


## Article

# In-Field Citrus Disease Classification via Convolutional Neural Network from Smartphone Images

Changcai Yang <sup>1</sup>, Zixuan Teng <sup>1</sup>, Caixia Dong <sup>1</sup>, Yaohai Lin <sup>1</sup>, Riqing Chen <sup>1,\*</sup> and Jian Wang <sup>2</sup><sup>1</sup> College of Computer and Information Science, Fujian Agriculture and Forestry University, Fuzhou 350002, China<sup>2</sup> Ningxia Academy of Agriculture and Forestry Sciences, Yinchuan 750002, China

\* Correspondence: riqing.chen@fafu.edu.cn

**Abstract:** A high-efficiency, nondestructive, rapid, and automatic crop disease classification method is essential for the modernization of agriculture. To more accurately extract and fit citrus disease image features, we designed a new 13-layer convolutional neural network (CNN13) consisting of multiple convolutional layer stacks and dropout in this study. To address the problem created by the uneven number of disease images in each category, we used the VGG16 network module for transfer learning, which we combined with the proposed CNN13 to form a new joint network, which we called OplusVNet. To verify the performance of the proposed OplusVNet network, we collected 1869 citrus pest and disease images and 202 normal citrus images from the field. The experimental results showed that the proposed OplusVNet can more effectively solve the problem caused by uneven data volume and has higher recognition accuracy, especially for image categories with a relatively small data volume. Compared with the state of the art networks, the generalization ability of the proposed OplusVNet network is stronger for classifying diseases. The classification accuracy of the model prediction results was 0.99, indicating the model can be used as a reference for crop image classification.



**Citation:** Yang, C.; Teng, Z.; Dong, C.; Lin, Y.; Chen, R.; Wang, J. In-Field Citrus Disease Classification via Convolutional Neural Network from Smartphone Images. *Agriculture* **2022**, *12*, 1487. <https://doi.org/10.3390/agriculture12091487>

Academic Editor: Xiuliang Jin

Received: 19 July 2022

Accepted: 6 September 2022

Published: 16 September 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

**Keywords:** citrus diseases; classification; convolutional neural network; transfer learning; smartphone image; field image

## 1. Introduction

Citrus fruit is cultivated all over southern China, being the primary cultivated fruit in the country and a main industry in the vast rural areas in south China. However, the areas in which citrus is grown are mostly warm and humid, and the fruit trees have a long growing period, so they are often infected with many diseases. About 50% of citrus fruits are affected by different diseases [1]. Intelligent identification of citrus diseases is an important step in building modern and intelligent agriculture systems, which can provide more scientific and effective guidance for citrus pest and disease control and field management [2,3]. The occurrence of crop diseases is affected by seasonal and climatic factors, resulting in long image data collection cycles and uneven distribution of various pest and disease data, affecting the performance of classification algorithms. In addition, changes in lighting and perspective during image acquisition also pose certain difficulties for classification. Therefore, automatically classifying citrus diseases in fields using smartphone images remains a challenge.

Various traditional computer vision methods have been applied to crop pest and disease image classification [4,5]. Traditional disease image classification usually includes feature extraction (e.g., SIFT, shape, and color features) and building a disease image classifier with some machine-learning algorithms. K. Jagan Mohan et al. [6] used scale-invariant features (SIFT), K-nearest neighbor classifier, and support vector machine (SVM) to identify three rice diseases: brown spot, rice blast, and white leaf blight. The accuracy

of disease identification using SVM was 0.91. In [7], the model learned an overcomplete dictionary to sparsely represent the training images of each leaf species using a sparse representation (SR) approach. This framework was able to effectively recognize leaves on a public leaf dataset. Shanwen Zhang et al. [8] used K-means clustering to segment diseased leaf images, extracted shape and color features to provide disease information, and classified diseased cucumber leaf images using SR. A major advantage of this method is that classification in SR space can effectively reduce computational effort and improve recognition performance: in the study, the overall recognition rate was 0.86. Because hand-crafted features may not be invariant to all diseases, finding an image classifier that is robust to all diseases is difficult. Traditional classification methods may be more accurate for one type of disease, but less so for another type.

To accurately identify crop pests and diseases, a variety of classification methods based on deep learning have been developed [5,9–18]. Mohanty et al. [19] used a deep convolutional neural network architecture to train a model on plant leaf images with the aim of classifying the crop species as well as the presence and identity of the disease. Sladojevic et al. [20] used the Caffe deep-learning framework to build a convolutional neural network model for disease image recognition using plant leaves, and the experimental results showed the model achieved an accuracy of more than 0.91 and 0.96 for the individual classes tested. However, for deep learning, large-scale data are used as the training basis, which is strongly dependent on the amount of image data [21]. Many deep neural networks based on transfer learning have been proposed to solve the problem caused by insufficient data. Thenmozhi and Reddy [22] designed a deep CNN model that can classify insect species using the NBAIR, Xie1, and Xie2 datasets. Selvaraj et al. [23] retrained ResNet50, InceptionV2, and MobileNetV1 to build disease- and pest-detection methods. The experimental results showed that ResNet50 and InceptionV2 outperformed MobileNetV1. They revealed that the DCNN is a robust and easy-to-deploy digital banana disease and pest detection strategy. Coulibaly et al. [24] proposed a transfer-learning-based deep neural network for identifying pearl millet disease, which had an average recognition accuracy of 0.95 and an F1 score of 0.92. Barman et al. [25] compared MobileNet and self-structured CNN for citrus leaf disease classification. They found that the self-structured CNN was more accurate than MobileNet in citrus disease classification using smartphone images. Khanramaki et al. [26] proposed an integrated classifier of deep convolutional neural networks to identify citrus pests. These methods can usually produce accurate results in farming laboratories; however, they may be less effective for the citrus diseases and pests in the field due to the light changes and complex backgrounds.

