, *213*, 118905. [CrossRef]