Article

Improved Multi-Plant Disease Recognition Method Using Deep Convolutional Neural Networks in Six Diseases of Apples and Pears

Yeong Hyeon Gu ^{1,†}, Helin Yin ^{1,†} , Dong Jin ^{1,2} , Ri Zheng ^{1,2} and Seong Joon Yoo ^{1,*}

¹ Department of Computer Science and Engineering, Sejong University, Seoul 05006, Korea; yhgu@sejong.ac.kr (Y.H.G.); yinhelin0608@gmail.com (H.Y.); kimdoubled@gmail.com (D.J.); zrchuangri@gmail.com (R.Z.)

² Department of Convergence Engineering for Intelligent Drone, Sejong University, Seoul 05006, Korea

* Correspondence: sjyoo@sejong.ac.kr

† These authors have contributed equally to this work and share first authorship.

Abstract: Plant diseases are a major concern in the agricultural sector; accordingly, it is very important to identify them automatically. In this study, we propose an improved deep learning-based multi-plant disease recognition method that combines deep features extracted by deep convolutional neural networks and k -nearest neighbors to output similar disease images via query image. Powerful, deep features were leveraged by applying fine-tuning, an existing method. We used 14,304 in-field images with six diseases occurring in apples and pears. As a result of the experiment, the proposed method had a 14.98% higher average similarity accuracy than the baseline method. Furthermore, the deep feature dimensions were reduced, and the image processing time was shorter (0.071–0.077 s) using the proposed 128-sized deep feature-based model, which processes images faster, even for large-scale datasets. These results confirm that the proposed deep learning-based multi-plant disease recognition method improves both the accuracy and speed when compared to the baseline method.

Keywords: deep feature; fine-tuning; k -nearest neighbors; plant disease recognition; transfer learning

Citation: Gu, Y.H.; Yin, H.; Jin, D.; Zheng, R.; Yoo, S.J. Improved Multi-Plant Disease Recognition Method Using Deep Convolutional Neural Networks in Six Diseases of Apples and Pears. *Agriculture* **2022**, *12*, 300. <https://doi.org/10.3390/agriculture12020300>

Academic Editor: Antonello Bonfante

Received: 11 January 2022

Accepted: 14 February 2022

Published: 21 February 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/>).

1. Introduction

In agriculture, damage caused by diseases and pests seriously affects crop production worldwide. The control and prevention of diseases and pests can minimize damage and economic losses to farms. Pears and apples are the most widely grown fruits in the world and are economically very beneficial. However, owing to many cases of diseases and pests in recent years, the production and quality of pears and apples have been severely affected. Generally, it is necessary to commission experts to identify the diseases or pests accurately, which requires a lot of time and great expense. If prescriptions and countermeasures for the diseases or pests are not provided in a timely manner, the problem can eventually spread, even to the farmhouse, resulting in secondary damage. Therefore, there is a growing demand for fast and accurate disease and pest diagnosis models to minimize the damage to crops.

Generally, it is very difficult to recognize crop diseases and pests because of a lack of training data and a variety of symptoms [1]. However, recent advances in deep learning techniques have made it possible to efficiently recognize diseases and pests in complex disease symptoms, and such techniques are being successfully applied in many fields [2]. Recently, machine and deep learning techniques have been applied in many studies on disease and pest recognition methods, among which deep convolutional neural network (DCNN) algorithms have shown a good performance [1,3–5].

A deep learning model typically requires extensive data, computational time, and effort to train [6]. Furthermore, critical issues, such as over-fitting, may arise [7,8]. Disease

and pest images have a seasonality, and it is difficult to collect enough accurate images because of a shortage of experts [9]. Furthermore, it is more difficult to collect images of diseases and pests that are uncommon.