In this study, based on an analysis of the characteristics of citrus image data collected by mobile devices in the field from 2019 to 2020, we designed a new deep-learning network that combines transfer learning to classify citrus disease and pest images. The original AlexNet architecture has eleven layers. Our proposed convolutional neural network has thirteen layers with a  $3 \times 3$  kernel size for both convolutional and pooling layers, obtaining more nonlinear transformation features from the disease image and reducing the number of parameters of the proposed network. Moreover, we combined the proposed 13-layer convolutional neural network (CNN13) with the pretrained VGG16 to address the problem caused by the uneven distribution of images among disease categories.

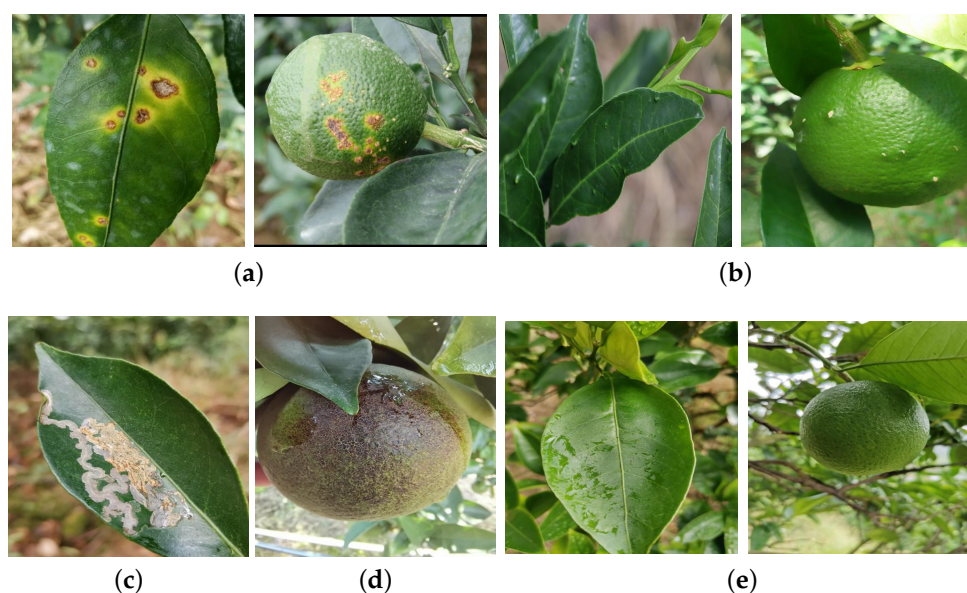
The main contributions of this study are twofold: First, we designed a new CNN13 to more accurately extract and fit citrus disease image features. Second, we designed a new joint network called OplusVNet, which combines the proposed CNN13 with transfer learning to alleviate the problem caused by uneven numbers of disease smartphone image data in each category.

The remainder of this paper is organized as follows: In Section 2, we describe the dataset construction. Section 3 presents the proposed OplusVNet. The citrus disease image classification experimental results are described in Section 4. Finally, we provide our conclusions in Section 5.

## 2. Dataset Construction

Citrus is mainly affected by 8 families and 9 species of diseases and 20 families and 24 species of insect pests, among which the most serious diseases and insect pests include canker, leaf miner, scab, and rusty wall. Citrus disease leaf and fruit image data were obtained by researchers carrying mobile devices in 2019–2020 in citrus plantations in Mingqing County, Fujian Province, China. The disease and pest categories of the image data were manually calibrated by two plant protection experts. The data contained a total of 1869 images of four common citrus pests and diseases, including 1040 images of citrus canker disease, 299 images of citrus scab disease, 320 images of leaf miner insect pest, and 210 images of citrus rusty wall insect pest. In addition, 202 images of normal citrus were collected. Examples of citrus disease and pest leaf and fruit images are shown in Figure 1.

The data distributions of various types of citrus disease and pest images are shown in Table 1. Before training the network, we divided the citrus images of various types of pests and diseases into training, validation, and test sets in a 3:1:1 ratio. We normalized the pixel values of all images so that the pixel values of the images were mapped within the range of 0–1. We used the image data generation class in TensorFlow to augment the data in the images in the training set, which included image horizontal flipping, random angle rotation, panning, and random cropping.



**Figure 1.** Examples of leaves/fruit of citrus pests and diseases. (a) Citrus canker disease. (b) Citrus scab disease. (c) Citrus leaf miner insect-pests. (d) Rust wall insect-pests. (e) Citrus normal leaf/fruit picture.

**Table 1.** Distribution of various data on citrus pests and diseases.

|                | Canker | Scab | Leaf Miner | Rust Wall | Normal |
|----------------|--------|------|------------|-----------|--------|
| Total          | 1040   | 293  | 320        | 210       | 202    |
| Training set   | 624    | 176  | 192        | 126       | 122    |
| Validation set | 208    | 59   | 64         | 42        | 40     |
| Test set       | 208    | 58   | 64         | 42        | 40     |

## 3. Methods

The proposed OplusVNet model contains the VGG16 transfer learning network and the proposed CNN13.

