



Article

# Deep Learning-Based Leaf Disease Detection in Crops Using Images for Agricultural Applications

Andrew J.<sup>1</sup>, Jennifer Eunice<sup>2</sup>, Daniela Elena Popescu<sup>3</sup>, M. Kalpana Chowdary<sup>4,\*</sup> and Jude Hemanth<sup>2</sup>

<sup>1</sup> Department of Computer Science and Engineering, Manipal Institute of Technology, Manipal Academy of Higher Education, Udupi 576104, Karnataka, India

<sup>2</sup> Department of ECE, Karunya Institute of Technology and Sciences, Coimbatore 641114, Tamil Nadu, India

<sup>3</sup> Faculty of Electrical Engineering and Information Technology, University of Oradea, 410087 Oradea, Romania

<sup>4</sup> Department of Computer Science and Engineering, MLR Institute of Technology, Dundigal 500043, Hyderabad, India

\* Correspondence: dr.kalpana@mlrinstitutions.ac.in

**Abstract:** The agricultural sector plays a key role in supplying quality food and makes the greatest contribution to growing economies and populations. Plant disease may cause significant losses in food production and eradicate diversity in species. Early diagnosis of plant diseases using accurate or automatic detection techniques can enhance the quality of food production and minimize economic losses. In recent years, deep learning has brought tremendous improvements in the recognition accuracy of image classification and object detection systems. Hence, in this paper, we utilized convolutional neural network (CNN)-based pre-trained models for efficient plant disease identification. We focused on fine tuning the hyperparameters of popular pre-trained models, such as DenseNet-121, ResNet-50, VGG-16, and Inception V4. The experiments were carried out using the popular PlantVillage dataset, which has 54,305 image samples of different plant disease species in 38 classes. The performance of the model was evaluated through classification accuracy, sensitivity, specificity, and F1 score. A comparative analysis was also performed with similar state-of-the-art studies. The experiments proved that DenseNet-121 achieved 99.81% higher classification accuracy, which was superior to state-of-the-art models.

**Keywords:** deep learning; transfer learning; CNN; leaf pathology; leaf disease



**Citation:** J., A.; Eunice, J.; Popescu, D.E.; Chowdary, M.K.; Hemanth, J. Deep Learning-Based Leaf Disease Detection in Crops Using Images for Agricultural Applications. *Agronomy* **2022**, *12*, 2395. <https://doi.org/10.3390/agronomy12102395>

Academic Editors: Ahmed Kayad and Ahmed Rady

Received: 30 August 2022

Accepted: 30 September 2022

Published: 3 October 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

Agriculture, being a substantial contributor to the world's economy, is the key source of food, income, and employment. In India, as in other low- and middle-income countries, where an enormous number of farmers exist, agriculture contributes 18% of the nation's income and boosts the employment rate to 53% [1]. For the past 3 years, the gross value added (GVA) by agriculture to the country's total economy has increased from 17.6% to 20.2% [2,3]. This sector provides the highest share of economic growth. Hence, the impact of plant disease and infections from pests on agriculture may affect the world's economy by reducing the production quality of food. Prophylactic treatments are not effective for the prevention of epidemics and endemics. Early monitoring and proper diagnosis of crop disease using a proper crop protection system may prevent losses in production quality.

Identifying types of plant disease is extremely important and is considered a crucial issue. Early diagnosis of plant disease may pave the way for better decision-making in managing agricultural production. Infected plants generally have obvious marks or spots on the stems, fruits, leaves, or flowers. Most specifically, each infection and pest condition leaves unique patterns that can be used to diagnose abnormalities. Identifying a plant disease requires expertise and manpower. Furthermore, manual examination when identifying the type of infection of plants is subjective and time-consuming, and, sometimes, the disease identified by farmers or experts may be misleading [4]. This may lead to the

usage of an unsuitable drug during the process of evaluating the plant disease, which may deteriorate the quality of the crops and end up polluting nature.

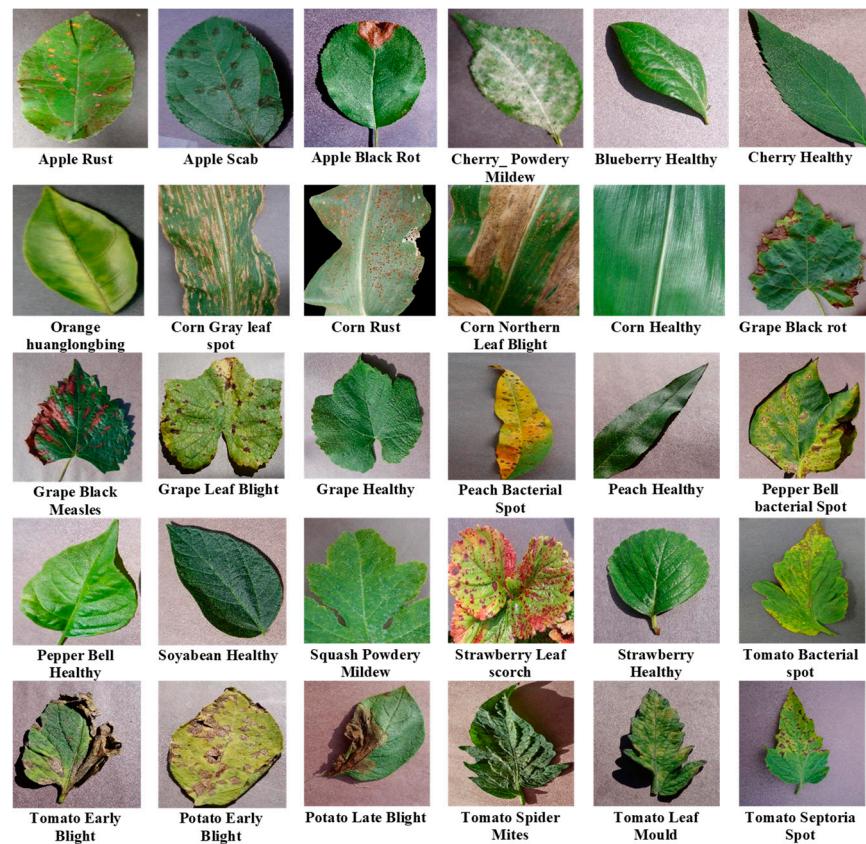
With the evolution of computer vision, there are numerous ways to resolve the detection issues for plants, since the infection spots are initially seen as spots and patterns on leaves [5]. Researchers have proposed several techniques to accurately detect and classify plant infections. Some use traditional image processing techniques that incorporate hand-crafted—that is, manual—feature extraction and segmentation [6]. Dubey et al. [7] proposed a K-means clustering algorithm to segment the infected portion of leaves, with the final classification achieved using a multi-class support vector machine (SVM). Yun et al. [8] used probabilistic neural network to extract meteorological and statistical features. The experiments were carried out in cucumber plants infected with cucumber downy mildew, anthracnose, and blights. Further, many models using traditional methods have been proposed for disease recognition in plants, such as in the work of Liu et al. [9], who used SVM and K-means clustering techniques, along with a backpropagation neural network. Although the image processing methods achieved promising results, the process involved in disease recognition is still tedious and time-consuming. Furthermore, the models rely on hand-crafted featuring techniques, classification, and spot segmentation. In the computer vision era, following the emergence of artificial intelligence, much research has utilized machine learning [10] and deep learning [11] models to achieve better recognition accuracy.