Accordingly, a transfer learning method is used when there is not enough data or if the model is trained poorly [6,10,11]. Transfer learning is a machine learning method that leverages the knowledge gained from solving existing problems to solve other, similar problems, and it is commonly used in computer vision, natural language processing, and other tasks. However, deep learning models (e.g., DCNN) require extensive computational resources for training from the beginning. Therefore, current models are fine-tuned and applied to specific domains using the data of pre-trained models [6,12]. Alternatively, as another transfer learning strategy, feature extraction can be used. Here, feature extraction refers to the process of extracting features from the pre-trained models, such as ResNet [13], VGG [14], and InceptionV3 [15], which have been trained using a large-scale dataset (e.g., the ImageNet dataset). The features extracted from the DCNN in this way are called “deep features”. A deep learning model creates the most effective feature map during processing by mapping inputs and target data via multi-level representation learning [16]. This effectively solves the limitation of traditional plant classification models using hand-crafted features [17,18].

Previous studies on disease and pest recognition use single recognition methods to present a single result to the user. However, in this case, the recognition model is not 100% accurate and may output incorrect results owing to misclassification. To address this problem, a multi-recognition method, such as Google Vision API, presents several candidates to users and allows them to intervene and make the final decision. The author of [19] recognized 58 types of diseases and pests for 25 types of crops, including apples, cabbages, and peppers, and implemented a multi-recognition method that outputs classes with a confidence value in the softmax layer. The study used several models (e.g., VGG and GoogleNet) and recorded a recognition accuracy of approximately 99% using the VGG model. A content-based image retrieval (CBIR) technique [20] can also be used alongside the multi-recognition method. CBIR extracts the features through descriptors from the content, such as the color, shape, and texture in the image, and it outputs the most similar images to a query image using similarity comparison between features. Many studies [21–25] on disease and pest recognition have already used the CBIR technique, but they showed relatively low recognition accuracies because of the limitations of the descriptors.

As mentioned, conventional deep learning-based disease and pest recognition studies show a high classification accuracy. However, there are difficulties in the final decision-making step, and although CBIR-based disease and pest recognition support this process, the problem of relatively low search accuracy persists. To address this problem, Yin et al. [9] proposed a novel disease and pest recognition model that applies deep features extracted using a deep learning model. They proposed a disease and pest recognition model that combined ImageNet dataset-based deep features and a k -nearest neighbor (KNN) algorithm. The model was applied to the diseases and pests of hot peppers, recording top ten accuracies of 85.6% and 93.62%, respectively. However, it was not a disease and pest domain-specific model because the study used deep features extracted from a pre-trained model with conventional ImageNet dataset weights. Therefore, in this study, we propose an improved, fine-tuning method-based multi-disease recognition method.

The contributions of this study are as follows: An improved deep learning-based multi-plant disease recognition method is proposed and applied to six diseases occurring in pears and apples. The data used are real, in-field images collected by agriculture experts, not laboratory-created images. The proposed model combines a KNN algorithm with the more powerful, deep features extracted from the fine-tuned convolutional neural network (CNN) models, and according to the performance measurement results, its similarity accuracy is approximately 14.98% higher than the baseline model [9]. Furthermore, we introduced a method to reduce the dimensions of deep features. As a result, the image processing time of the proposed model, based on deep features with a size of 128, was 0.071–0.077 s,

even for large-scale datasets. Hence, the proposed deep learning-based multi-plant disease recognition method improves both the accuracy and speed.

The remainder of this paper is organized as follows: Section 2 introduces the dataset used and the proposed method. Section 3 explains the experimental setup, and Section 4 provides the experimental results and discussion. Finally, Section 5 concludes the paper.

2. Materials and Methods

2.1. Dataset Description

In this study, we used a disease image dataset for six diseases (scab, black necrotic leaf spot, fire blight, anthracnose, Marssonina blotch, and Alternaria leaf spot) occurring in apples and pears. Sample images of the six diseases are shown in Figure 1. This dataset was provided by the National Institute of Horticultural, the Herbal Science of South Korea, and AI Hub (<https://aihub.or.kr/>, accessed on 10 January 2022).