### 3.1. Network-Based Transfer Learning

In this study, we introduced transfer learning to solve the problem of insufficient training in citrus images and the uneven number of images of various diseases. The proposed network-based transfer learning method is shown in Figure 2, where a trained deep network in the source domain is transformed into part of our proposed CNN13 that we used to train the target domain. In this study, we used the VGG16 network [27] as the source domain network for transfer learning, and the migrated part includes the network structure and connection parameters of VGG16. The VGG16 model uses ImageNet data as training data, and its final output has 1000 classes. However, in this study, we had only five classes of pests and diseases, so the migration part does not include the fully connected layer and output unit of the VGG16 network. We combined the VGG16 transfer learning network with our proposed CNN13 to form a new network named OplusVNet. Some layers of the VGG16 network are frozen. The parameters of these frozen layers are not updated during the training phase of the OplusVNet network. At this point, the frozen layer can be regarded as a whole as a feature extractor.

To facilitate the construction of the OplusVNet network, we retained the nonfrozen layers of the VGG16 network. We use OplusVNet\_10, which is a frozen 10-layer network, as an example to illustrate the construction form of the proposed transformed learning network. Table 2 shows the structure of the VGG16 transformed learning network in OplusVNet\_10 and its parameters. We set the input image size to  $512 \times 512 \times 3$  to accommodate the number of layers of the OplusVNet network. As shown in Table 2, the frozen layer of the VGG16 transfer learning network contains seven convolutional layers and three maximum pooling layers, and the parameters of the frozen layer are not updated during the training phase of the network. The nonfrozen layer of the VGG16 transfer learning network contains six convolutional layers and two maximum pooling layers, and the parameters of the nonfrozen layer are updated during the training phase of the network. The output data size of the VGG16 transfer learning network is  $16 \times 16 \times 512$ , which we used as the input of our proposed CNN13. The output shapes of each layer of our network in OplusVNet are shown in column 5 of Table 3. First, the six convolutional layers and four maximum pooling layers have an output of  $1 \times 1 \times 256$ . Then, a flatten layer has an output size of 256, followed by two fully connected layers with 512 neurons. The final layer is a fully connected SoftMax layer with five types of probability distributions.

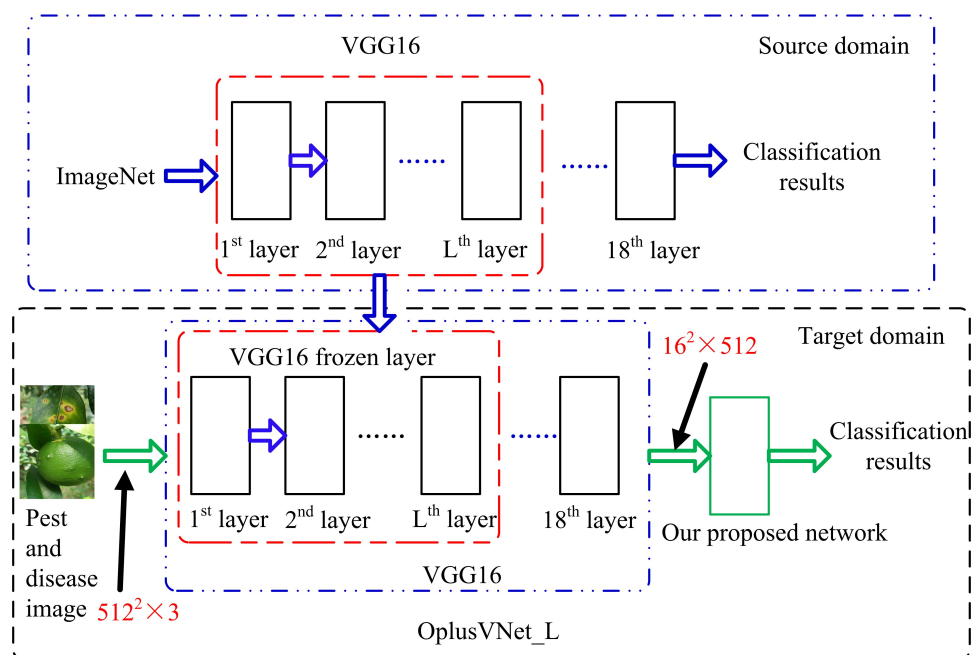


Figure 2. Transfer learning network based on our proposed network.

**Table 2.** The structure and parameters of the VGG16 transfer learning network of OplusVNet\_10 with the input data size of  $512 \times 512 \times 3$ .

|                    | Number of Layers | VGG16               | Output Shape                | Number of Parameters |
|--------------------|------------------|---------------------|-----------------------------|----------------------|
| Frozen layer       | 1                | <i>Conv_1</i>       | $512 \times 512 \times 64$  | 1792                 |
|                    | 2                | <i>Conv_2</i>       | $512 \times 512 \times 64$  | 36,928               |
|                    | 3                | <i>MaxPooling_1</i> | $256 \times 256 \times 64$  | 0                    |
|                    | 4                | <i>Conv_3</i>       | $256 \times 256 \times 128$ | 73,856               |
|                    | 5                | <i>Conv_4</i>       | $256 \times 256 \times 128$ | 147,584              |
|                    | 6                | <i>MaxPooling_2</i> | $128 \times 128 \times 128$ | 0                    |
|                    | 7                | <i>Conv_5</i>       | $128 \times 128 \times 256$ | 295,168              |
|                    | 8                | <i>Conv_6</i>       | $128 \times 128 \times 256$ | 590,080              |
|                    | 9                | <i>Conv_7</i>       | $128 \times 128 \times 256$ | 590,080              |
|                    | 10               | <i>MaxPooling_3</i> | $64 \times 64 \times 256$   | 0                    |
| Non-freezing layer | 11               | <i>Conv_8</i>       | $64 \times 64 \times 512$   | 1,180,160            |
|                    | 12               | <i>Conv_9</i>       | $64 \times 64 \times 512$   | 2,359,808            |
|                    | 13               | <i>Conv_10</i>      | $64 \times 64 \times 512$   | 2,359,808            |
|                    | 14               | <i>MaxPooling_4</i> | $32 \times 32 \times 512$   | 0                    |
|                    | 15               | <i>Conv_11</i>      | $32 \times 32 \times 512$   | 2,359,808            |
|                    | 16               | <i>Conv_12</i>      | $32 \times 32 \times 512$   | 2,359,808            |
|                    | 17               | <i>Conv_13</i>      | $32 \times 32 \times 512$   | 2,359,808            |
|                    | 18               | <i>MaxPooling_5</i> | $16 \times 16 \times 512$   | 0                    |