With the advent of machine learning and deep learning techniques, the progress made in plant disease recognition has been enormous and represents a massive breakthrough in research. This has made it easy for automatic classification and feature extraction to express the original characteristics of an image. Furthermore, the availability of datasets, GPU machines, and software supporting complex deep-learning architectures with lower complexity has made it feasible to switch from traditional methods to the deep-learning platform. In recent times, convolution neural networks (CNNs) have gained wide attention for their recognition and classification abilities, which work by extracting low-level complex features from images. Hence, CNNs are preferred for the replacement of traditional methods in automated plant disease recognition as they achieve better outcomes [12]. A CNN-based predictive model has been proposed by Sharma et al. [13] for classification and image processing in paddy plants. Further, Asritha et al. [14] used a CNN for disease detection in paddy fields. In general, researchers use four- to six-layer convolutional neural networks for the classification of different plant species. Mohanty et al. [15] also used a CNN with a transfer learning approach for the classification, recognition, and segmentation of different diseases in plants. Although many kinds of research have been carried out using CNNs and better outcomes have been reported, there is little diversity in the datasets used [16]. The best outcome is likely to be achieved by training the deep-learning model using a large dataset. Although very good outcomes have been attained in the previous studies, improvement in the diversity of the image databases is still required. The models trained with the existing datasets lack diversity in the data and backgrounds compared to realistic photographed materials obtained from real agricultural fields.

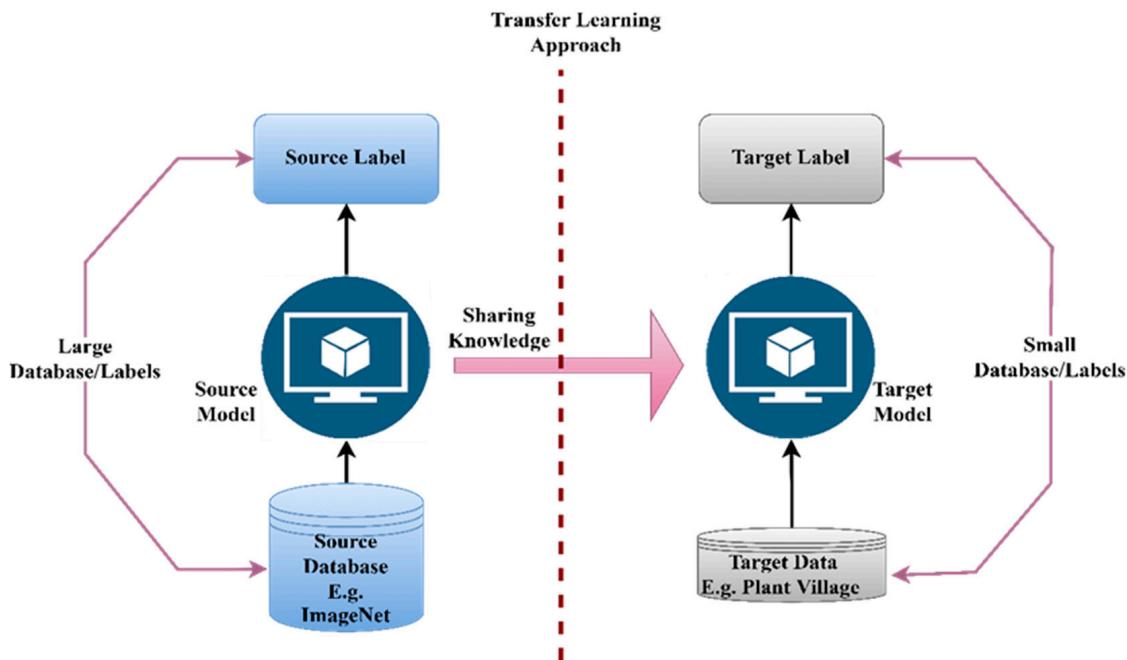
Pennsylvania State University published a plant disease dataset named PlantVillage [17]. PlantVillage consists of 54,305 RGB images in 38 plant disease classes. It contains the images of 14 different plants. Each plant has at least two classes of images showing healthy leaves and diseased leaves with dimensions of  $256 \times 256$ . Sample images from the dataset are shown in Figure 1. Since the release of this dataset, several plant disease identification studies have been carried out [18–21].

CNN deep-learning models are popular for image-based research. They are efficient in learning low-level complex features from images. However, deep CNN layers are difficult to train as this process is computationally expensive. To solve such issues, transfer learning-based models have been proposed by various researchers [22–26]. Popular transfer learning models include VGG-16, ResNet, DenseNet, and Inception [27]. These models are trained with the ImageNet dataset, which consists of multiple classes. Such models can be used for training with any dataset as the features of the images, such as edges and contours, are

common among the datasets. Hence, the transfer learning approach has been found to be the most suitable and robust model for image classification [28]. Further, transfer learning can improve learning even when there is a smaller dataset. Figure 2 shows the basic idea behind transfer learning.