Figure 1. Sample images of six different categories used in the proposed work.

The “fire blight image dataset” of AI Hub (<https://aihub.or.kr/aidata/30732>, accessed on 10 January 2022) is a collection of diseases frequently occurring in pears and apples and comprises nine classes and 211,555 images. Furthermore, it provides bounding box annotation information for the ROI to facilitate disease detection. The dataset includes many images that are not suitable for training because of ambiguous symptoms, an inappropriate image capturing distance, and out-of-focus images, among other issues. Accordingly, such images were excluded from the training and testing datasets.

In this study, we used cropped images containing the symptom parts instead of the original disease images. Image cropping refers to the task of removing unnecessary parts from the image, which can accelerate the image recognition process and improve the recognition accuracy [26,27]. In this study, the image cropping was performed manually by fruit disease and pest experts with a rectangular box to include the symptom part, as shown in Figure 2, and several cropped images could be generated from a single original image. Table 1 shows the statistics of the data used. In this study, six disease datasets of apples and pears were used, and the data statistics are shown in Table 1. From the original

images of 4202, a total of 14,304 cropped images were created through image cropping, as in Figure 2.

Figure 2. Image cropping process.

Table 1. Summary of the six disease dataset.

Plant	Class	Original Images	Cropped Images
Pear	Fire blight (<i>Erwinia amylovora</i>)	357	865
	Scab (<i>Venturia nashicola</i>)	770	3911
	Black necrotic leaf spot (<i>Apple stem grooving virus</i>)	130	894
Apple	Marssonia blotch (<i>Diplocarpon mali</i>)	976	2711
	Alternaria leaf spot (<i>Alternaria mali</i>)	990	4416
	Anthraxnose (<i>Glomerella cingulata</i>)	979	1507
Total		4202	14,304

2.2. Pre-Trained Models and Fine-Tuning

The DCNN is widely used in the computer vision field. It generally consists of convolutional layers, a pooling layer, a fully connected layer, and an output layer. The training images go through multiple convolutional layers and a pooling layer to extract the local and global features of the images, and the final feature map is learned.

In this study, we extracted deep features from an ImageNet weight-based, pre-trained model. Here, deep features refer to features extracted from the DCNN. For this, we used seven pre-trained models (VGG16, VGG19, ResNet 50, Inception ResNet, NASNet, EfficientNetB0, and DenseNet 121) as base models and measured their performance.

In a previous study, Yin et al. [9] used deep features extracted from the “Global Pooling Layer” of an ImageNet weight-based, pre-trained model, as shown in Figure 3a. In this study, we extracted improved deep features by fine-tuning a pre-trained model, which was trained using ImageNet weights, as shown in Figure 3b. In the proposed method, the existing convolutional and pooling layers were fixed, and the fully connected and softmax

layers were modified. The number of nodes in the fully connected layer was changed to 128, meaning that the size of the deep features extracted in the fine-tuned model was 128. Furthermore, the number of outputs of softmax was changed to six because six diseases occurring in apples and pears were recognized.

We divided the dataset into a training set and a test set to fine-tune the pre-trained model. The training and test sets were randomly chosen from each category, with ratios of 90% and 10%, respectively. The batch size was set to 256, and a categorical cross-entropy loss function was used. Stochastic gradient descent was used as an optimizer, and a learning rate of 0.001 was used in this study. The epoch was set to 500, and early stopping was added to avoid over-fitting problems. The early stopping was set to terminate the training when the validation accuracy no longer improved over the next 20 times.

Figure 3. Comparison of the baseline and proposed methods: feature extraction from (a) the ImageNet-based pre-trained CNN model and (b) the fine-tuned CNN models.

2.3. K-Nearest Neighbor

The KNN algorithm is a supervised learning method that classifies unlabeled observations based on the most similarly labeled examples in the attribute space [28]. When classifying, this algorithm refers to the information of k instances around the given point and makes a final decision through the majority voting method. The KNN algorithm calculates the distance between the input data and all data. It refers to the information of k instances near the input data to determine the class of the input data.