**Table 3.** The parameters and structure of our proposed network with the input data size of  $16 \times 16 \times 512$ .

| Layers                     | Layer Type              | Core Size/Number | Convolution Step | Output Shape             | Parameters |
|----------------------------|-------------------------|------------------|------------------|--------------------------|------------|
| 1                          | <i>Conv_1</i>           | $3 \times 3/64$  | $1 \times 1$     | $16 \times 16 \times 64$ | 294,976    |
|                            | <i>PReLU_1</i>          |                  |                  |                          | 64         |
| 2                          | <i>Conv_2</i>           | $3 \times 3/64$  | $1 \times 1$     | $8 \times 8 \times 64$   | 36,928     |
|                            | <i>PReLU_2</i>          |                  |                  |                          | 64         |
| 3                          | <i>MaxPooling_1</i>     | $3 \times 3$     | $2 \times 2$     | $8 \times 8 \times 128$  | 0          |
| 4                          | <i>Conv_3</i>           | $3 \times 3/128$ | $1 \times 1$     |                          | 73,856     |
|                            | <i>PReLU_3</i>          |                  |                  | 128                      |            |
| 5                          | <i>Conv_4</i>           | $3 \times 3/128$ | $1 \times 1$     | $4 \times 4 \times 128$  | 147,584    |
|                            | <i>PReLU_5</i>          |                  |                  |                          | 128        |
| 6                          | <i>MaxPooling_2</i>     | $3 \times 3$     | $2 \times 2$     | $4 \times 4 \times 256$  | 0          |
| 7                          | <i>Conv_5</i>           | $3 \times 3/256$ | $1 \times 1$     |                          | 259,168    |
|                            | <i>PReLU_5</i>          |                  |                  | 256                      |            |
| 8                          | <i>MaxPooling_3</i>     | $3 \times 3$     | $2 \times 2$     | $2 \times 2 \times 256$  | 0          |
| 9                          | <i>Conv_6</i>           | $3 \times 3/256$ | $1 \times 1$     |                          | 590,080    |
|                            | <i>PReLU_6</i>          |                  |                  | 256                      |            |
| 10                         | <i>MaxPooling_4</i>     | $3 \times 3$     | $2 \times 2$     | $1 \times 1 \times 256$  | 0          |
| 11                         | <i>Flattern</i>         |                  |                  | 256                      | 0          |
| 12                         | <i>Fullyconnected_1</i> | /512             |                  | 512                      | 131,584    |
| 13                         | <i>PReLU_7</i>          |                  |                  |                          | 256        |
|                            | <i>Fullyconnected_2</i> | /5               |                  | 5                        | 2565       |
| Total number of parameters |                         |                  |                  |                          | 1,538,149  |

### 3.2. Proposed 13-Layer Convolutional Neural Network (CNN13)

One of the popular image classification networks is AlexNet, which was proposed by Krizhevsky et al. [28,29]. It contains five convolutional, three maximum pooling, and three fully connected layers. In this study, to more accurately extract and fit the features of the disease and pest image data, we designed a new CNN13 that includes six convolutional, four maximum pooling, one flatten, and two fully connected layers. Table 2 shows the proposed CNN13 structure with an input data size of  $16 \times 16 \times 512$ .

To obtain more nonlinear transformation features of disease and pest image data and reduce the number of parameters of the network, the kernel size of both convolutional

and pooling layers in the whole network is  $3 \times 3$ . In the pooling layer operation, the overlapping maximum pooling is used to increase the richness of the texture features of the crop pest images. The activation function we used in this study was the PReLU function with parameters, whose equation is defined by:

$$PReLU(x) = \begin{cases} x, & x \geq 0 \\ \alpha x, & x < 0 \end{cases} \quad (1)$$

where  $\alpha$  is updated according to the data change. When  $\alpha$  is updated in the back propagation (BP), a momentum update is used. When the input value is negative, the output value of the function does not simply go to zero, so it retains more useful information and causes less “death” of neurons. We also use the PReLU function as the activation function for each convolutional layer and after the first fully connected layer to add nonlinear elements to improve the representation of the network model. Table 3 shows that the PReLU function only adds a small number of parameters, which has little impact on the computational effort and overfitting of the network.

The last fully connected layer returns the class labels with the SoftMax function to generate the probability distribution, and the class with the highest probability is used as the classification result. The SoftMax function is given by

$$\text{SoftMax}(x_i) = \frac{e^{x_i}}{\sum_{c=1}^C e^{x_c}} \quad (2)$$

where  $x_i$  is the output value of the  $i$ th class, and  $C$  is the number of classes.

When the number of images is small, a network that has more layers and is deeper and more complex is prone to overfitting problems. Therefore, we introduced dropout [30] in the fully connected layer to reduce overfitting. The neurons in the fully connected layer are “inactivated” with a certain probability  $P$ . The inactivated neurons are no longer involved in the forward and backward propagation of the layer. Compared with traditional methods, dropout reduces the size of the network: it is equivalent to allowing multiple dropout networks to learn data characteristics, and each subnet only learns the local characteristics of the data. These subnets eventually share weights, increasing the power of the overall generalization of the network model. When the input network image size is  $16 \times 16$  for 512 channels of data, the overall number of parameters in our network is reduced to 1,538,149.