**Figure 1.** Sample images from PlantVillage dataset for 38 types of leaf diseases.

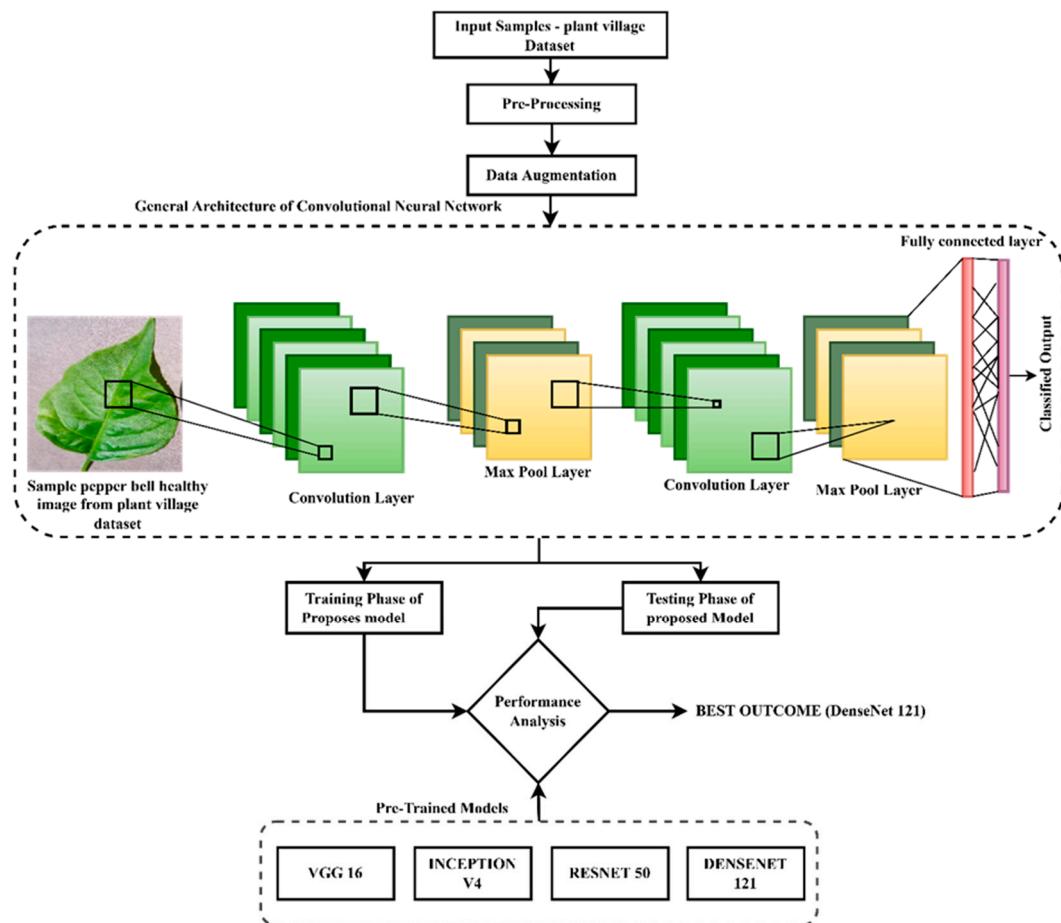


**Figure 2.** Basic idea behind transfer learning.

With transfer learning [22], tasks are more precise, as the model can be trained by freezing the last or the first layers. Thus, by freezing the layers, the model parameters can be retained and tuned for feature extraction and classification [29]. In this study, we performed a comparative performance analysis of different transfer learning models with deep CNNs in order to enhance recognition and classification accuracy and attenuate time complexity. Our workflow architecture is depicted in Figure 3. The experiments were carried out using the PlantVillage dataset with pre-trained CNN models, such as VGG-16, DenseNet-121, ResNet-50, and Inception V4. The major contributions of this manuscript can be summarized as follows:

- Development of a deep learning model for the diagnosis of various plant diseases;
- Determination of the best transfer learning technique to achieve the most accurate classification and optimal recognition accuracy for multi-class plant diseases;
- Resolution of distinct labeling and class issues in plant disease recognition by proposing a multi-class, multi-label transfer learning-based CNN model;
- Resolution of the overfitting problem through data augmentation techniques;

The rest of the article is arranged as follows. Section 2 provides a literature survey. The methodology used in this work is presented in Section 3. Section 4 discusses the various experiments conducted. The results and discussion are presented in Section 5. Finally, Section 6 concludes the paper with future directions.



**Figure 3.** Overall workflow diagram.

## 2. Related Work

In the field of agricultural production, ignoring the early signs of plant disease may lead to losses in food crops, which could eventually destroy the world's economy [30].

This section presents an in-depth survey of state-of-the-art research in the field of leaf disease identification.

A CNN-based deep learning model was proposed for the accurate classification of plant disease in [31], and the model was trained using a publicly available dataset with 87,000 RGB images. Initially, preprocessing was undertaken, followed by segmentation. For classification, a CNN was used. Although this model attained a recognition accuracy of 93.5%, it failed to classify some classes, leading to confusion with the classes in subsequent stages. Further, the performance of the model deteriorated due to limited availability of data. However to improve recognition accuracy, Narayanan et al. [32] proposed a hybrid convolutional neural network to classify banana plant disease. In their approach, the raw input image was preprocessed without altering any default information, and the standard image dimensions were maintained using a median filter. This approach used a fusion SVM along with a CNN. A multiclass SVM was used in the testing phase to identify the type of infection or disease in infected banana leaves, whereas the SVM was used in phase 1 to classify whether the banana leaves were healthy or infected. The classified CNN output was fetched as an input to the support vector machine, attaining a classification accuracy of 99%. The previous work stated that the CNN had better accuracy outcomes than traditional methods but this approach lacked diversity.

Jadhav et al. [33] proposed a CNN for the identification of plant disease. In this approach, they used pre-trained CNN models to identify diseases in soybean plants. The experiments were carried out using pre-trained transfer learning approaches, such as AlexNet and GoogleNet, and attained better outcomes, but the model fell behind in the diversity of classification. Many existing models focus on identifying single classes of plant disease rather than building a model to classify various plant diseases. This is mainly due to the limited databases for training deep learning models with diversified plant species.

Jadhav et al. [34] were the first to propose a novel histogram transformation approach, which enhanced the recognition accuracy of deep learning models by generating synthetic image samples from low-quality test set images. The motive behind this work was to enhance the images in the cassava leaf disease dataset using Gaussian blurring, motion blurring, resolution down-sampling, and over-exposure with a modified MobileNetV2 neural network model. In their approach, synthetic images using modified color value distributions were generated to address the data shortage that a data-hungry deep-learning model faces during its training phase and achieve better outcomes.