In the KNN algorithm, the value of k and the distance function are important elements. Here, the value of k is set by the user, and the information of k instances near the input data is used. As such, the classification performance varies depending on the value of k . Thus, it is very important to find an appropriate value of k [29]. However, the search model proposed in this study, unlike the classification model, is not significantly affected by the value of k because there is no majority voting. Therefore, we set the value of k to 10 for the convenience of performing the measurement.

The distance function calculates the similarity based on the distances among all data. The use of an appropriate distance function can improve model performance because it has the advantage of handling high-dimensional data well or speeding up calculations. In

this study, we used the Bray–Curtis distance [30] as the distance function. The Bray–Curtis distance is a normalization method that is commonly used in ecology and environmental sciences. The Bray–Curtis distance between vectors A and B can be calculated with Equation (1). A and B are vectors with a length of N . The Bray–Curtis distance has a range of values of $(0, 1)$; as it approaches zero, it means that they are more similar.

$$\text{Bray–Curtis distance}(A, B) = \frac{\sum_{i=1}^N |A_i - B_i|}{\sum_{i=1}^N A_i + \sum_{i=1}^N B_i}. \quad (1)$$

KNN algorithms are commonly used for classification, but they can also be used for searching. Apart from majority voting (the final step), the index number and the distance of the nearest points from the input data can be returned.

2.4. Proposed Method

Figure 4 shows the flow of the improved multi-plant disease recognition method proposed in this study. First, a disease symptom showing a region of interest (ROI) in the input image is cropped to increase the recognition accuracy. Next, disease and pest data are used to fine-tune various pre-trained CNN models, which are used as deep feature extractors. Then, the extracted deep features are applied to the KNN algorithm to facilitate the search for similar symptoms. During the testing process, each cropped image is input into the fine-tuned, deep feature extractor; afterward, it is input into the pre-trained KNN model to output k disease images that are most similar to the input in the vector space.

Figure 4. Architecture of the proposed multi-plant disease-recognition model.

3. Experiments

3.1. Experimental Setup

All models used in this study were compiled with graphical processing-unit (GPU) support. The experimental work was performed using Python v.3.7 on a desktop computer running Windows with two Nvidia GeForce RTX 3090 Ti GPUs with a 24-GB memory. Ten-

sortFlow, Keras, Scikit-learn, OpenCV, and Matplotlib libraries were used in the Anaconda Jupyter Notebook.

3.2. Performance Metrics

We used *precision* as a metric to evaluate the performance of the classification model for six diseases occurring in apples and pears, as shown in Equation (2). Here, *True Positive* is the number of correctly classified diseased images in each category, and *False Positive* is the number of misclassified images in all other categories except for the relevant category. The *precision* is used when evaluating the classification performance of the fine-tuned model.

$$precision = \frac{1}{N} \sum_{i=1}^N \frac{True\ Positive}{True\ Positive + False\ Positive}. \quad (2)$$

Because the proposed model is a similarity-based plant disease recognition model, we used *similarity accuracy* as another performance evaluation metric, as shown in Equation (3). Here, *relevant images* refer to the number of images that have the same class as the query image, and *retrieved images* refer to the number of output images of the proposed model. N is the number of images belonging to each disease class in the test image set, and i is the index of each query image.

$$similarity\ accuracy = \frac{1}{N} \sum_{i=1}^N \frac{|\{relevant\ images\} \cap \{retrieved\ images\}|}{|\{retrieved\ images\}|}. \quad (3)$$

4. Results and Discussion

This study proposes a method for building a similarity-based disease image-recognition model using deep features extracted by a fine-tuned CNN model. In this section, we examine the efficiency of the proposed fine-tuning method.

Table 2 shows the performance measurement results after fine-tuning the seven pre-trained models. All models exhibited a precision greater than 90% and, among them, the ResNet50 model showed the highest precision (98.83%). Figure 5 shows the fine-tuning training process for the ResNet50 model.