#### 4. Experimental Results and Analysis

To evaluate the performance of the proposed OplusVNet network for citrus diseases= image classification, we compared its results with those of AlexNet network [28,29] and the VGG16 network (TL-VGG16) [27]. The data, data pre-processing, and data enhancement operations used by all networks were the same. In this study, we set the learning rate, epoch, and optimizer of the OplusVNet network to  $1 \times 10^{-5}$ , 50, and Nadam, respectively. We conducted the experiments using Python programming language and TensorFlow deep learning, with Windows 10 as the operating system with a CPU Intel Core i7-8700 with 6 CPU cores, 32 G RAM, and a GeForce GTX 1080 Ti GPU.

##### 4.1. Evaluation Metrics

In this study, we used the F1 score and accuracy rate as metrics to evaluate the effectiveness of the different network models for citrus disease and pest image classification. The F1 score is defined as

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{(\text{Precision} + \text{Recall})}$$

where Precision is the ratio between the number of correctly identified disease images and the number of correctly predicted disease images; Recall is the ratio between the number of correctly identified disease images and the number of all correct disease images in that category.

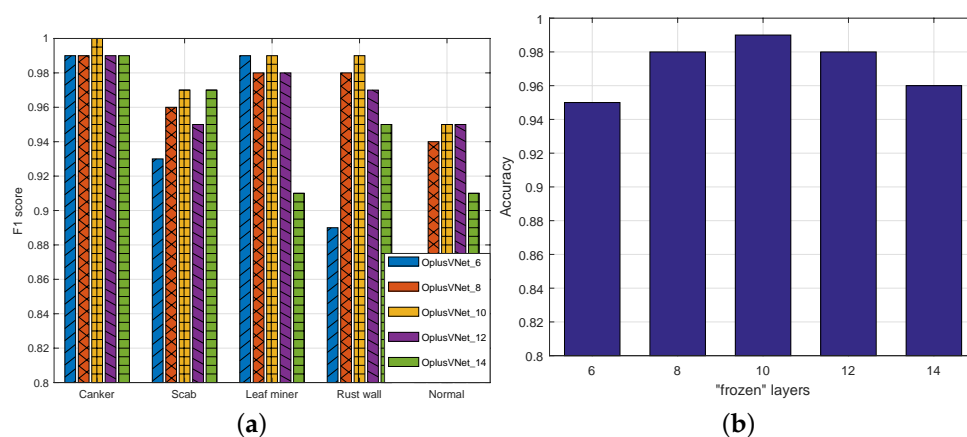
Accuracy is defined as:

$$\text{Accuracy} = \frac{\text{Identify the correct total number of disease and pest images}}{\text{Total number of disease and pest images}}$$

#### 4.2. Experiments with OplusVNet with Different Frozen Mechanisms

Network-based transfer learning is part of the OplusVNet network, and the extracted feature images are used as the input of the subsequent layers. The parameters of the subsequent layer are trained with the target domain data, which plays an important role in the subsequent layer, enabling the network to more accurately fit the target domain data and thus further improve the model prediction results.

We designed different frozen mechanisms for OplusVNet, where OplusVNet\_L indicates that the first L layers are frozen. The experimental results for each network model are shown in Figure 3. Figure 3a shows the F1 score of the OplusVNet network with different frozen mechanisms for different disease and pest categories. Figure 3a shows that the OplusVNet\_10 network achieved more accurate results on different citrus pest categories. The accuracy rates of OplusVNet networks with different frozen mechanisms are shown in Figure 3b. Figure 3b shows that OplusVNet\_10 was the most accurate. From the above results and analysis, we found that (1) the fewer the number of frozen layers, the more layers can be trained, which creates the risk of network overfitting. Conversely, the larger the number of frozen layers, the smaller the role played by the target domain data in the network, so the network may not be able to accurately fit the target domain data. (2) When the number of frozen layers of the transformed learning network was 10, our proposed OplusVNet network achieved an F1 score of more than 95% for individual pest category identification, and the overall classification accuracy was 95%, showing that the risk of network overfitting was effectively reduced and the target data were better learned in the network. The training time per batch size of the proposed OplusVNet network was 132 ms.



**Figure 3.** Performance analysis of different transfer learning network models. (a) F1 score of different OplusVNet models on various citrus pests and diseases. (b) The accuracy of different OplusVNet models on the validation set.

#### 4.3. Comparison with State-of-the-Art Networks

To further validate the performance of the OplusVNet network, we compared it with AlexNet [28,29], TL-VGG16 [27], and RepVGG [31]. All the convolutional layers of the VGG16 network are frozen. These frozen layers retain the parameter weights obtained by the VGG16 network training in ImageNet, and only the fully connected layer and output