Following Olusola et al., Abbas et al. [35], in their work proposed, a conditional generative adversarial network to generate a database of synthetic images of tomato plant leaves. With the advent of generative networks, previously expensive, time-consuming and laborious real-time data acquisition or data collection have become possible. Anh et al. [36] proposed a benchmark dataset-based multi-leaf classification model using a pre-trained MobileNet CNN model and found it efficient in classification, attaining a reliable accuracy of 96.58%. Further, a multi-label CNN was put forward in [20] for the classification of multiple plant diseases using transfer learning approaches, such as DenseNet, Inception, Xception, ResNet, VGG, and MobileNet, and the authors claim that theirs' is the first research work that classifies 28 classes of plant disease using a multi-label CNN. Classification of plant diseases using the Ensemble Classifier was proposed in [37]. The best ensemble classifier was evaluated with two datasets; namely, PlantVillage and Taiwan Tomato Leaves. Pradeep et al. [21] proposed the EfficientNet model using a convolutional neural network for multi-label and multi-class classification. The secret layer network in the CNN had a better impact on the identification of plant diseases. However, the model underperformed when validated with benchmark datasets. An effective, loss-fused, resilient convolutional neural network (CNN) was proposed in [38] using the publicly available benchmark dataset PlantVillage and achieved a classification accuracy of 98.93%. Though this method improved the classification accuracy, the model lagged in its performance when using real-time images under different environmental conditions. Later, Enkvetchakul and Surinta [39] proposed a CNN network with a transfer learning approach for two plant

diseases. NASMobileNet and MobileNetV2 were the two pre-trained network models used for the classification of plant diseases, among which the most accurate prediction outcome was that based on the NASMobileNet algorithm. Overfitting in deep learning can be resolved using the data augmentation approach. The data augmentation technique was implemented in an experimental setup that included cut-out, rotation, zoom, shift, brightness, and mix-up. Leaf disease datasets and iCassava 2019 were the two kinds of dataset used. The maximum test accuracy attained after the evaluation was 84.51%. Table 1 shows the different convolutional neural network models that have been proposed to improve accuracy.

**Table 1.** Detailed summary of the CNN models used in the recognition and classification of plant disease.

Reference	Crop Focus	Disease Addressed	Dataset	Classes	Model	Model Performance
[29]	Several	Citrus canker, black mould, bacterial blight, etc.	Plant disease symptoms database	12 56 diseases under 12 classes	CNN GoogLeNet with tenfold cross-validation	Accuracy: 84%
[40]	Several	Black rot, late blight, early blight	Self-collected database	527 species of diseases under 5 classes	CNN	Accuracy: 96.5%
[41]	Tomato plant	Various diseases and pests in tomato plant	Self-generated database	9	Faster Region-based CNN with SSD <sup>1</sup> and Region-based Fully Convolutional Network	Precision: 85.98%
[42]	Several	Powdery mildew, early and late blights, cucumber mosaic, downy mildew, etc.	Open dataset	58	CNN with pre-trained VGG network	Accuracy: 99.53%
[27]	Several	Black rot, late blight, early blight	PlantVillage	38	VGG-16, Inception V4, ResNet with 50, 101, and 152 layers, and DenseNet with 121 layers	Accuracy: 99.75%
[43]	Several	Pepper bell bacterial spot, tomato early and late blight	PlantVillage	38	Pre-trained with ImageNet, GoogLeNet, and VGG-16 models	Accuracy: 99.09%
[44]	Apple	Apple scab, apple grey spot, general and serious cedar apple rust, serious apple scab	AI-Challenger plant disease recognition	6	DenseNet-121	Accuracy: 93.71%
[45]	Tomato	ToMV, leaf mould fungus, powdery mildew, blight	AI-Challenger plant disease recognition	4	Faster regional CNN	Accuracy: 98.54%
[46]	Several	Rice leaf smut, maize common rust, maize eyespot, rice bacterial leaf streak	Public database	7	Pre-trained models	Accuracy: 92%
[47,48]	Rice plant	Sheath blight, rice blast, bacterial blight	Self-generated database	4	Pre-trained CNN with SVM classifier	Accuracy: 91.37%

<sup>1</sup> Single shot detector.

### 3. Methodology

CNN models are best suited for object recognition and classification with image databases. Despite the advantages of CNNs, challenges still exist, such as the long duration

of training and the requirement for large datasets. To extract the low-level and complex features from the images, deep CNN models are required; this increases the complexity of the model training. Transfer learning approaches are capable of addressing the aforementioned challenges. Transfer learning uses pre-trained networks, in which model parameters learned on a particular dataset can be used for other problems. In this section, we discuss the methodologies used in this work.

### 3.1. Multi-Class Classification

Plant disease datasets hold multiple images infected and healthy plant samples, with each sample mapped to a particular class. For instance, if we consider the banana plant as a class, then all the images of healthy and infected samples of banana plants will be mapped to that specific class. Now, the classification of the target image is purely based on the features extracted from the source image. Considering the same example of the banana plant, the banana class has four sets of diseases; namely, xanthomonas wilt, fusarium wilt, bunchy top virus, and black sigatoka [32]. When a sample of one particular disease is fetched as input after training with all four sets of disease samples under the banana class, the testing phase output will classify the exact label of the disease from among the four categories mapped under that particular class. Thus, multi-class classification is mutually exclusive, whereas, in multi-label classification, each category inside a class is itself considered a different class. Suppose we have N classes, then we can refer to N multi-classes, and if the N classes have M categories, then each category inside each of the N classes is itself considered a class.

### 3.2. Transfer Learning Approach

In general, it takes several days or weeks to train and tune most state-of-art models, even if the model is trained on high-end GPU machines. Training and building a model from scratch is time-consuming. A CNN model built from scratch with a publicly available plant disease dataset seemed to attain 25% accuracy in 200 epochs, whereas using a pre-trained CNN model using a transfer learning approach attained 63% accuracy in almost half the number of iterations (over 100 epochs). Transfer learning methods include several approaches, the choice of which depends on the choice of the pre-trained network model for classification and the particular nature of the dataset.