Table 2. Performance results of the seven fine-tuned models used in this study.

Model	Precision
ResNet50	98.83%
VGG16	94.53%
VGG19	95.70%
Inception ResNet	93.12%
NASNet Large	91.48%
EfficientNetB0	98.05%
DenseNet121	97.75%

Table 3 shows a performance comparison between the proposed and the baseline methods (Yin et al. [9]). Each cell in the table contains two values. The first value is the similarity accuracy when the proposed method is applied, and the value in parentheses is the performance difference between the proposed and baseline methods. As shown in the table, the proposed fine-tuned method exhibited a higher similarity accuracy than the baseline method in every category. In each model, the proposed method improved its performance by 10.07–33.55% over the baseline method. Figure 6 shows the average similarity accuracy comparison, showing an average performance improvement of approximately 14.98%.

Figure 5. Accuracy and loss with the epoch for the ResNet50 model.

Table 3. Performance comparison between seven fine-tuned models and the baseline.

	ResNet50	VGG16	VGG19	Inception ResNet	NASNet Large	Efficient NetB0	DenseNet121
Similarity Accuracy/Compared with the Baseline Method [9]							
Fire Blight	99.58%/ (+26.44%)	79.35%/ (+16.61%)	89.61%/ (+26.33%)	81.06%/ (+31.22%)	81.46%/ (+25.40%)	99.28%/ (+62.25%)	90.07%/ (+25.15%)
Scab	99.79%/ (+5.02%)	96.90%/ (+4.61%)	97.81%/ (+5.82%)	95.34%/ (+5.36%)	93.19%/ (+3.88%)	99.60%/ (+17.56%)	94.40%/ (+2.73%)
Black necrotic leaf spot	99.98%/ (+8.31%)	92.90%/ (+10.41%)	96.64%/ (+13.69%)	88.28%/ (+18.82%)	89.40%/ (+13.02%)	99.87%/ (+36.09%)	93.76%/ (+9.33%)
Marssonina blotch	99.66%/ (+25.19%)	86.57%/ (+19.13%)	91.87%/ (+22.60%)	79.23%/ (+18.01%)	76.56%/ (+13.14%)	99.10%/ (+49.33%)	87.95%/ (+13.52%)
Alternaria leaf spot	99.80%/ (+8.86%)	93.73%/ (+8.19%)	95.23%/ (+7.97%)	89.47%/ (+8.22%)	90.33%/ (+9.54%)	99.54%/ (+23.45%)	90.85%/ (+2.24%)
Anthracoise	99.87%/ (+4.81%)	95.47%/ (+1.49%)	97.64%/ (+3.44%)	95.75%/ (+3.02%)	96.56%/ (+2.21%)	99.85%/ (+12.62%)	96.98%/ (+4.25%)
Average	99.78%/ (+13.10%)	90.82%/ (+10.07%)	94.80%/ (+13.31%)	88.19%/ (+14.11%)	87.92%/ (+11.20%)	99.54%/ (+33.55%)	92.34%/ (+9.54%)

Figure 6. Performance comparison between the proposed and baseline methods.

In the proposed fine-tuning method, a fully connected layer consisting of 128 nodes was used, and the performance was higher than that of the baseline model. However, com-

pared with the conventional ResNet50 model, in which a fully connected layer consisting of 2048 nodes is used, this study uses relatively few nodes. Therefore, we also measured the performance by applying the fine-tuning method to a fully connected layer consisting of 256, 512, and 1024 nodes, and the results are shown in Table 4. The first row in Table 4 shows the classification accuracy of the fine-tuned model with five different numbers of nodes. The second row is the similarity accuracy when using the deep features extracted in the fine-tuned model. For the fine-tuned model with five different numbers of nodes, the precision and similarity accuracy were 98.83–99.14% and 99.74–99.80%, respectively. As a result of fine-tuning the model with five different numbers of nodes, we found no significant difference in performance. This demonstrates that the fine-tuning method used in this study can help improve model performance.

