

Article

The First Study of White Rust Disease Recognition by Using Deep Neural Networks and Raspberry Pi Module Application in Chrysanthemum

Toan Khac Nguyen ¹, L. Minh Dang ², Truong-Dong Do ³  and Jin Hee Lim ^{1,*}¹ Department of Plant Biotechnology, Sejong University, Seoul 05006, Republic of Korea; toanchrys@sju.ac.kr² Department of Information and Communication Engineering, Convergence Engineering for Intelligent Drone, Sejong University, Seoul 05006, Republic of Korea; minhdl@sejong.ac.kr³ Department of Aerospace System Engineering, Convergence Engineering for Intelligent Drone, Sejong University, Seoul 05006, Republic of Korea; dongdo@sju.ac.kr

* Correspondence: jinheelim@sejong.ac.kr

Abstract: Growth factors affect farm owners, environmental conditions, nutrient adaptation, and resistance to chrysanthemum diseases. Healthy chrysanthemum plants can overcome all these factors and provide farms owners with a lot of income. Chrysanthemum white rust disease is a common disease that occurs worldwide; if not treated promptly, the disease spreads to the entire leaf surface, causing the plant's leaves to burn, turn yellow, and fall prematurely, reducing the photosynthetic performance of the plant and the appearance of the flower branches. In Korea, chrysanthemum white rust disease most often occurs during the spring and autumn seasons, when temperature varies during the summer monsoon, and when ventilation is poor in the winter. Deep neural networks were used to determine healthy and unhealthy plants. We applied the Raspberry Pi 3 module to recognize white rust and test four neural network models. The five main deep neural network processes utilized for a dataset of non-diseased and white rust leaves include: (1) data collection; (2) data partitioning; (3) feature extraction; (4) feature engineering; and (5) prediction modeling based on the train–test loss of 35 epochs within 20 min using Linux. White rust recognition is performed for comparison using four models, namely, DenseNet121, ResNet50, VGG-19, and MobileNet v2. The qualitative white rust detection system is achieved using a Raspberry Pi 3 module. All models accomplished an accuracy of over 94%, and MobileNet v2 achieved the highest accuracy, precision, and recall at over 98%. In the precision comparison, DenseNet121 obtained the second highest recognition accuracy of 97%, whereas ResNet50 and VGG-19 achieved slightly lower accuracies at 95% and 94%, respectively. Qualitative results were obtained using the Raspberry Pi 3 module to assess the performance of the four models. All models had accuracies of over 91%, with ResNet50 obtaining a value of 91%, VGG19 reaching a value of 93%, and DenseNet121 reaching 95%. The highest accuracy rate was 97% (MobileNet v2). MobileNet v2 was validated as the most effective model to recognize white rust in chrysanthemums using the Raspberry Pi 3 system. Raspberry Pi 3 module was considered, in conjunction with the MobileNet v2 model, to be the best application system. MobileNet v2 and Raspberry Pi require a low cost for the recognition of chrysanthemum white rust and the diagnosis of chrysanthemum plant health conditions, reducing the risk of white rust disease and minimizing costs and efforts while improving floral production. Chrysanthemum farmers should consider applying the Raspberry Pi module for detecting white rust, protecting healthy plant growth, and increasing yields with low-cost.



Citation: Nguyen, T.K.; Dang, L.M.; Do, T.-D.; Lim, J.H. The First Study of White Rust Disease Recognition by Using Deep Neural Networks and Raspberry Pi Module Application in Chrysanthemum. *Inventions* **2023**, *8*, 76. <https://doi.org/10.3390/inventions8030076>

Academic Editor: Konstantinos G. Arvanitis

Received: 2 May 2023

Revised: 24 May 2023

Accepted: 26 May 2023

Published: 31 May 2023



Copyright: © 2023 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: clustering; disease detection; image processing; plant health condition; Raspberry Pi; white rust chrysanthemum

1. Introduction

Chrysanthemum (*Chrysanthemum* sp.) is an important species of Asteraceae and is a high-profit floricultural crop, ranked second in the global florist market business [1,2]. Chrysanthemums are identified by type, shapes of flower, and their various flower colors. Additionally, medicinal secondary compounds have a high value in floral species [1,2]. In spite of global climate change and burgeoning human populations, smart-farm challenges should be overcome to increase agricultural product yield [3]. The need to develop smart farms in the future requires knowledge of how to apply science and technology to increase agricultural output, especially of chrysanthemums, as their yield and quality of flowers are in the top-priority group [1–3]. At the same time, smart farms must limit harmful pests [1–3]. Smart-farming software could be upgraded to produce large levels of production and increase profit from chrysanthemums to decrease the risk of farming control [3]. Therefore, the application approach should be utilized to produce the strongest economic profit from chrysanthemums that should be obtained with disease and insect resistance, which are important characteristics for chrysanthemum breeding and are introduced in various colors of petals, shapes, and abundant types of flowers [2,3].

Chrysanthemums may be acutely damaged by a disease known as white rust. Chrysanthemum white rust (*Puccinia horiana* P. Henn.) is not an injurious disease that can spread quickly in greenhouse and pandemic environments, but it does cause severe crop losses that can be bad news for farm owners [4]. The symptoms of chrysanthemum white rust are obvious; they can be distinguished by small white (or light green) or yellow spots (~4 mm wide) on the upper surface of the leaf [5]. Heavy infestations can stunt chrysanthemum plant development and reduce vigor, eventually causing death [5]. This disease is most often found from late summer–autumn or in winter (in greenhouse); however, it is generally active all year round [5]. Chrysanthemum cultivars have a problem in that they are susceptible to white rust [4,5].

Precise disease detection can act as a developed technique and an upgraded application to protect and apply farming prevention and treatment process [6]. Deep learning has now been widely used in computer networks, item recognition, speech recognition, natural processing language, and recommendation systems [7]. Deep neural networks have currently been profitably used in various diverse domains as patterns of learning modules [8]. Neural networks suggest a mapping approach between input data, such as a picture of an unhealthy plant (or part of the diseased plant), to release output data, such as a crop-matching disease [8]. The matching nodes of neural networks are linked to mathematical performances that install concerning algorithmic inputs from the entering edges and produce a numerical output as a releasing edge [9]. In simple terms, deep neural networks use automatic mapping from the input-layer data to the output-layer data over a series of stacked layers of evolving nodes [8–10]. The challenging approach involves creating a deep network in such a way that the structure of the network, as well as their functions and edge weights, accurately maps data from the input to the output [11]. The training step involves improving the mapping data by tuning the network parameters in deep neural networks [9]. These processes require automatically challenging computation and have been proven to be effective by numerous conceptual and engineering breakthroughs [10,12].