### 3.3. ResNe-50

ResNet-50 is a convolutional neural network that has 50 deep layers. The model has five stages, with convolution and identity blocks. These residual networks act as a backbone for computer vision tasks. ResNet [49] introduced the concept of stacking convolution layers one above the other. Besides stacking the convolution layers, they also have several skip connections, which bypass the original input to reach the output of the convolutional neural network. Furthermore, the skip connection can be placed before the activation function to mitigate the vanishing gradient issue. Thus, deeper models end up with more errors, and to resolve these issues, skip connections in the residual neural network were introduced. These shortcut connections are simply based on identity mapping.

Let us consider  $x$  as the input image,  $F(x)$  as the nonlinear layers fitting mappings, and  $H(x)$  as the residual mapping. Thus, the function for residual mapping becomes:

$$H(x) = F(x) + x \quad (1)$$

ResNet-50 has convolution as an identity block. Each identity block has three convolutional layers and over 23 M trainable parameters. Input  $x$  and shortcut  $x$  are the two matrices, and they can only be added if the output dimension from a shortcut and the convolution layer after the convolution and batch normalization are the same. Otherwise, shortcut  $x$  must go through a convolution layer and batch normalization to match the dimension.

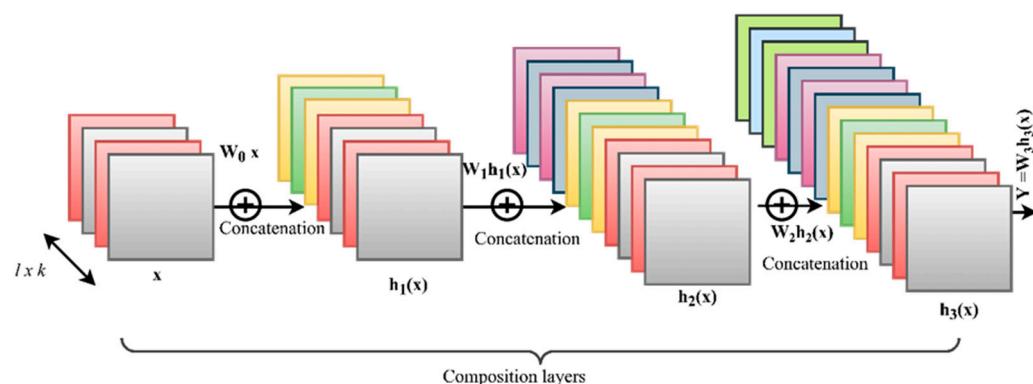
### 3.4. VGG-16

The VGG-16 [50] network model, also known as the Very Deep Convolutional Network for Large-Scale Image Recognition, was built by the Visual Geometry Group from Oxford University. The depth is pushed to 16–19 weight layers and 138 M trainable parameters. The depth of the model is also expanded by reducing the convolution filter size to  $3 \times 3$ . This model requires more training time and occupies more disk space.

### 3.5. DenseNet-121

DenseNet-121 [51] is a deep CNN model designed for image classification using dense layers with shorter connections between them. In this network, each layer receives additional inputs from its preceding layers and passes its generated feature maps to the succeeding layer. Concatenation is performed between each layer, through which the next successive layer receives collective knowledge from all the preceding layers. Further, the network is thin and small since the preceding layers' feature maps are mapped to the subsequent layers. In this manner, the number of channels in a dense block is reduced, and the growth rate of a channel is denoted by  $k$ . Figure 4 shows the working principle of a dense block in DenseNet. For each composition layer, regularization, activation, and convolution operations are carried out for the output feature maps of  $k$  channels. Batch normalization, ReLu activation and convolution, and pooling are performed to transform the outcome of subsequent layers:

$$Y = W_3\{x, h_1(x), h_2(x), h_3(x)\} \quad (2)$$



**Figure 4.** Working principle of a dense block.

The layers have a strong gradient flow and more diversified features. DenseNet is small compared to ResNet. Further, the classifiers in the standard ConvNet model process complex features, whereas DenseNet uses all features, even with different complexities, and provides smooth decision boundaries.

### 3.6. Inception V4

Images contain lots of details and salient features and may vary in size. With these variations in size, choosing the right filter size for feature extraction is challenging. For local information extraction, a smaller kernel size should be chosen, whereas, for global information, the kernel size should be large. Stacking up the convolution layers may result in overfitting and vanishing gradient problems. To solve this, the Inception modules incorporate different kernel sizes in each block, such that the network model becomes wider instead of deeper [52]. For instance, the naïve Inception module can use  $3 \times 3$ ,  $1 \times 1$ , or  $5 \times 5$  sizes for the filter after three different stages of convolution. Max-pooling is then performed and the outcome is concatenated and passed to the next layer. The stem of the Inception layer is meant for setting up an initial set of operations to be performed before the Inception module. Further, Inception V4 has reduction blocks to alter the height and width of the grids.

## 4. Experiments

The baseline system for evaluation of our experiments was a GPU NVIDIA GeForce GTX workstation. The operating environment was Windows 10, GDDR5 graphic memory type, Core i5 9th generation, 8 GB RAM. Software implementation was undertaken using the Anaconda3, Keras, OpenCV, Numpy CuDNN, and Theano libraries. CUDNN and CuMeM are simple libraries specially designed to carry out deep learning implementations with less memory and faster execution. Both these libraries were designed by NVIDIA to work in the Theano backend. OpenCV supports both academic and commercial project development and supports Linux, Windows, Mac OS, iOS, Python, Java, and Android interfaces. In this work, for each experiment, the training accuracy and the testing accuracy were evaluated. The losses obtained during the testing and training phases were calculated for each model. The models were trained using the PlantVillage dataset with the aim of accelerating the learning speed of the CNN with transfer learning models. The pre-trained models chosen for our study included ResNet-50, Inception V4, VGG-16, and DenseNet-121, which had been previously trained using the ImageNet dataset with 1.2 M images and 1000 image categories.

### 4.1. Description of Dataset

The PlantVillage [17] dataset is a publicly available dataset with different categories of plant diseases. This dataset comprises 38 classes with 54,305 images. For our experimental analysis, we split the dataset into training samples, testing samples, and validation samples. The pre-trained models were trained with 80% of the PlantVillage dataset, and 20% was used for validation and testing. Further, the total number of samples available for the plant classes was 54,305, out of which 43,955 samples were used for training, 4902 for validation, and 5488 for testing. All these train, test and validation sets include all the 38 classes of the different plant diseases. The details of the dataset split are presented in Table 2.