Table 4. Performance evaluation on fine-tuning with a different number of nodes.

Performance	ResNet50 with Different Number of Nodes				
	128	256	512	1024	2048
Precision	98.83%	98.83%	99.14%	98.91%	99.06%
Similarity accuracy	99.78%	99.74%	99.77%	99.80%	99.78%

The proposed similarity-based disease-recognition method calculates the distance between all vectors through the KNN algorithm and outputs most similar to the k vectors from the input vector. Here, a vector refers to a deep feature extracted from the image. Owing to the nature of the KNN algorithm, where the distance between all vectors must be measured, the image search speed may vary depending on the length of the vectors. In the experiment, we measured the image search speed when using the deep features extracted from the fine-tuned fully connected layer consisting of 128, 256, 512, 1024, and 2048 nodes. The image search speed refers to the time consumed to extract the deep features by inputting an image into the fine-tuned ResNet50 model and extracting similar images by inputting the extracted deep features into the KNN model. To compare the search time in the model according to the number of data, we used datasets consisting of 100, 1000, 10,000, 100,000, and 1,000,000 images. Figure 7 shows the search time comparison when differently sized deep features are used. The x-axis of the figure represents the number of images used, and the y-axis represents the search time (s/image). As shown in Figure 7, when 128-sized deep features are used, the processing time per image is 0.071–0.077 s, which is the fastest search speed. There was no significant difference in the search time between the five models in the section with 100–10,000 images; however, the difference was quite noticeable in the cases of 100,000 and 1,000,000 images. Hence, the search times increased as the size of the deep features and the dataset increased.

As seen in Table 4, there was no significant difference in the similarity accuracy when the fine-tuning method was applied to various, fully connected layers; however, the size of the deep features extracted in each layer affected the search time (as shown in Figure 7). Among them, the use of 128-sized deep features resulted in the highest efficiency, showing high performance and fast search speed, even for large datasets.

In the past, several studies have been proposed for plant disease detection from images. We have compared our proposed method with some of the existing studies on our disease dataset. The performance comparison with existing methods is represented in Table 5. The proposed method has shown its superiority over the methods in Yin et al. [9], Elhassouny and Smarandache [31], Kathiresan et al. [32], and Sagar and Jacob [33].

Figure 7. Comparison of image search time according to the size of deep features.

Table 5. Comparison of the result with existing models.

Author(s)	Method(s)	Accuracy
Yin et al. [9]	Pre-trained model (Transfer Learning), KNN	86.68%
Elhassouny and Smarandache [31]	MobileNet	96.88%
Kathiresan et al. [32]	Modified Densenet-169 (Transfer learning), GAN Augmentation	96.97%
Sagar and Jacob [33]	Pre-trained ResNet50 (Transfer Learning)	98.52%
Proposed Method	Pre-trained model (Transfer learning) with fine-tuning, KNN	99.78%

5. Conclusions

In this study, we proposed an improved deep learning-based multi-disease recognition method. The proposed model helps users make a final decision by outputting disease images that are most similar to the input query image. We leveraged more powerful deep features by applying a fine-tuning method to the model proposed in a previous study [9] to recognize six diseases occurring in apples and pears. As a result, the proposed method improved performance by 14.98% on average over the baseline method. Furthermore, we introduced a method to reduce the dimensions of the deep features, which resulted in faster image processing speeds, even for large-scale datasets.

In future work, we will apply the proposed model to images of additional disease types to expand its practical utility in farmhouses.

Author Contributions: Conceptualization, Y.H.G., H.Y. and R.Z.; investigation, H.Y., Y.H.G. and D.J.; methodology, D.J. and R.Z.; project administration, Y.H.G. and S.J.Y.; writing—original draft preparation, H.Y. and R.Z.; writing—review and editing, H.Y., Y.H.G., D.J. and S.J.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work was carried out with the support of the “Cooperative Research Program for Agriculture Science and Technology Development (Project No. PJ015638, construction of fruit tree fire blight early diagnosis system)” and the Rural Development Administration, Republic of Korea.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data is available from its original source as cited in the article. The image dataset of diseases is available at AI Hub (<https://aihub.or.kr/aidata/30732>).