Automatic classification models in plant diseases have been constructed by several machine learning approaches and have been widely applied to vegetable crops [3,7,13]. However, few models have been applied to flower crops [7,8,13]. Based on this requirement, deep learning approaches have resulted in the emergence of high-configuration systems [13,14]. There are two critical applications that use deep learning algorithms for automatic processing detection in agriculture-controlling systems: classification and disease detection [8,15]. Plant disease detection and plant classification are two significant applications where deep learning algorithms are widely used to automate processes in agriculture [14,15]. Convolutional neural networks (CNNs) are a suitable selection for image classification in deep learning [10,16,17]. CNNs can automatically extract features

and overcome the complications involved in deciding the relevant features from images in manual engineering feature steps [16,17].

MobileNet is a CNN architecture network that is observed well on mobile devices [18]. The model network is constructed as open-sourced software by Google [18,19]. To date, there are three stable versions, namely MobileNet v1 [19], MobileNet v2 [20], and MobileNet v3 [21]. MobileNet architecture is an especially open-sourced network because it uses much less computing power to run [19–21]. This makes the network a perfect fit for devices of mobile applications as it has fastened systems and is able to run without GPUs [19–21]. MobileNet v1 is presented as the first version of MobileNet models that has more complicated convolution layers and matrix parameters when compared to MobileNet v2 [19,20]. MobileNet v2 is the next version of MobileNet models, which significantly reduces a number of parameters to be a lower matrix in the deep neural network [19,20]. MobileNet v3 is faster and more precise than MobileNet v2, but it has only top-1 accuracy, while top-5 accuracy is not indicated at all [21]. These versions can make the results more lightweight in deep neural networks. Lightweight results are best suited for embedded systems and mobile devices using pre-trained models. IT users do not need to build or train a neural network from scratch; therefore, it can save time for development models. There are several models for image classification and computer vision linked to pre-trained networks, such as AlexNet, Inception v3, LeNet, DenseNet-121, MobileNet, ResNet-50, and VGG-19. In this study, we used MobileNet v2 for the identification of white rust in chrysanthemums.

Smartphones are a common device used to identify plant diseases because they include high-level smart CPU attachments and high display resolution, and they can install useful accessories, such as LED microscopes [22]. The interface factors of broad smartphone perforations, such as HD cameras and high processors installed into mobile devices, force an appropriate solution in which the disease is identified based on automatic image recognition [22].

Raspberry Pi, which is a low-cost mini computer, is as small as a credit card and can be plugged into a monitor, keyboard, or mouse. It is an accomplished miniature construction that allows people of all genders and ages to entertain and explore computing works and to determine how to run processes by writing codes in Scratch and Python. It can perform everything that a desktop computer can, including internet communication and video playing; making spreadsheets, processes, and writing words; and playing games. Third-generation Raspberry Pi is representative of the earliest model called Raspberry Pi 3 Model B. Raspberry Pi 3 Model B was replaced by Raspberry Pi 2 Model B in February 2016, which offers a wider range of uses than Pi 2. It is equipped with the standard HDMI and USB ports, contains 1 GB of RAM, can connect to Wi-Fi and Bluetooth, and can install Ethernet functions. This model is characterized by low heat and power and has been authorized by the following European standards: Electromagnetic Compatibility Directive (EMC) 2014/30/EU and the Restriction of Hazardous Substances (RoHS) Directive 2011/65/EU. In previous papers, convolutional neural network models have been used for disease classification in tomato leaves [23]. Raspberry Pi was established with a graphical user interface [23]. However, Raspberry Pi has not yet been used in studies on chrysanthemum diseases.

In recent years, agricultural researchers have been concerned with smart applications for every method of machine learning and deep learning to be used for image-based plant disease detection [24–26], crop pests recognition [27–29], leaves identification [30], leaf disease detection [31–34], plant disease classification [35], and so on.

We conducted a study on the identification of white rust on chrysanthemums with the following objectives: (1) to aid in the early detection of the disease and to prevent the disease from spreading to healthy chrysanthemums; and (2) to build a model system to accurately identify rust disease on chrysanthemums and apply it to other diseases related to chrysanthemum cultivation accordingly.

The study aimed to develop a cost-effective method for detecting chrysanthemum white rust disease using deep neural networks and the Raspberry Pi 3 module. This objective is significant for floral agriculture, particularly in chrysanthemum farming and breeding. The disease causes severe damage to chrysanthemum plants, which affects leaf health, photosynthesis, and flower quality. Early detection is crucial to prevent its spread and minimize negative impacts. The study applied deep neural networks, including DenseNet121, ResNet50, VGG-19, and MobileNet v2, to identify and differentiate healthy areas and white rust disease in chrysanthemum plants. We established the appropriate tools using a deep learning method to utilize 3264 images of chrysanthemum species with white rust disease and non-disease chrysanthemums to detect chrysanthemum disease. Furthermore, we optimized Raspberry Pi 3 to perform and process real information regarding disease conditions to obtain results that assist in the early recognition of anomalies in chrysanthemum crops. The use of the Raspberry Pi 3 module made the detection system practical and accessible. The potential benefits of the developed tool include early disease detection, reduced costs, improved plant health, increased yields, and easy implementation. By leveraging deep neural networks and Raspberry Pi 3, chrysanthemum growers can effectively manage and mitigate white rust disease, resulting in healthier plants and improved agricultural outcomes.

2. Methodology

Figure 1 illustrates the five main processes utilized in a dataset of non-diseased and white rust leaves, which include (1) data collection; (2) data partitioning; (3) feature extraction; (4) feature engineering; and (5) prediction modeling. A train–test loss of 35 epochs within 20 min under Linux environment was achieved using CPU i7-11800H, GPU NVIDIA GeForce RTX 3060 with 32 GB RAM.

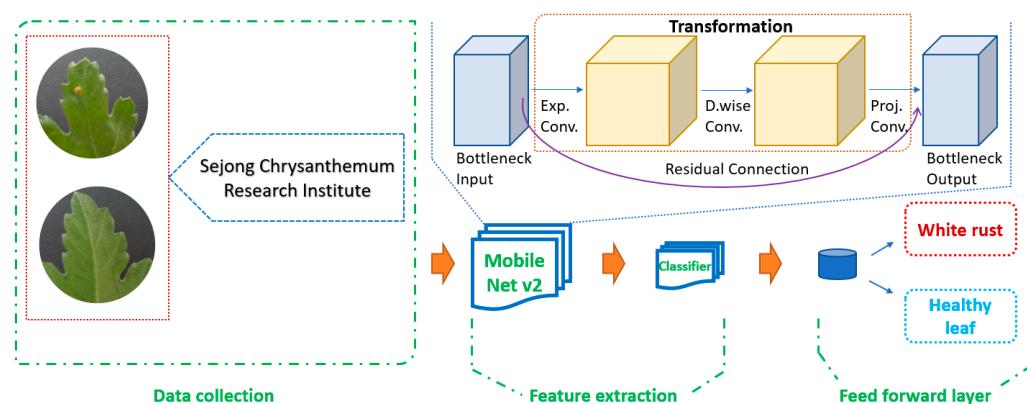


Figure 1. Overview architecture of the framework. MobileNet v2 where Exp. Conv., D.wise Conv., and Proj. Conv. represent expansion convolution, depth-wise convolution, and projection convolution, respectively.