**Table 2.** Details of PlantVillage dataset split for training, validation, and testing.

Plant Type	Diseases Classes	Total Samples	Training Samples	Test Samples	Validation Samples
Apple	Apple_scab	573	510	63	57
	Apple_black_rot	565	502	63	56
	Apple_cedar_apple_rust	250	222	28	25
	Apple_healthy	1497	1332	165	148
Blueberry	Blueberry_healthy	1366	1215	151	136
Cherry	Cherry_powdery_mildew	957	851	106	95
	Cherry_healthy	777	691	86	77
Corn	Corn_gray_leaf_spot	466	414	52	47
	Corn_common_rust	1084	964	120	108
	Corn_northern_leaf_blight	896	797	99	89
	Corn_healthy	1057	940	117	105
Grape	Grape_black_rot	1073	955	118	107
	Grape_black_measles	1258	1119	139	125
	Grape_leaf_blight	979	871	108	97
	Grape_healthy	385	342	43	38
Orange	Orange_haunglongbing	5011	4460	551	496
Peach	Peach_bacterial_spot	2090	1860	230	207
	Peach_healthy	327	291	36	33
Pepper	Pepper_bell_bacterial_spot	997	807	100	90
	Pepper_Bell_healthy	1478	1197	148	133

**Table 2.** Cont.

Plant Type	Diseases Classes	Total Samples	Training Samples	Test Samples	Validation Samples
Potato	Potato_early_blight	1000	810	100	90
	Potato_healthy	1000	810	100	90
	Potato_late_blight	152	122	16	14
Raspberry	Raspberry_healthy	664	299	38	34
Soybean	Soybean_healthy	5295	4122	509	459
Squash	Squash_powdery_mildew	1669	1485	184	166
Strawberry	Strawberry_healthy	1009	898	111	100
	Strawberry_leaf_scorch	415	369	46	41
Tomato	Tomato_bacterial_spot	2127	1722	213	192
	Tomato_early_blight	1000	810	100	90
	Tomato_healthy	1591	1546	191	172
	Tomato_late_blight	1909	770	96	86
	Tomato_leaf_mold	952	1433	178	160
	Tomato_septoria_leaf_spot	1771	1357	168	151
	Tomato_spider_mites_two-spotted_spider_mite	1676	1136	141	127
	Tomato_target_spot	1404	4338	536	483
	Tomato_mosaic_virus	373	301	38	34
	Tomato_yellow_leaf_curl_virus	3209	1287	160	144
Total		54,305	43,955	5448	4902

#### 4.2. Preprocessing and Data Augmentation

The dataset held 38 classes with 26 diseases and 14 species of crops. For our experimental purpose, we used the colour images from the PlantVillage dataset, as they fit well with the transfer learning models. The images were downsampled to  $256 \times 256$  pixels as a standardized format since we used different pre-trained network models that require different input sizes. For VGG-16, DenseNet-121 and ResNet-50, the input size is  $224 \times 224 \times 3$  (height, width, and channel width), whereas, for Inception V4, the input shape of images is  $299 \times 299 \times 3$  (height, width, and channel width). Though the dataset is huge, with around 54,000 images of different crop diseases, the images match the real-life images captured by farmers using different image acquisition techniques, such as Kinect sensors, high-definition cameras, and smart phones. Further, a dataset of such a size is prone to overfitting. Therefore, to overcome this, overfitting regularization techniques, such as data augmentation after preprocessing, were introduced. The augmentation processes used with the preprocessed images included clockwise and anticlockwise rotation, horizontal and vertical flipping, zoom intensity, and rescaling. The images were not duplicated but augmented during the training process, so the physical copies of the augmented images were not stored but were temporarily used in the process. This augmentation technique not only prevents the model from overfitting and model loss but also increases the robustness of the model so that, when the model is used to classify real-life plant disease images, it can classify them with better accuracy.

#### 4.3. Fine-Tuning of Hyperparameters in Pre-Trained Models

The advantages of the transfer learning model are that it learns faster compared to models built from the scratch and that layers of the model can be frozen and the last layers trained for more accurate classification. Initially, certain standardizations of the hyperparameters for different pre-trained models were performed. The details of the hyperparameter tuning are listed in Table 3.

**Table 3.** Hyperparameter specifications.

Hyperparameters	Epochs
Dropout	0.5
Epochs	30
Activation	ReLU
Regularization	Batch normalization
Optimizer	Stochastic gradient descent (SGD)
Learning rate	0.001
Output classes	38

The models were optimized using stochastic gradient descent. The initial learning rates of the DenseNet-121, ResNet-50, VGG-16, and Inception V4 models were set to 0.001. Each model was run for 30 epochs and the dropout value was fixed as 0.5. In our experiment, the output graph started to converge after a few iterations (i.e., from 30 epochs the graph started to converge); thus, our experiment overcame overfitting and degradation issues.

#### 4.4. Network Architecture Model

The pre-trained network models where chosen based on their applicability for the plant disease classification task. The details of the model architecture are given in Table 4. Each network has different filter sizes for extracting specific features from feature maps. Filters play a key role in feature extraction. Further, each filter, when convolved with the input, will extract different features from it, and the specific feature extraction from the feature maps depends on the specific values of the filters. In our experiments, we used the actual pre-trained network models with the actual combinations of convolution layers and actual filter sizes used for each network model.

**Table 4.** Pre-trained network architecture model.

Network Model	VGG-16	Inception V4	ResNet-50	DenseNet-121
Total layers	16	22	50	121
Max pool layers	5	5	1	4
Dense layers	3	-	3	4
Drop-out layers	2	-	2	-
Flatten layers	1	-	1	-
Filter size	$3 \times 3$	$1 \times 1, 3 \times 3, 5 \times 5$	$3 \times 3$	$3 \times 3, 1 \times 1$
Stride	$2 \times 2$	$2 \times 2$	$2 \times 2$	$2 \times 2$
Trainable parameters	41.2 M	119.6 M	23.6 M	7.05 M

##### 4.4.1. VGG-16 Tuning Details

The input image dimensions for the network are  $224 \times 224 \times 3$ , and it has 64 channels in the first two layers with a filter size of  $3 \times 3$  and stride of 2. The next two layers in the VGG-16 have 256 channels with  $3 \times 3$  filters; followed by this is a max-pooling layer with stride of 2. After the pooling layer, there are two convolution layers with 256 channels with a  $3 \times 3$  filter size. Following the two convolution layers, there are two sets of three convolution layers, along with a pooling layer, with  $3 \times 3$  filters. The network includes one flatten layer, five max pool layers, and two dense layers.