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

References

1. Dhaka, V.S.; Meena, S.V.; Rani, G.; Sinwar, D.; Kavita, K.; Ijaz, M.F.; Woźniak, M. A survey of deep convolutional neural networks applied for prediction of plant leaf diseases. *Sensors* **2021**, *21*, 4749. [\[CrossRef\]](#) [\[PubMed\]](#)
2. Yu, H.; Miao, C.; Leung, C.; White, T.J. Towards AI-powered personalization in MOOC learning. *Npj Sci. Learn.* **2017**, *2*, 1–5. [\[CrossRef\]](#) [\[PubMed\]](#)
3. Traore, B.B.; Kamsu-Foguem, B.; Tangara, F. Deep convolution neural network for image recognition. *Ecol. Inform.* **2018**, *48*, 257–268. [\[CrossRef\]](#)
4. Abeywardhana, D.L.; Dangalle, C.D.; Nugaliyadde, A.; Mallawarachchi, Y. Deep learning approach to classify Tiger beetles of Sri Lanka. *Ecol. Inform.* **2021**, *62*, 101286. [\[CrossRef\]](#)
5. Atila, Ü.; Uçar, M.; Akyol, K.; Uçar, E. Plant leaf disease classification using EfficientNet deep learning model. *Ecol. Inform.* **2021**, *61*, 101182. [\[CrossRef\]](#)
6. Kaya, A.; Keceli, A.S.; Catal, C.; Yalic, H.Y.; Temucin, H.; Tekinerdogan, B. Analysis of transfer learning for deep neural network based plant classification models. *Comput. Electron. Agric.* **2019**, *158*, 20–29. [\[CrossRef\]](#)
7. Nitish, S.; Geoffrey, H.; Alex, K.; Ilya, S.; Ruslan, S. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
8. Penatti, O.A.B.; Nogueira, K.; Dos Santos, J.A. Do deep features generalize from everyday objects to remote sensing and aerial scenes domains? In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Boston, MA, USA, 7–12 October 2015; pp. 44–51. [\[CrossRef\]](#)
9. Yin, H.; Gu, Y.H.; Park, C.J.; Park, J.H.; Yoo, S.J. Transfer learning-based search model for hot pepper diseases and pests. *Agriculture* **2020**, *10*, 439. [\[CrossRef\]](#)
10. Zhuang, F.; Qi, Z.; Duan, K.; Xi, D.; Zhu, Y.; Zhu, H.; Xiong, H.; He, Q. A Comprehensive Survey on Transfer Learning. *Proc. IEEE* **2021**, *109*, 43–76. [\[CrossRef\]](#)
11. Deng, Z.; Zhang, X.; Zhao, Y. Transfer learning based method for frequency response model updating with insufficient data. *Sensors* **2020**, *20*, 5615. [\[CrossRef\]](#)
12. Dourado-Filho, L.A.; Calumby, R.T. An experimental assessment of deep convolutional features for plant species recognition. *Ecol. Inform.* **2021**, *65*, 101411. [\[CrossRef\]](#)
13. 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, USA, 27–30 June 2016; pp. 770–778.
14. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. In Proceedings of the 3rd International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2015.
15. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the Inception Architecture for Computer Vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826.
16. Lecun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [\[CrossRef\]](#) [\[PubMed\]](#)
17. Pawara, P.; Okafor, E.; Surinta, O.; Schomaker, L.; Wiering, M. Comparing local descriptors and bags of visual words to deep convolutional neural networks for plant recognition. In Proceedings of the ICPRAM 2017—6th International Conference on Pattern Recognition Applications and Methods, Porto, Portugal, 24–26 February 2017; pp. 479–486. [\[CrossRef\]](#)
18. Ur Rahman, H.; Ch, N.J.; Manzoor, S.; Najeeb, F.; Siddique, M.Y.; Khan, R.A. A comparative analysis of machine learning approaches for plant disease identification. *Adv. Life Sci.* **2017**, *4*, 120–126.
19. Konstantinos, P. Ferentinos Deep learning models for plant disease detection and diagnosis. *Comput. Electron. Agric.* **2018**, *145*, 311–318.
20. Torres, R.; Papa, J.P.; Da, R.; Torres, S.; Falcão, A.X. Content-based image retrieval: Theory and applications. *RITA* **2006**, *13*, 161–185.
21. Marwaha, S.; Chand, S.; Saha, A. Disease diagnosis in crops using content based image retrieval. In Proceedings of the 2012 12th International Conference on Intelligent Systems Design and Applications (ISDA), Kochi, India, 27–29 November 2012; pp. 729–733.
22. Patil, J.K.; Kumar, R. Comparative analysis of content based image retrieval using texture features for plant leaf diseases. *Int. J. Appl. Eng. Res.* **2016**, *11*, 6244–6249.
23. Baquero, D.; Molina, J.; Gil, R.; Bojacá, C.; Franco, H.; Gómez, F. An image retrieval system for tomato disease assessment. In Proceedings of the 2014 19th Symposium on Image, Signal Processing and Artificial Vision, STSIVA, Armenia, Colombia, 17–19 September 2014.
24. Yin, H.; Da Woon Jeong, Y.H.G.; Yoo, S.J.; Jeon, S.B. A Diagnosis and Prescription System to Automatically Diagnose Pests. In Proceedings of the Third International Conference on Computer Science, Computer Engineering, and Education Technologies (CSCEET2016), Lodz University of Technology, Lodz, Poland, 19–21 September 2016; p. 47.