MobileNet v2 introduces two architectural features, namely, linear bottlenecks between the layers and residual connection between the bottlenecks. The first layer is a 1×1 convolution with ReLU6 (expansion convolution), the second layer is the depth-wise convolution, and the third layer is another 1×1 convolution but without any non-linearity. Finally, traditional residual connections contain shortcuts that yield faster training and better accuracy.

Figure 2 describes the Raspberry Pi 3 module for chrysanthemum white rust disease detection using power supply (DC 5 V, 2.1 A), monitor display (JOOYONTECH-JT17JTFT), and Pi camera (V2-913-2664).

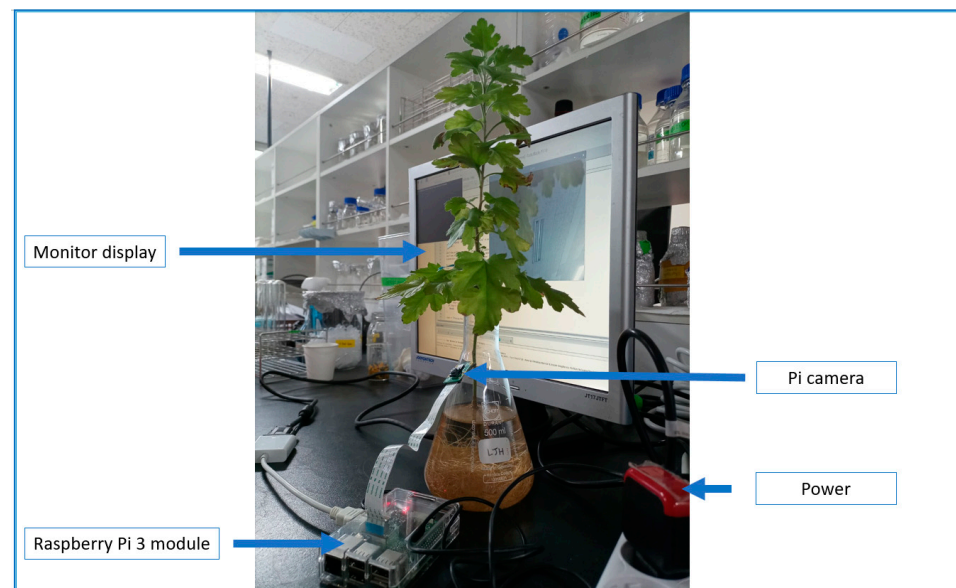


Figure 2. Raspberry Pi 3 with Pi camera installed in a module for chrysanthemum white rust detection. The power supplies for Raspberry Pi 3 module and the power supplies for the monitor are two independent power sources.

2.1. Plant Material and Data Collection

The cultivated chrysanthemum plant ‘Holiday Dream’ was planted at green houses at the Chrysanthemum Research Institute of Sejong University, Korea, with the following growth conditions: a 16 h photoperiod at $25 \pm 2^\circ\text{C}$ with 8 h of darkness at $20 \pm 2^\circ\text{C}$, including a relative humidity (RH) close to 60% for healthy plants. After 4 weeks planting, the plants ‘Holiday Dream’ were divided into 2 groups: non-disease with the same growth condition previously; and the contracted white rust disease (*P. horiana* P. Henn.) under the diseased-growth condition with an RH close to 100%, followed by a 16 h photoperiod at $25 \pm 2^\circ\text{C}$ with 8 h of darkness at $20 \pm 2^\circ\text{C}$. Plants were watered three times per week by using water-drop-pipeline, and we added nutrients two times per week. At the plant leaf age of 6 and 8 weeks, the white rust disease was found in the leaves near ground soil at about 20 cm. All plants (non-disease and white rust disease) were picked up directly and moved to the laboratory for image data collection. The collected dataset was achieved at the chrysanthemum research laboratory using an LG Q52 smartphone with a main quad camera with the following specifications: 48 MP, $f/1.8$, (wide), $1/2.0''$, $0.8\mu\text{m}$, PDAF; 5 MP, $f/2.2$, 115° (ultrawide), $1/5.0''$, $1.12\mu\text{m}$; 2 MP, $f/2.4$, (macro); and 2 MP, $f/2.4$, (depth). The illumination of the laboratory is 641 lx. In total, 3264 images were captured with the same shooting conditions at 9 am. The dataset of 3264 images captured under consistent shooting conditions at 9 am was divided into three sets for training, testing, and validation. The training set consisted of 2000 images, the testing set included 800 images, and the remaining 464 images were allocated to the validation set.

2.2. Data Augmentation

In the deep learning method, the tools used to creating more data include rotation, translation, flipping, and various corresponding changes, which are called data augmentation [36,37]. The augmentation tool is applied offline in this study because our collected data were small.

2.3. Data Partitioning

In total, 3264 images for non-disease and white rust disease samples presented on the abaxial surface of the leaves were collected. The train-valid-test dataset is reported in Table 1 as a detailed description.

Table 1. Train–valid–test dataset.

Class	Training Set	Validation Set	Testing Set	Total
Non-disease	1083	326	259	1668
White rust	1042	318	236	1596

The original dataset included the selection of 1083 images (non-disease class) and 1042 images (white rust class) for a training set (80%), as well as the selection of 326 images (non-disease class) and 318 images (white rust class) for a validation set (20%). An additional 259 images (non-disease class) and 236 images (white rust class) were obtained to serve the purpose of the testing evaluation of the CNN model.

2.4. Model Description

In this section, leaf disease detection was tested using a deep learning algorithm. Images were used for training via MobileNet, teaching the model about the new classes that we want to recognize. MobileNet is known as an efficient convolutional neural network (CNN).

Table 2 provides a specific definition equivalent to the structure of Figure 1, one by one. The table accommodates a variety of mainstream networks with different characteristics using the residual layer with a stride of 1 and the downsizing layer with a stride of 2, alongside the rectified linear unit (ReLU) component of the literature. The parameter structure was designed with two branches as a residual branch and a downsizing branch, which is encompassed by three sub-layers of each branch. When the network consorts deeply, it can be communicated to low-level information so that the network does not disappear. ReLU6 is the first layer of the 1×1 convolution, and depth-wise convolution is the second layer in the model, which is assumed from the literature. Each block links all of the layers to obtain the effect of feature reuse, especially in the back procreation, which contributes more to the spread of the gradient. The depth-wise layer builds a single convolution layer, which performs a lightweight filtering process. The 1×1 convolution layer is the third layer in the proposed architecture, which is without non-linearity. In the third layer, the output domain is linked to the ReLU6 component. ReLU6 is used to guarantee robustness made by low-precision situations and to devise the randomness of the model. Total layers have the same output quantity channels within the overall sequence. A 3×3 filtering size is common in contemporary architecture models, and dropout and batch normalization are used during the training phase. The residual component supports the gradient flow across the network and ReLU6 as the activation component through batch processing.