unit of the network are trained. The classification results of the different networks on the test set are shown in Table 4. The TL-VGG16 network shown in Table 4 is a transfer-learning-based VGG16 network. For canker disease, the highest F1 score was obtained by the proposed OplusVNet\_10 (1.00), followed by RepVGG (0.99). For scab disease, the highest F1 score was obtained by the proposed OplusVNet\_10 and RepVGG (0.97), followed by TL-VGG16 (0.91). For leaf miner, the highest F1 score was obtained by the proposed OplusVNet\_10 (0.99), followed by RepVGG (0.96). For rust wall, the highest F1 score was obtained by the proposed OplusVNet\_10 (0.99), followed by RepVGG (0.94). For normal leaves, the highest F1 score was obtained by the proposed OplusVNet\_10 (0.95), followed by RepVGG (0.93). For most experiments, the proposed OplusVNet\_10 obtained the highest F1 score, followed by RepVGG. For leaf miner and normal leaves, TL-VGG16 performed substantially worse than the other methods. The overall classification accuracy of the TL-VGG16 network model was lower than that of the AlexNet network because the features obtained by the TL-VGG16 network feature extractor by learning the source domain data were not sufficiently learned. For the proposed OplusVNet\_10, RepVGG, and AlexNet, the larger the number of images for a specific type, the more accurate the performance. The accuracy rate of the proposed OplusVNet\_10 was 0.99, which was greater than the values 0.93, 0.88, and 0.97 of AlexNet, TL-VGG16, and RepVGG, respectively.

In summary, OplusVNet\_10 outperformed the other networks in terms of both the F1 score for individual classes of disease and pest and overall classification accuracy, especially when the number of image classes was relatively low. The proposed network, combined with a network-based transfer learning network, can effectively fit data features and overcome the problems caused by a small and uneven data volume.

**Table 4.** F1 score and accuracy rate values of different methods.

| Methods      | F1 Score    |             |             |             |             | Accuracy    |
|--------------|-------------|-------------|-------------|-------------|-------------|-------------|
|              | Canker      | Scab        | Leaf Miner  | Rust Wall   | Normal      |             |
| AlexNet      | 0.98        | 0.88        | 0.93        | 0.87        | 0.83        | 0.93        |
| TL-VGG16     | 0.94        | 0.91        | 0.74        | 0.85        | 0.68        | 0.88        |
| RepVGG       | 0.99        | <b>0.97</b> | 0.96        | 0.94        | 0.93        | 0.97        |
| OplusVNet_10 | <b>1.00</b> | <b>0.97</b> | <b>0.99</b> | <b>0.99</b> | <b>0.95</b> | <b>0.99</b> |

The bold values mean the best results.

#### 4.4. OplusVNet Network Performance Analysis

From the above analyses, we found that for most classification methods, the results largely depended on the number of images used for training. However, for some diseases, obtaining enough images for the classification task may be challenging. To analyze the performance of our proposed OplusVNet network on sets with different numbers of images, we set the number of images in the training set to 170, 120, and 70, separately. The number of images in the test set was 30 for each category. First, we randomly selected 200 images from the original dataset of each class, and then randomly selected 30 images from these 200 images as the test set; we used the remaining 170 images as the first training set. Then, we randomly selected 120 images from the 170 images in the first training set for the second training set. We randomly selected 70 images from the 170 images in the first training set for the third training set. Due to the small number of images in the training set, we set the batch size of the OplusVNet network to 32 in this part of the study.

##### 4.4.1. Experimental Results of Proposed OplusVNet on Small Datasets

Table 5 shows the results of OplusVNet with different frozen layers for different numbers of images in the training set: 170, 120, and 70. Table 5 shows that when the number of images in the training set was 170, the highest recognition accuracy was achieved with four, six, eight, and ten frozen layers. When the number of images in the training set was 120, the highest recognition accuracy was achieved with six and twelve frozen layers. When the number of images in the training set was 70, the highest recognition



accuracy was achieved with six and eight frozen layers. When the number of images in the training set was less than 170, the network with six frozen layers had higher generalization performance. Therefore, the OplusVNet network with six frozen layers should be used when the number of images in the training set is small.

**Table 5.** The accuracy of the proposed OplusVNet with different frozen layer networks on different number of image training sets.

| Number | Number of Frozen Layers |             |             |             |             |      |      |
|--------|-------------------------|-------------|-------------|-------------|-------------|------|------|
|        | 4                       | 6           | 8           | 10          | 12          | 14   | 16   |
| 170    | <b>0.97</b>             | <b>0.97</b> | <b>0.97</b> | <b>0.97</b> | 0.96        | 0.94 | 0.91 |
| 120    | 0.95                    | <b>0.96</b> | 0.94        | 0.95        | <b>0.96</b> | 0.91 | 0.89 |
| 70     | 0.93                    | <b>0.94</b> | <b>0.94</b> | 0.92        | 0.93        | 0.89 | 0.89 |

The bold values mean the best results.

#### 4.4.2. Experimental Results of Different Network Models on Small Datasets

Table 6 shows the experimental results of the OplusVNet\_6, AlexNet [28,29], and TL-VGG16 [27] networks on different small training sets. Table 6 shows that OplusVNet\_6 outperformed the other networks in terms of both the F1 score for individual classes of disease and pest and overall classification accuracy. Our proposed OplusVNet network effectively avoids the risk of overfitting in small datasets and can more effectively learn and fit disease and pest image data with texture features.