##### 4.4.2. Inception V4 Tuning Details

The Inception V4 block has two phases: one is for feature extraction and the other uses fully connected layers. Inception V4 includes a stem block and the Inception A, B, and C blocks, which are followed by the reduction blocks A and B and an auxiliary classifier block.

#### 4.4.3. ResNet-50 Tuning Details

This residual CNN network has 50 layers, and the first layer is a convolutional layer with kernel size  $7 \times 7$ , a stride of 2, and 64 channels. The next three stages are convolution layers with filter sizes of  $1 \times 1$ ,  $3 \times 3$ , and  $1 \times 1$  and 64, 64, 256 channels. These are repeated three times. Similarly, the next convolution layers are repeated four times and the subsequent convolutional blocks are repeated six times.

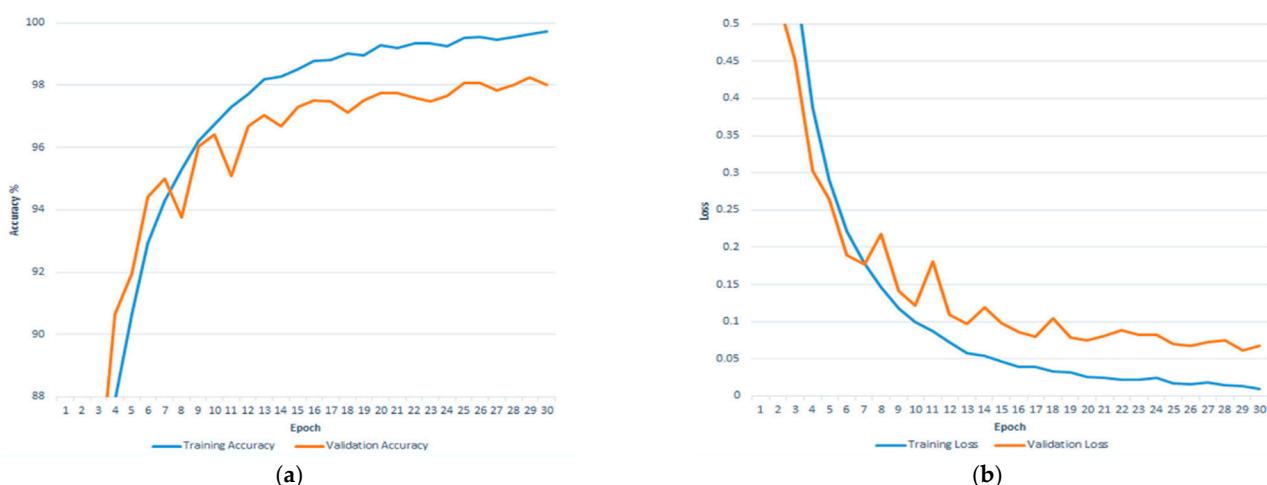
#### 4.4.4. DenseNet-121 Tuning Details

DenseNet-121 increases the depth of the convolutional neural network by solving the vanishing gradient issues. It has four dense blocks. In the first dense block, convolution is performed with  $1 \times 1$  and  $3 \times 3$  filter sizes, and this is repeated six times. Similarly, in the second dense block, convolution is performed using the filter sizes  $3 \times 3$  and  $1 \times 1$  and the steps are repeated 12 times. In the third dense block, convolution operations with the same filter size are repeated 24 times, and in the fourth dense block, the steps are repeated 16 times. In between the dense blocks are transition blocks with convolution and pooling layers.

## 5. Results and Discussion

This part of the study employed state-of-the art deep learning models using the transfer learning approach for the diagnosis of plant diseases. PlantVillage, a publicly available dataset, was used to further train the pre-trained deep CNN networks, which were previously trained with the ImageNet dataset. For our experiment, each model was standardized with a learning rate of 0.01, a dropout of 0.5, and 38 output classes.

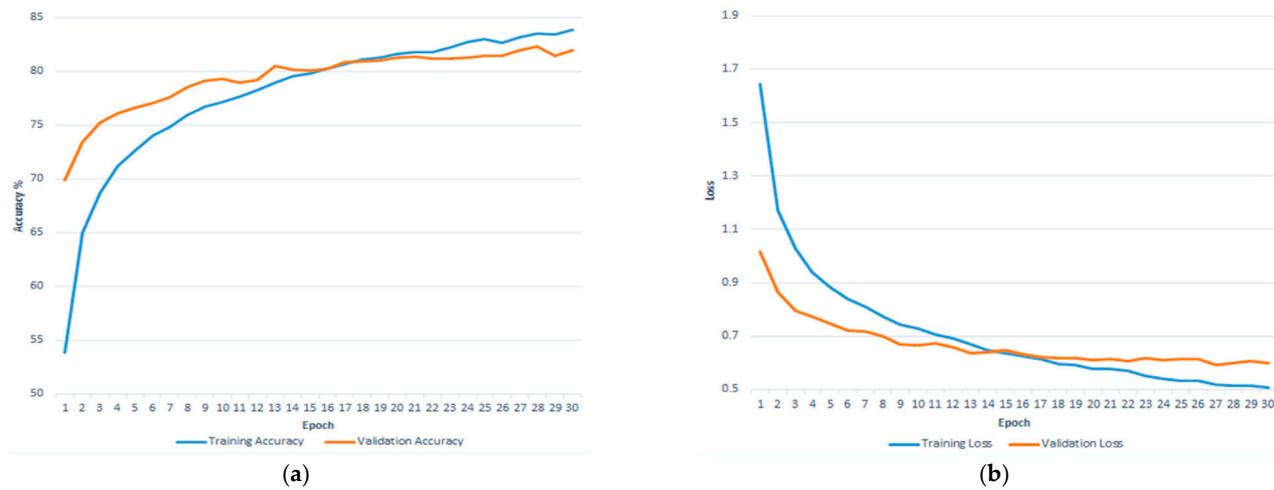
The dataset was split into training, test, and validation samples. A total of 80% of the samples from PlantVillage were used for training the pre-trained Inception V4, VGG-16, ResNet, and DenseNet-121 models. Each model was run for 30 epochs and it was found that our model started to converge after 10 epochs with high accuracy. The graph in Figure 5a depicts the recognition accuracy of the Inception V4 model. The training accuracy achieved using the inception V4 model was 99.78, and Figure 5b shows the log loss of the Inception V4 model.