The confusion matrix is calculated by the actual classes versus the predicted classes. The correctly classified occurrence during classification is represented by the diagonal of confusion matrix.

$$P = \frac{TP}{TP + FP};$$

$$R = \frac{TP}{TP + FN};$$

$$F1 = 2 \times \frac{P \times R}{P + R};$$

$$ACC = \frac{TP + TN}{TP + TN + FB + FN}$$

For each class (white rust and non-disease class), P , R , $F1$, and ACC represent precision, recall, $F1$ -score, and accuracy, respectively, and TP , TN , FP , and FN represent true positive, true negative, false positive, and false negative for each class, respectively.

Table 2. Body architecture of MobileNet v2. Note: the network contains 19 residual bottleneck layers. Depth-wise convolution and spatial convolution are performed using 3×3 kernels, whereas pointwise convolution is performed using a 1×1 kernel.

Input	Type/Stride	Expansion Factor/ Block Repetition	Output Channels
$224 \times 224 \times 3$	conv2d/2	-/1	32
$112 \times 112 \times 32$	bottleneck/1	1/1	16
$112 \times 112 \times 16$	bottleneck/2	6/2	24
$56 \times 56 \times 24$	bottleneck/2	6/3	32
$28 \times 28 \times 32$	bottleneck/2	6/4	64
$14 \times 14 \times 64$	bottleneck/1	6/3	96
$14 \times 14 \times 96$	bottleneck/2	6/3	160
$7 \times 7 \times 160$	bottleneck/1	6/1	320
$7 \times 7 \times 320$	conv2d (1×1)/1	-/1	1280
$7 \times 7 \times 1280$	avgpool (7×7)/-	-/1	-
$1 \times 1 \times 1280$	conv2d (1×1)	-/-	k

2.5. Raspberry Pi Implementation

There are many operating systems for Raspberry Pi, such as Raspberry Pi OS, a supported operating system of Raspberry Pi organization, which was applied for the performance of Raspberry Pi 3, and Raspberry Pi OS (64 bit) with a desktop was installed. Firstly, a computer program called graphic user interface (GUI), which enables communication with electronic devices via visual indicator representation, was designed to allow the capture of photographs. This interface was processed through Python Tkinter to avoid third-party data libraries and to avoid sympathy arguments, suggesting two main Windows uses in the form of webcam captures and the region of interest frame screenshot. Detection at the same date and time were recorded in detail in saved files. For the evaluation of images from chrysanthemum plants, the predicted top class had a confidence value of 0.8.

3. Results and Discussion

3.1. Features Extraction

Learning rate was managed directly according to the network gradient through the training step, and it directly affected the tolerance capacity of the model. MobileNetv2 automatically extracted discriminative features from input images through multiple layers of convolutional and pooling operations without explicit human intervention. At lower layers, the network may capture low-level features such as edges, corners, and textures. As the information flows through deeper layers, the network becomes capable of capturing higher-level features, such as shapes, patterns, and structures, relevant to the specific disease being detected. MobileNetv2 was trained using the ImageNet dataset with a training accuracy rate of 99.81%. The training results are presented in Figure 3. As shown in Figure 3, the accuracy of the training and validation sets of the MobileNet v2 model, based on 99.81% training accuracy, is higher than the valid accuracy of 99.64%, and the cross entropy was also lower. We only iterated 35 epochs to compare the network initialization model with the transfer learning model due to limited computational resources. In Figure 3, the validating method for all layers of MobileNet v2 performed the best, achieving 99.64% classification accuracy on the validation set. The corresponding train–valid–test dataset is shown in Table 1.

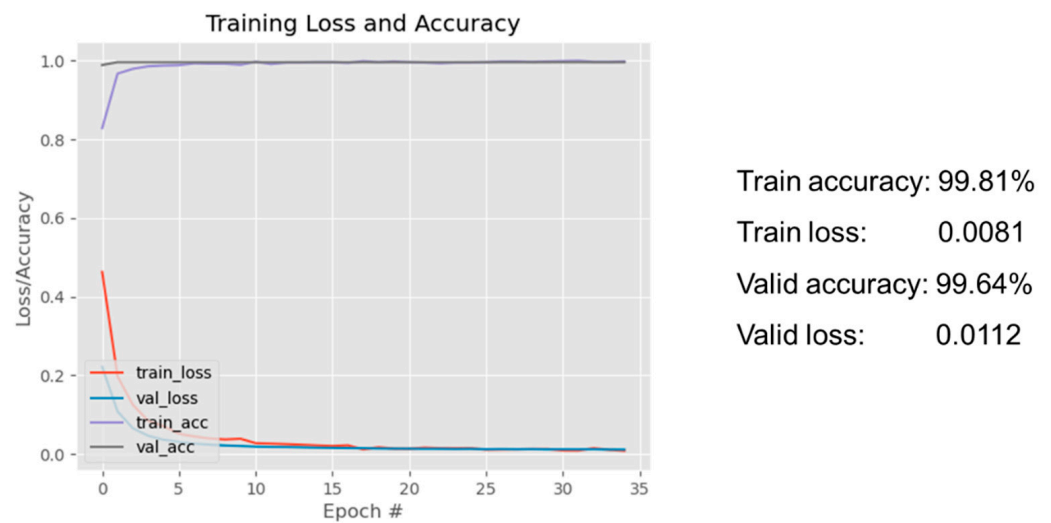


Figure 3. Training/validation accuracy and loss in the proposed dataset using the MobileNet v2 model, where train_loss, val_loss, train_acc, and val_acc represent train loss, valid loss, train accuracy, and valid accuracy, respectively.