**Table 6.** Experimental results of different network models in different small training sets.

| Number | Network      | F1 Score    |             |             |             |             | Accuracy    |
|--------|--------------|-------------|-------------|-------------|-------------|-------------|-------------|
|        |              | Canker      | Scab        | Leaf Miner  | Rust Wall   | Normal      |             |
| 170    | AlexNet      | 0.87        | 0.75        | 0.85        | 0.91        | 0.74        | 0.83        |
|        | TL-VGG16     | 0.92        | 0.90        | 0.87        | 0.85        | 0.83        | 0.87        |
|        | OplusVNet_10 | <b>0.98</b> | <b>0.97</b> | <b>0.97</b> | <b>0.97</b> | <b>0.98</b> | <b>0.97</b> |
|        | OplusVNet_6  | <b>0.98</b> | <b>0.97</b> | <b>0.97</b> | <b>0.97</b> | <b>0.98</b> | <b>0.97</b> |
| 120    | AlexNet      | 0.81        | 0.75        | 0.85        | 0.87        | 0.70        | 0.80        |
|        | TL-VGG16     | 0.90        | 0.90        | 0.84        | 0.88        | 0.75        | 0.85        |
|        | OplusVNet_10 | 0.93        | 0.94        | 0.97        | 0.96        | 0.96        | 0.95        |
|        | OplusVNet_6  | <b>0.94</b> | <b>0.95</b> | <b>0.98</b> | <b>0.97</b> | <b>0.97</b> | <b>0.96</b> |
| 70     | AlexNet      | 0.81        | 0.63        | 0.84        | 0.84        | 0.62        | 0.75        |
|        | TL-VGG16     | 0.86        | 0.82        | 0.79        | 0.79        | 0.64        | 0.80        |
|        | OplusVNet_10 | 0.93        | 0.90        | 0.94        | 0.90        | 0.93        | 0.92        |
|        | OplusVNet_6  | <b>0.95</b> | <b>0.92</b> | <b>0.97</b> | <b>0.91</b> | <b>0.95</b> | <b>0.94</b> |

The bold values mean the best results.

In summary, the performance of all methods generally decreased with the decrease in the number of images. For scab disease images, AlexNet performed substantially worse than the others. For normal citrus images, AlexNet and TL-VGG16 received low scores. For all the experiments, the proposed OplusVNet obtained the best F1 and accuracy scores. This demonstrated that the proposed OplusVNet is robust and effective for the identification of different disease and pest types and small datasets.

## 5. Conclusions

To improve the training efficiency and prevent overfitting, we designed a new CNN13 to more accurately extract and fit the image features of the data. To address the problem caused by small amounts of image data and uneven image data of various diseases, we constructed OplusVNet by combining the proposed CNN13 with network-based transfer learning. Compared with the general image classification networks (e.g., AlexNet and TL-VGG16), the proposed OplusVNet obtained considerably higher F1 and accuracy scores

on small and unbalanced datasets. In contrast to RepVGG, the accuracy of the proposed OplusVNet was higher. The experimental results showed that the proposed OplusVNet performs better than the state of the art feature image classification networks. The F1 score of the proposed OplusVNet network for individual disease categories was above 0.95, and the overall accuracy on the test set was 0.99. When the amount of image data in the training set was small, the frozen six-layer OplusVNet showed high generalization performance and effectively reduced the risk of overfitting. However, the proposed OplusVNet network cannot be directly used on a mobile application. In future work, we will develop a lightweight OplusVNet network and design a smartphone APP assistant for fruit farmers to identify citrus diseases and insect pests to promote agricultural automation and intelligent agricultural systems.

**Author Contributions:** Conceptualization, C.Y. and R.C.; methodology, C.Y.; software, Z.T. and C.D.; validation, C.Y., Z.T. and C.D.; resources, J.W.; data curation, J.W.; writing—original draft preparation, C.Y. and C.D.; writing—review and editing, Z.T., R.C., J.W. and Y.L.; supervision, C.Y.; project administration, R.C.; funding acquisition, R.C.; investigation, Y.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported in part by the National Natural Science Foundation of China under Grant 61802064, in part by the Fujian Science and Technology Planning Project under Grant 2021S0007 and 2022H6006, in part by the Natural Science Fund of Ningxia Province under Grant 2020AAC03307.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Citrus disease dataset: [https://drive.google.com/file/d/1w6\\_TpebZJ4hvUnlAgGWMBiFcDINTxk5\\_/view?usp=sharing](https://drive.google.com/file/d/1w6_TpebZJ4hvUnlAgGWMBiFcDINTxk5_/view?usp=sharing) (accessed on 18 July 2022).

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

## References

1. Khan, M.A.; Akram, T.; Sharif, M.; Awais, M.; Javed, K.; Ali, H.; Saba, T. CCDF: Automatic system for segmentation and recognition of fruit crops diseases based on correlation coefficient and deep CNN features. *Comput. Electron. Agric.* **2018**, *155*, 220–236. [CrossRef]
2. Moriya, E.A.S.; Imai, N.N.; Tommaselli, A.M.G.; Berveglieri, A.; Santos, G.H.; Soares, M.A.; Marino, M.; Reis, T.T. Detection and mapping of trees infected with citrus gummosis using UAV hyperspectral data. *Comput. Electron. Agric.* **2021**, *188*, 106298. [CrossRef]
3. Ngugi, L.C.; Abelwahab, M.; Abo-Zahhad, M. Recent advances in image processing techniques for automated leaf pest and disease recognition: A review. *Inf. Process. Agric.* **2021**, *8*, 27–51. [CrossRef]
4. Abdu, A.M.; Mokji, M.M.; Sheikh, U.U. Automatic vegetable disease identification approach using individual lesion features. *Comput. Electron. Agric.* **2020**, *176*, 105660. [CrossRef]
5. Nanekaran, Y.A.; Zhang, D.; Chen, J.; Tian, Y.; Al-Nabhan, N. Recognition of plant leaf diseases based on computer vision. *J. Ambient. Intell. Humaniz. Comput.* **2020**, 1–18. [CrossRef]
6. Jagan, K.; Balasubramanian, M.; Palanivel, S. Detection and Recognition of Diseases from Paddy Plant Leaf Images. *Int. J. Comput. Appl.* **2016**, *144*, 34–41. [CrossRef]
7. Hsiao, J.K.; Kang, L.W.; Chang, C.L.; Hsu, C.Y.; Chen, C.Y. Learning sparse representation for leaf image recognition. In Proceedings of the 2014 IEEE International Conference on Consumer Electronics, Taipei, Taiwan, 26–28 May 2014; pp. 209–210. [CrossRef]
8. Zhang, S.; Wu, X.; You, Z.; Zhang, L. Leaf image based cucumber disease recognition using sparse representation classification. *Comput. Electron. Agric.* **2017**, *134*, 135–141. [CrossRef]
9. Liu, J.; Wang, X. Plant diseases and pests detection based on deep learning: A review. *Plant Methods* **2021**, *17*, 22. [CrossRef]
10. Chen, J.; Zhang, D.; Nanekaran, Y. Identifying plant diseases using deep transfer learning and enhanced lightweight network. *Multimed. Tools Appl.* **2020**, *79*, 31497–31515. [CrossRef]
11. Chen, J.; Yin, H.; Zhang, D. A self-adaptive classification method for plant disease detection using GMDH-Logistic model. *Sustain. Comput. Inform. Syst.* **2020**, *28*, 100415. [CrossRef]
12. Chen, J.; Chen, J.; Zhang, D.; Sun, Y.; Nanekaran, Y. Using deep transfer learning for image-based plant disease identification. *Comput. Electron. Agric.* **2020**, *173*, 105393. [CrossRef]