25. Piao, Z.; Ahn, H.G.; Yoo, S.J.; Gu, Y.H.; Yin, H.; Jeong, D.W.; Jiang, Z.; Chung, W.H. Performance analysis of combined descriptors for similar crop disease image retrieval. *Cluster Comput.* **2017**, *20*, 3565–3577. [[CrossRef](#)]
26. Suh, B.; Ling, H.; Bederson, B.B.; Jacobs, D.W. Automatic thumbnail cropping and its effectiveness. In Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology, Vancouver, BC, Canada, 2–5 November 2003; pp. 95–104.
27. Chen, J.; Bai, G.; Liang, S.; Li, Z. Automatic image cropping: A computational complexity study. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 507–515.
28. Zhang, Z. Introduction to machine learning: K-nearest neighbors. *Ann. Transl. Med.* **2016**, *4*, 218. [[CrossRef](#)]
29. Koklu, M.; Ozkan, I.A. Multiclass classification of dry beans using computer vision and machine learning techniques. *Comput. Electron. Agric.* **2020**, *174*, 105507. [[CrossRef](#)]
30. Bray, J.R.; Curtis, J.T. An Ordination of the Upland Forest Communities of Southern Wisconsin. *Ecol. Monogr.* **1957**, *27*, 325–349. [[CrossRef](#)]
31. Elhassouny, A.; Smarandache, F. Smart mobile application to recognize tomato leaf diseases using Convolutional Neural Networks. In Proceedings of the 2019 International Conference of Computer Science and Renewable Energies (ICCSRE), Agadir, Morocco, 22–24 July 2019; pp. 10–13. [[CrossRef](#)]
32. Kathiresan, G.; Anirudh, M.; Nagharjun, M.; Karthik, R. Disease detection in rice leaves using transfer learning techniques. *J. Phys. Conf. Ser.* **2021**, *1911*, 012004. [[CrossRef](#)]
33. Sagar, A.; Jacob, D. On using transfer learning for plant disease detection. *bioRxiv* **2021**. [[CrossRef](#)]