Figure 4 shows the confusion matrix on the testing dataset. The accuracy was calculated from confusion matrix as follows:

$$\Delta = \frac{257 + 231}{257 + 231 + 2 + 5} = 0.9858$$

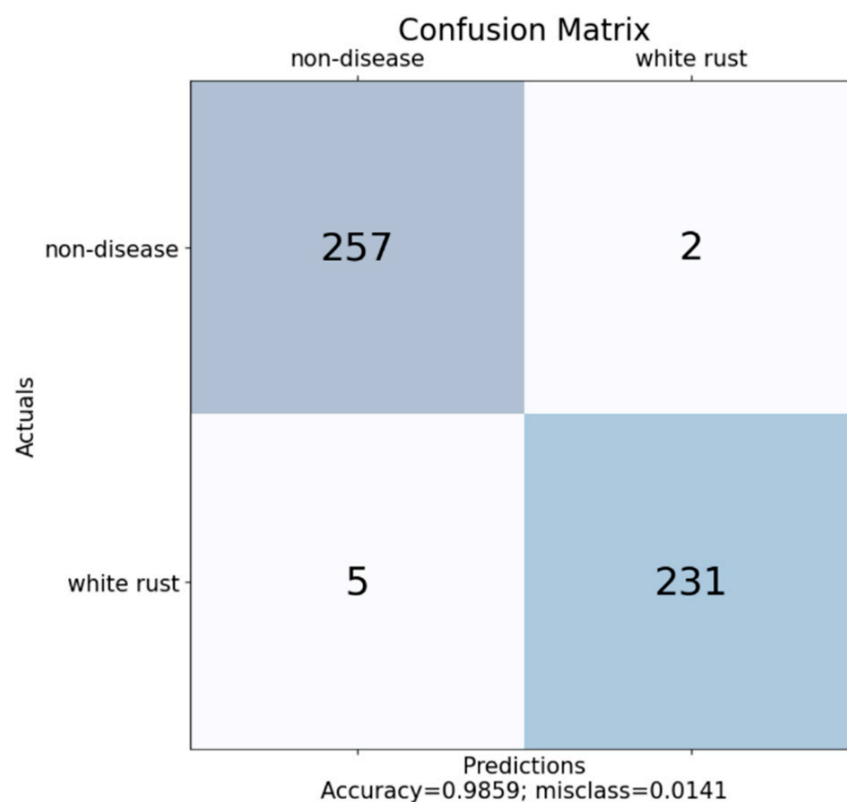


Figure 4. Confusion matrix.

Basically, from the confusion matrix, we can decide the performance in terms of final prediction for each layer to calculate the matrix. We further operate on the information from the confusion matrix in the network proposed.

The testing results are shown in Figure 5. It is shown that the network can predict non-disease and white rust leaves with a high accuracy of confidence (over 90%).

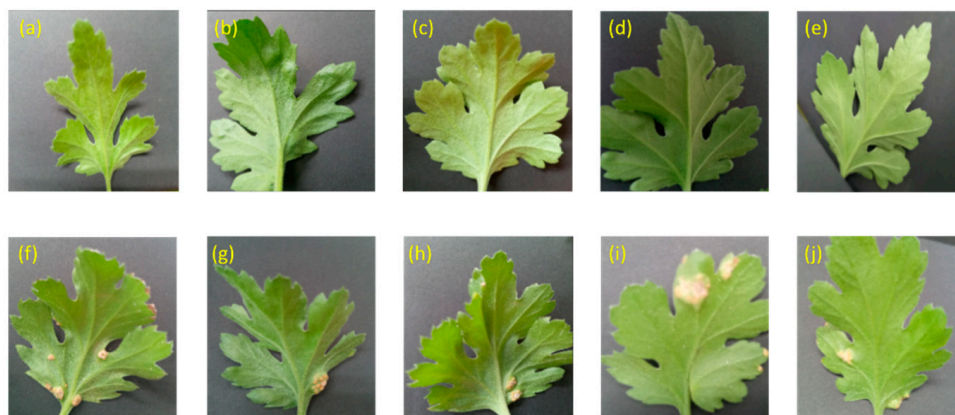


Figure 5. Testing results for the identification of non-disease and white rust disease. (a) Prediction: non-disease; confidence: 98.33%. (b) Prediction: non-disease; confidence: 91.72%. (c) Prediction: non-disease; confidence: 92.88%. (d) Prediction: non-disease; confidence: 97.21%. (e) Prediction: non-disease; confidence: 95.38%. (f) Prediction: white rust; confidence: 98.82%. (g) Prediction: white rust; confidence: 99.75%. (h) Prediction: white rust; confidence: 97.78%. (i) Prediction: white rust; confidence: 96.71%. (j) Prediction: white rust; confidence: 98.64%.

This study illuminates the fact that the MobileNet v2 model is the first suggested method that performs recognition of white rust according to the released results of each individual layer. This solution was encouraged by a real-life scheme, whereas the next study applies to opinions on network conditions of more accurate decision results related to almost all chrysanthemum diseases. From our experiments, we agree that large networks can be built that perform well for some pathogen classes.

3.2. Comparison with Other Models

In this section, white rust recognition examined the performance of three deep learning models using some evaluation approaches and testing image dataset to confirm the model with the best performance. Three models were constructed to apply the same optimizer, classifier, and learning rate. Then, a three-fold cross-validation method was achieved to reduce the over-fit and under-fit complements [38].

Table 3 shows the recognition results of the three models compared to the performance of the proposed model. Overall, they accomplished an accuracy of over 94%, and MobileNet v2 achieved the highest accuracy, precision, and recall at over 98%. In the accuracy comparison, DenseNet121 was recognized as having the second highest accuracy at 97%, whereas ResNet50 and VGG-19 achieved slightly lower accuracy at 95% and 94%, respectively. As a result, MobileNet v2 is the most suitable model for the white rust dataset.

Table 3. Performance of the proposed model compared to the other approaches.

Model	Accuracy	Precision	Recall
ResNet50 [17]	95.21%	95.12%	96.47%
VGG19 [39]	94.59%	95.26%	95.19%
DenseNet121	97.20%	98.06%	98.35%
MobileNet v2	99.24%	99.16%	98.39%

In this study, the MobileNet v2 model can automatically recognize chrysanthemum white rust in greenhouse conditions. Based on the experimental results, we considered four models that are used to select the most suitable deep learning models for the proposed dataset. The MobileNet v2 model and other models can correctly classify healthy leaves. Among them, MobileNet v2 was presented as the appropriate model because the highest

performance was decided according to the approaches of model comparison. In basic detail, the appropriate model for classification recognizes plant leaves which present as both non-diseased and diseased. It is complementary to the other models' classifications, but it is more powerful than the significant differences between healthy and unhealthy leaves. This model was evaluated by achieving further comparisons, and the experimental results indicate that the MobileNet v2 model achieved a classification accuracy 2% higher than that of the DenseNet121 model. The results prove that the MobileNet v2 model is robust and effective in the identification of chrysanthemum white rust, and it can significantly decrease processing times and farming costs if it is integrated into practical applications. Although MobileNet v2 achieves better accuracy than other models, it does have certain limitations, including its requirement of a high-speed computer and more extensive training time.

3.3. Qualitative Results with Raspberry Pi 3 Module and Comparison with Previous Works

a. Qualitative results with Raspberry Pi 3 module

Qualitative results were released from 400 images that were captured from a chrysanthemum plant that obtained visible degeneration due to disease (Figure 6). The seven training models were used to figure out the images (Table 4). The chrysanthemum plants were placed in a laboratory, and after three months of no disease, visible signs of white rust disease began to develop in the leaves within two weeks. According to Figures 2 and 6, we tried to establish the application of detecting white rust from an individual leaf (single detached leaf—shown in Figures 1 and 5). The Pi camera was focused on a single detached leaf, although a few side leaves were still presented in the camera's range (Figure 6). However, this did not affect its ability to detect chrysanthemum white rust (Table 4).