13. Chen, J.; Zhang, D.; Nanehkaran, Y.; Li, D. Detection of rice plant diseases based on deep transfer learning. *J. Sci. Food Agric.* **2020**, *100*. [[CrossRef](#)]
14. Haque, M.; Marwaha, S.; Deb, C.; Nigam, S.; Arora, A.; Hooda, K.; Soujanya, P.L.; Aggarwal, S.; Lall, B.; Kumar, M.; et al. Deep learning-based approach for identification of diseases of maize crop. *Sci. Rep.* **2022**, *12*, 6334. [[CrossRef](#)] [[PubMed](#)]
15. Kundu, N.; Rani, G.; Dhaka, V.S.; Gupta, K.; Nayak, S.C.; Verma, S.; Ijaz, M.F.; Woniak, M. IoT and Interpretable Machine Learning Based Framework for Disease Prediction in Pearl Millet. *Sensors* **2021**, *21*, 5386. [[CrossRef](#)] [[PubMed](#)]
16. Rehman, M.Z.U.; Ahmed, F.; Khan, M.A.; Tariq, U.; Jamal, S.S.; Ahmad, J.; Hussain, I. Classification of Citrus Plant Diseases Using Deep Transfer Learning. *Comput. Mater. Contin.* **2022**, *70*, 1401–1417. [[CrossRef](#)]
17. Nazari, K.; Ebadi, M.; Berahmand, K. Diagnosis of Alternaria disease and Leafminer pest on Tomato Leaves using Image Processing Techniques. *J. Sci. Food Agric.* **2022**. [[CrossRef](#)] [[PubMed](#)]
18. Dhiman, P.; Kukreja, V.; Manoharan, P.; Kaur, A.; Kamruzzaman, M.M.; Dhaou, I.B.; Iwendi, C. A Novel Deep Learning Model for Detection of Severity Level of the Disease in Citrus Fruits. *Electronics* **2022**, *11*, 495. [[CrossRef](#)]
19. Mohanty, S.P.; Hughes, D.P.; Salath, M. Using Deep Learning for Image-Based Plant Disease Detection. *Front. Plant Sci.* **2016**, *7*, 1419. [[CrossRef](#)]
20. Sladojevic, S.; Arsenovic, M.; Anderla, A.; Culibrk, D.; Stefanovic, D. Deep Neural Networks Based Recognition of Plant Diseases by Leaf Image Classification. *Comput. Intell. Neurosci.* **2016**, *2016*, 3289801. [[CrossRef](#)]
21. Tan, C.; Sun, F.; Kong, T.; Zhang, W.; Yang, C.; Liu, C. A Survey on Deep Transfer Learning. In Proceedings of the 27th International Conference on Artificial Neural Networks, Rhodes, Greece, 4–7 October 2018; pp. 270–279.
22. Thenmozhi, K.; Reddy, U.S. Crop pest classification based on deep convolutional neural network and transfer learning. *Comput. Electron. Agric.* **2019**, *164*, 104906. [[CrossRef](#)]
23. Selvaraj, M.G.; Vergara, A.; Ruiz, H.; Safari, N.; Elayabalan, S.; Ocimati, W.; Blomme, G. AI-powered banana diseases and pest detection. *Plant Methods* **2019**, *15*, 92. [[CrossRef](#)]
24. Coulibaly, S.; Kamsu-Foguem, B.; Kamissoko, D.; Traore, D. Deep neural networks with transfer learning in millet crop images. *Comput. Ind.* **2019**, *108*, 115–120. [[CrossRef](#)]
25. Barman, U.; Choudhury, R.D.; Sahu, D.; Barman, G.G. Comparison of convolution neural networks for smartphone image based real time classification of citrus leaf disease. *Comput. Electron. Agric.* **2020**, *177*, 105661. [[CrossRef](#)]
26. Morteza, K.; Ezzatollah, A.A.A.; Ehsan, K. Citrus pests classification using an ensemble of deep learning models. *Comput. Electron. Agric.* **2021**, *186*, 106192.
27. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arXiv:1409.1556.
28. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. In Proceedings of the Annual Conference on Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–8 December 2012; Volume 25.
29. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. *Commun. ACM* **2017**, *60*, 84–90. [[CrossRef](#)]
30. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
31. 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, Nashville, TN, USA, 20–25 June 2021; pp. 13733–13742.