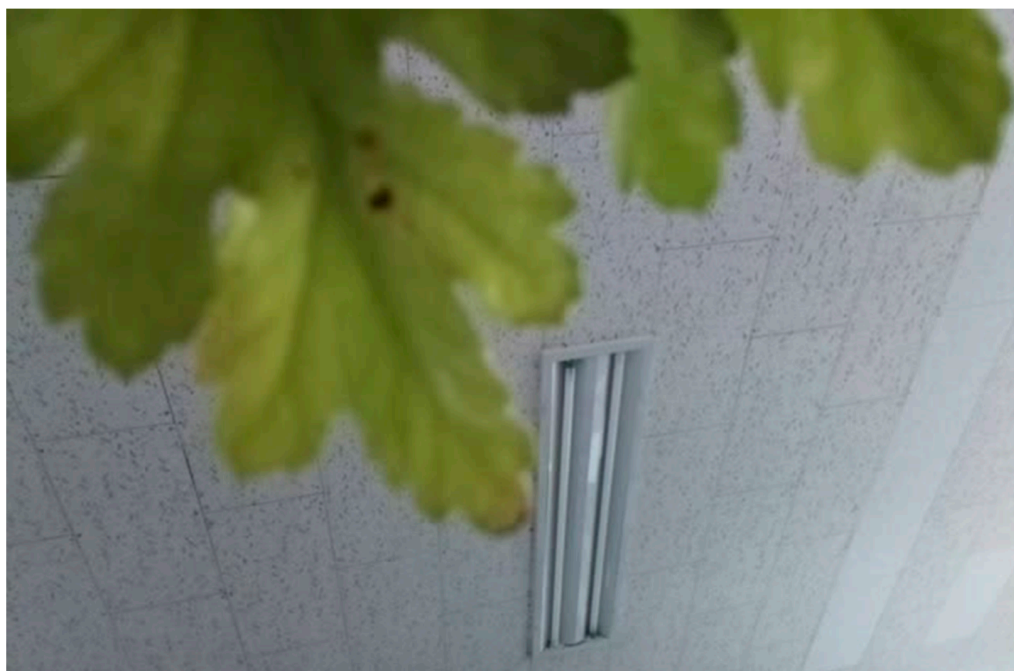


Figure 6. Testing results of Raspberry Pi camera module for chrysanthemum white rust disease detection: white rust; confidence: 98.42%.

Table 4 shows the qualitative results of the seven models using the Raspberry Pi 3 module. Overall, all four of the models had accuracy results of over 91.17% (ResNet50), followed by VGG19 (93.26%), DenseNet121 (95.31%), SqueezeNet (95.85%), MobileNet (96.72%), MobileNetv2-YOLOv3 (92.04%), and MobileNet v2 (97.12%).

Table 4. Performance of the qualitative model results with Raspberry Pi 3 module.

Model	Accuracy
ResNet50 [17]	91.17%
VGG19 [40]	93.26%
DenseNet121 [16]	95.31%
SqueezeNet [26]	95.85%
MobileNet [26]	96.72%
MobileNetv2-YOLOv3 [40]	92.04%
MobileNet v2 [ours]	97.12%

The highest accuracy was 97.12% (MobileNet v2). As a result, MobileNet v2 was validated as the suitable model to recognize white rust in chrysanthemums using the Raspberry Pi 3 system because it had the highest accuracy, precision, and recall. Raspberry systems can help provide a low-cost recognition of chrysanthemum white rust.

b. Comparison with previous works

In this section, the MobileNet v2 model for white rust detection in chrysanthemums is compared with some previous studies using the same model for plant disease detection.

Colombian researchers presented a novel method for diagnosing plant diseases, which involves capturing images of every part of the plant, such as leaves, fruits, and roots [26]. They used the images from the PlantVillage dataset and first removed the background noise [26]. Then, the tiles from selected images were reduced to eliminate any potential bias from the leaf shape [26]. Finally, cutting-edge tiny CNNs, with contexts created using little processing power, were trained on a new dataset of $85 \times 85 \times 3$ px images [26]. The accuracy rates of all models were over 95%, with SqueezeNet achieving a 95.05% accuracy rate and MobileNet achieving a 96.31% accuracy rate and providing the best performance [26]. The MobileNet model applied to our dataset achieved an accuracy of 96.72% (Table 4).

The MobileNetv2-YOLOv3 model was used to study tomato leaf spots and provide an early recognition method, achieving both good accuracy and real-time detection [40]. By improving the MobileNetv2-YOLOv3 lightweight model with MobileNetv2 as the backbone model, the progress of migration to mobile terminals was further enabled [40]. The experimental results showed a significant increase in the recognition effect of the improving model [40]. In the test dataset, the F1 score and average precision (AP) value were 94.13% and 92.53%, respectively [40]. In all test sets, the F1 score and AP value were 93.24% and 91.32%, respectively [40]. Applying the MobileNetv2-YOLOv3 model to our dataset yielded an accuracy of 92.04% (Table 4). Although the suitable model, MobileNet v2, produced good results for all data, some minor problems still need to be addressed:

- The MobileNet v2 model produces a small incorrect recalling prediction due to the wide range of small speckles that are found in individual plant leaves that present a significant degree of color changing, similar to other leaves' colors. Thus, a suggestion for future studies is to explore the application of multiple classification classes in each step to achieve the best model [41].
- Furthermore, Raspberry Pi 3 can be applied for white rust detection in chrysanthemums at a low cost and with low energy. It can easily be set up in smart farms, and it can be used to screen the growth conditions in plants and provide an early detection of white rust disease in chrysanthemums, and can also be used in human healthcare [42].

3.4. The Utility of MobileNet v2 and Raspberry Pi to Better Clarify the Motivation of Our Study

Our study emphasizes the cost-effectiveness of the recognition process, reducing reliance on manual labor. The high-throughput screening capability of MobileNet v2 allows for rapid processing of large amounts of data, leading to timely interventions and reductions in the spread of disease. The early detection is made possible by combining the utility of MobileNet v2 and Raspberry Pi, minimizing crop damage and treatment costs. The model's reliability provides consistent and accurate predictions to improve decision making for farmers. These advancements contribute to increased economic returns through reduced labor costs, improved resource allocation, and decreased crop losses.

3.5. The Improvements Can Be Introduced in the Future to Enhance the Tool's Effectiveness, and the Criteria Can Be Used to Build a Repeatable System

To enhance the effectiveness and reliability of the white rust detection system using MobileNet v2 on a Raspberry Pi, several key improvements can be applied. Dataset expansion by including diverse chrysanthemum images with various variations and conditions can improve the model's generalization capabilities. Fine-tuning the pre-trained MobileNet v2 model on specific chrysanthemum disease images can enhance its detection performance. Algorithm optimization techniques, such as model compression and quantization, as well as exploring alternative object detection algorithms, can improve accuracy and efficiency. Integrating real-time monitoring and alerts can provide timely notifications to farmers when white rust is detected.

3.6. The Potential Challenges and Corresponding Measures to Address the Uses of MobileNet v2 and Raspberry Pi for the Recognition of Chrysanthemum white Rust and Plant Health Conditions

Deploying a white rust detection system using Raspberry Pi and MobileNet v2 involves several challenges and considerations. The hardware limitations of Raspberry Pi, including limited processing power and memory, can be addressed through model optimization techniques, such as compression and quantization. Data collection and labeling for chrysanthemum images, especially for specific diseases such as white rust, can be labor-intensive, and collaboration with experts or data augmentation techniques can help overcome this challenge. Model training and optimization may require more powerful machines or cloud resources, and real-time performance can be improved through code and model optimization, including techniques such as quantization and hardware accelerators. Environmental conditions, such as lighting variations and occlusions, should be considered during system development, and robustness can be enhanced through image preprocessing and multiple camera angles. System deployment and maintenance require reliable enclosures, power management, and regular updates, while user training and support are crucial for farmers to effectively utilize the system for disease detection and plant health diagnosis.

3.7. Specific Applications Can Be Developed for Identifying Diseases in Agricultural Crops on Smartphones, and Benefits Can Be Gained from Their Use

Smartphone applications can be developed for identifying diseases in agricultural crops, offering several benefits to farmers. These applications can include features such as disease recognition and diagnosis, pest and pathogen monitoring, disease management and treatment recommendations, crop health monitoring, and knowledge- and information-sharing platforms. By leveraging image recognition algorithms and machine learning techniques, these apps enable farmers to detect diseases at early stages, accurately diagnose the problems, and receive prompt intervention and treatment recommendations. This leads to minimized crop losses, improved farm productivity, and cost and resource efficiency. Additionally, these applications facilitate improved crop management practices; optimize resource usage, foster knowledge sharing and collaboration among farmers, experts, and researchers; and ultimately contribute to increased productivity and sustainable farming practices.

4. Conclusions

This article documents a tool based on the convolution of intelligence neural networks for the detection, classification, and identification of white rust disease in chrysanthemums, defined in the non-disease and white rust disease database with statistical performance. In the training phase, ResNet50, VGG-19, DenseNet121, and MobileNet v2 were investigated. The results indicate that the MobileNet v2 model outperformed the other models in terms of accuracy, precision, and recall. The following future improvements should be introduced: (i) a criterion for constructing the repetitive system, upgrading networks until saturation is acquired, which results in a minimal increase in consequence, and an approach to prosperous training with different disease classes; and (ii) a collecting approach to identify more than one disease. Raspberry Pi 3 acts as a reference when establishing white rust detection in chrysanthemums. Although the identified study on white rust is the first step in a chain of identification both diseases and insects in chrysanthemum, there are many challenges ahead for us, and further efforts are needed. Moreover, various concentrations should be applied to the development of crop disease identification applications on smartphones because they are broadly accessible to farmers.

Author Contributions: T.K.N.: data curation, methodology, visualization, writing—original draft, and writing—review and editing; L.M.D.: investigation and software; T.-D.D.: formal analysis and software; J.H.L.: conceptualization, funding acquisition, validation, project administration, and supervision. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the project “Establishment of infrastructure for efficient management of clonal resources at the national seed cluster of central bank and sub-bank”, funded by the Rural Development Administration (RDA) (Project No. PJ0166632023).

Data Availability Statement: Not applicable.

Acknowledgments: We would like to thank the cooperation between three departments: Plant Biotechnology; Information and Communication Engineering, and Convergence Engineering for Intelligent Drone; and Aerospace System Engineering, and Convergence Engineering for Intelligent Drone, Sejong University, Seoul 05006, Republic of Korea. We would like to thank Ngoc Phi Nguyen for helping us accomplish our goals in this article.

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

References

1. Nguyen, T.K.; Lim, J.-H. Tools for Chrysanthemum genetic research and breeding: Is genotyping-by-sequencing (GBS) the best approach? *Hortic. Environ. Biotechnol.* **2019**, *60*, 625–635. [\[CrossRef\]](#)
2. Nguyen, T.K.; Jung, Y.O.; Lim, J.H. Tools for cut flower for export: Is it a genuine challenge from growers to customers? *Flower Res. J.* **2020**, *28*, 241–249. [\[CrossRef\]](#)
3. Nguyen, T.K.; Kwon, M.J.; Lim, J.H. Tools for controlling smart farms: The current problems and prospects in smart horticulture. *Flower Res. J.* **2019**, *27*, 226–241. [\[CrossRef\]](#)
4. Park, S.K.; Lim, J.H.; Shin, H.K.; Jung, J.A.; Kwon, Y.S.; Kim, M.S.; Kim, K.S. Identification of chrysanthemum genetic resources resistant to white rust caused by *Puccinia horiana*. *Plant Breed. Biotechnol.* **2014**, *2*, 184–193. [\[CrossRef\]](#)
5. Trolinger, J.C.; McGovern, R.J.; Elmer, W.H.; Rechcigl, N.A.; Shoemaker, C.M. Diseases of chrysanthemum. In *Handbook of Florists' Crops Diseases*; McGovern, R.J., Elmer, W.H., Eds.; Springer International Publishing: Cham, Switzerland, 2017; pp. 1–66. [\[CrossRef\]](#)
6. Ebrahimi, M.A.; Khoshtaghaza, M.H.; Minaei, S.; Jamshidi, B. Vision-based pest detection based on SVM classification method. *Comput. Electron. Agric.* **2017**, *137*, 52–58. [\[CrossRef\]](#)
7. Liu, W.; Wang, Z.; Liu, X.; Zeng, N.; Liu, Y.; Alsaadi, F.E. A survey of deep neural network architectures and their applications. *Neurocomputing* **2017**, *234*, 11–26. [\[CrossRef\]](#)
8. Yang, B.; Xu, Y. Applications of deep-learning approaches in horticultural research: A review. *Hort. Res.* **2021**, *8*, 123. [\[CrossRef\]](#)
9. Ren, C.; Kim, D.-K.; Jeong, D. A survey of deep learning in agriculture: Techniques and their applications. *J. Inf. Process. Syst.* **2020**, *16*, 1015–1033. [\[CrossRef\]](#)
10. Schmidhuber, J. Deep learning in neural networks: An overview. *Neural Netw.* **2015**, *61*, 85–117. [\[CrossRef\]](#)
11. Sze, V.; Chen, Y.-H.; Yang, T.-J.; Emer, J.S. Efficient processing of deep neural networks. *Synth. Lect. Comput. Archit.* **2020**, *15*, 1–341. [\[CrossRef\]](#)
12. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [\[CrossRef\]](#) [\[PubMed\]](#)

13. Bhuvana, J.; Mirnalinee, T.T. An approach to plant disease detection using deep learning techniques. *Iteckne* **2021**, *18*, 161–169. [[CrossRef](#)]
14. Pandian, J.A.; Kumar, V.D.; Geman, O.; Hnatiuc, M.; Arif, M.; Kanchanadevi, K. Plant disease detection using deep convolutional neural network. *Appl. Sci.* **2022**, *12*, 6982. [[CrossRef](#)]
15. Mishra, S.; Sachan, R.; Rajpal, D. Deep convolutional neural network based detection system for real-time corn plant disease recognition. *Procedia Comput. Sci.* **2020**, *167*, 2003–2010. [[CrossRef](#)]
16. Huang, G.; Liu, Z.; Maaten, L.V.D.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2261–2269.
17. 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, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
18. Vasilev, I.; Slater, D.; Spacagna, G.; Roelants, P.; Zocca, V. *Python Deep Learning: Exploring Deep Learning Techniques and Neural Network Architectures with Pytorch, Keras, and TensorFlow*; Packt Publishing Ltd.: Birmingham, UK, 2019.
19. Howard, A.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
20. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. MobileNetV2: Inverted residuals and linear bottlenecks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.
21. Howard, A.; Sandler, M.; Chen, B.; Wang, W.; Chen, L.C.; Tan, M.; Chu, G.; Vasudevan, V.; Zhu, Y.; Pang, R.; et al. Searching for MobileNetV3. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1314–1324.
22. Blahnik, V.; Schindelbeck, O. Smartphone imaging technology and its applications. *Adv. Opt. Technol.* **2021**, *10*, 145–232. [[CrossRef](#)]
23. Gonzalez-Huitron, V.; León-Borges, J.A.; Rodriguez-Mata, A.E.; Amabilis-Sosa, L.E.; Ramírez-Pereda, B.; Rodriguez, H. Disease detection in tomato leaves via CNN with lightweight architectures implemented in Raspberry Pi 4. *Comput. Electron. Agric.* **2021**, *181*, 105951. [[CrossRef](#)]
24. Bi, L.; Hu, G. Improving image-based plant disease classification with generative adversarial network under limited training set. *Front. Plant Sci.* **2020**, *11*, 583438. [[CrossRef](#)]
25. Mohanty, S.P.; Hughes, D.P.; Salathé, M. Using deep learning for image-based plant disease detection. *Front. Plant Sci.* **2016**, *7*. [[CrossRef](#)] [[PubMed](#)]
26. Restrepo-Arias, J.F.; Branch-Bedoya, J.W.; Awad, G. Plant Disease Detection Strategy Based on Image Texture and Bayesian Optimization with Small Neural Networks. *Agriculture* **2022**, *12*, 1964. [[CrossRef](#)]
27. Liu, J.; Wang, X. Plant diseases and pests detection based on deep learning: A review. *Plant Methods* **2021**, *17*, 22. [[CrossRef](#)] [[PubMed](#)]
28. Li, Y.; Wang, H.; Dang, L.M.; Sadeghi-Niaraki, A.; Moon, H. Crop pest recognition in natural scenes using convolutional neural networks. *Comput. Electron. Agric.* **2020**, *169*, 105174. [[CrossRef](#)]
29. Karar, M.E.; Alsunaydi, F.; Albusaymi, S.; Alotaibi, S. A new mobile application of agricultural pests recognition using deep learning in cloud computing system. *Alex. Eng. J.* **2021**, *60*, 4423–4432. [[CrossRef](#)]
30. Nguyen, T.K.; Dang, L.M.; Song, H.-K.; Moon, H.; Lee, S.J.; Lim, J.H. Wild chrysanthemums core collection: Studies on leaf identification. *Horticulturae* **2022**, *8*, 839. [[CrossRef](#)]
31. Bi, C.; Wang, J.; Duan, Y.; Fu, B.; Kang, J.-R.; Shi, Y. MobileNet based apple leaf diseases identification. *Mob. Netw. Appl.* **2022**, *27*, 172–180. [[CrossRef](#)]
32. Ou, L.; Zhu, K. Identification algorithm of diseased leaves based on MobileNet model. In Proceedings of the 2022 4th International Conference on Communications, Information System and Computer Engineering (CISCE), Shenzhen, China, 27–29 May 2022; pp. 318–321.
33. Akiyama, T.; Kobayashi, Y.; Sasaki, Y.; Sasaki, K.; Kawaguchi, T.; Kishigami, J. Mobile leaf identification system using CNN applied to plants in Hokkaido. In Proceedings of the 2019 IEEE 8th Global Conference on Consumer Electronics (GCCE), Osaka, Japan, 15–18 October 2019; pp. 324–325.
34. Hong, Q.; Jiang, L.; Zhang, Z.; Ji, S.; Gu, C.; Mao, W.; Li, W.; Liu, T.; Li, B.; Tan, C. A Lightweight model for wheat ear fusarium head blight detection based on RGB images. *Remote Sens.* **2022**, *14*, 3481. [[CrossRef](#)]
35. Borhani, Y.; Khoramdel, J.; Najafi, E. A deep learning based approach for automated plant disease classification using vision transformer. *Sci. Rep.* **2022**, *12*, 11554. [[CrossRef](#)]
36. Cui, X.; Goel, V.; Kingsbury, B. Data Augmentation for deep neural network acoustic modeling. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2015**, *23*, 1469–1477. [[CrossRef](#)]
37. Montserrat, D.M.; Lin, Q.; Allebach, J.; Delp, E.J. Training object detection and recognition CNN models using data augmentation. *Electron. Imaging* **2017**, *2017*, 27–36. [[CrossRef](#)]
38. Bergmeir, C.; Hyndman, R.J.; Koo, B. A note on the validity of cross-validation for evaluating autoregressive time series prediction. *Comput. Stat. Data Anal.* **2018**, *120*, 70–83. [[CrossRef](#)]
39. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.

40. Liu, J.; Wang, X. Early recognition of tomato gray leaf spot disease based on MobileNetv2-YOLOv3 model. *Plant Methods* **2020**, *16*, 83. [[CrossRef](#)] [[PubMed](#)]
41. Dang, L.M.; Wang, H.X.; Li, Y.F.; Nguyen, T.N. Explainable artificial intelligence: A comprehensive review. *Artif. Intell. Rev.* **2022**, *55*, 3503–3568. [[CrossRef](#)]
42. Dang, L.M.; Piran, M.J.; Han, D.G.; Min, K.B.; Moon, H.J. A survey on Internet of things and cloud computing for healthcare. *Electronics* **2019**, *8*, 768. [[CrossRef](#)]

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