**Figure 5.** Performance analysis of Inception V4 model using PlantVillage dataset. (a) Model recognition accuracy; (b) train and test loss.

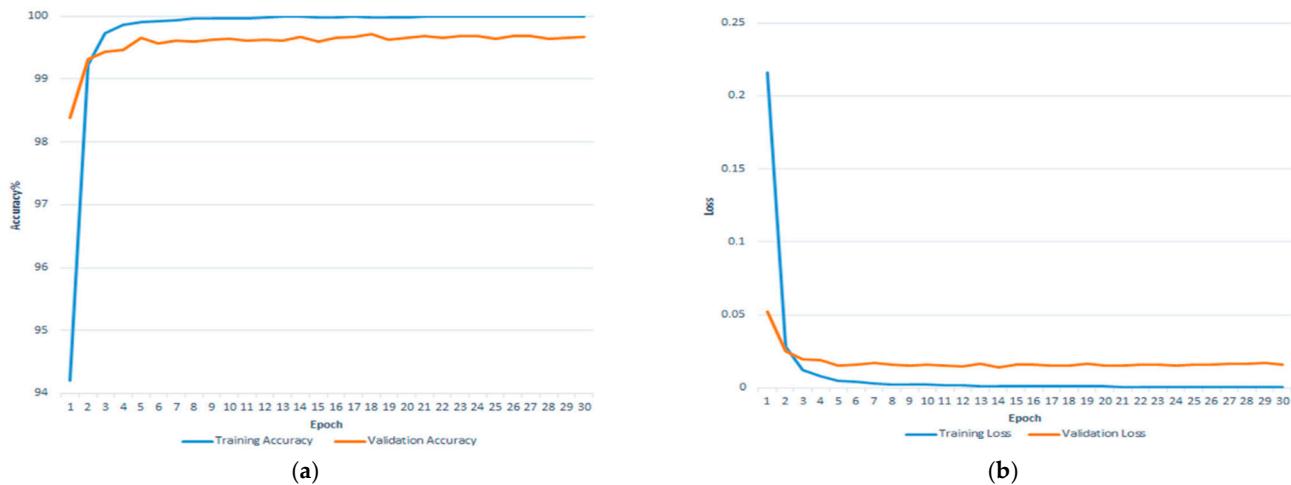
The second experiment evaluated the VGG-16 model using the same dataset. After standardization of the hyperparameters, the model was trained with 80% of the same dataset, with 10% used for testing and the remaining 10% of the image samples used for testing and validation. It can be observed from Figure 6a that the model recognition accuracy reached around 78% in the initial 10 epochs, after which is steadily increased to attain the maximum recognition accuracy of 84.27%, which was lower than the Inception

V4 model. The training loss and the validation model were found to be 0.52% and 0.64%, respectively, as seen in Figure 6b.



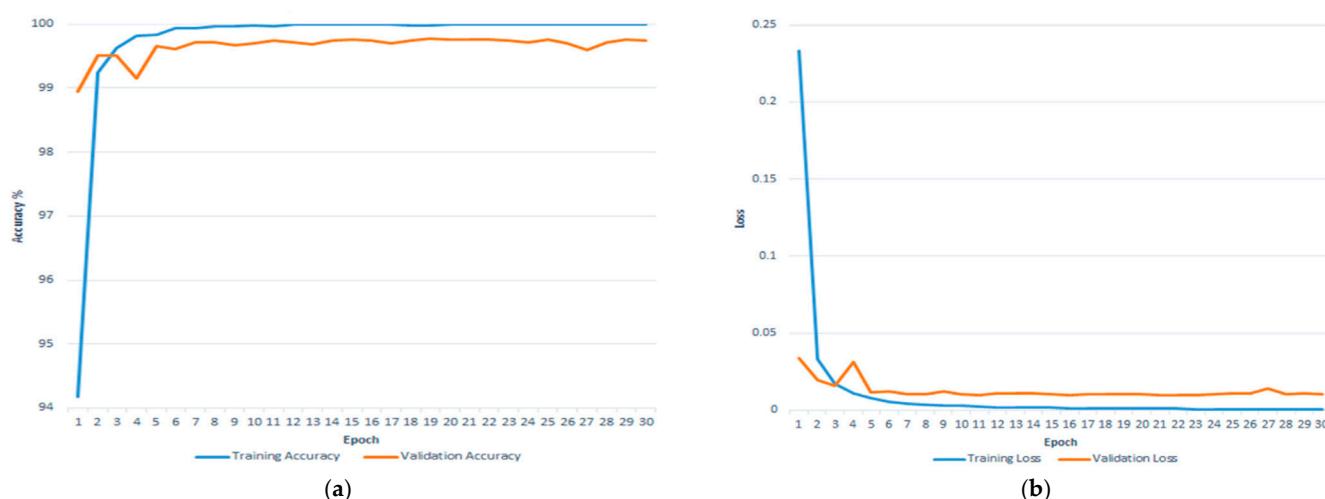
**Figure 6.** Recognition accuracy of VGG-16. (a) Training and testing accuracy; (b) training and validation loss in VGG-16 using PlantVillage dataset.

The third experiment was undertaken with the ResNet-50 model. The same method was applied in the evaluation of model loss and recognition accuracy, and the graphs for recognition accuracy and validation and training loss are plotted in Figure 7a,b. This model achieved an accuracy of 99.83 and a model loss of 0.027. It outperformed the Inception V4 and VGG-16 models.



**Figure 7.** Recognition accuracy of ResNet-50. (a) Training and testing accuracy; (b) training and validation loss in ResNet-50 using PlantVillage dataset.

After hyperparameter standardization, the final experiment was executed with DenseNet-121, which has 121 layers with four dense blocks and a transition layer between each dense block. Figure 8a,b show the graphs plotted for the training and validation accuracy/loss for 30 epochs. In the testing phase after training, the maximum accuracy achieved was 99.81% and the maximum validation loss calculated was 0.0154%. A comparative performance analysis is shown in Table 5 for the pre-trained network model experiments.



**Figure 8.** Recognition accuracy of DenseNet-121. (a) Training and testing accuracy; (b) training and validation loss in DenseNet-121 using PlantVillage dataset.

**Table 5.** Comparative performance analysis of various network models.

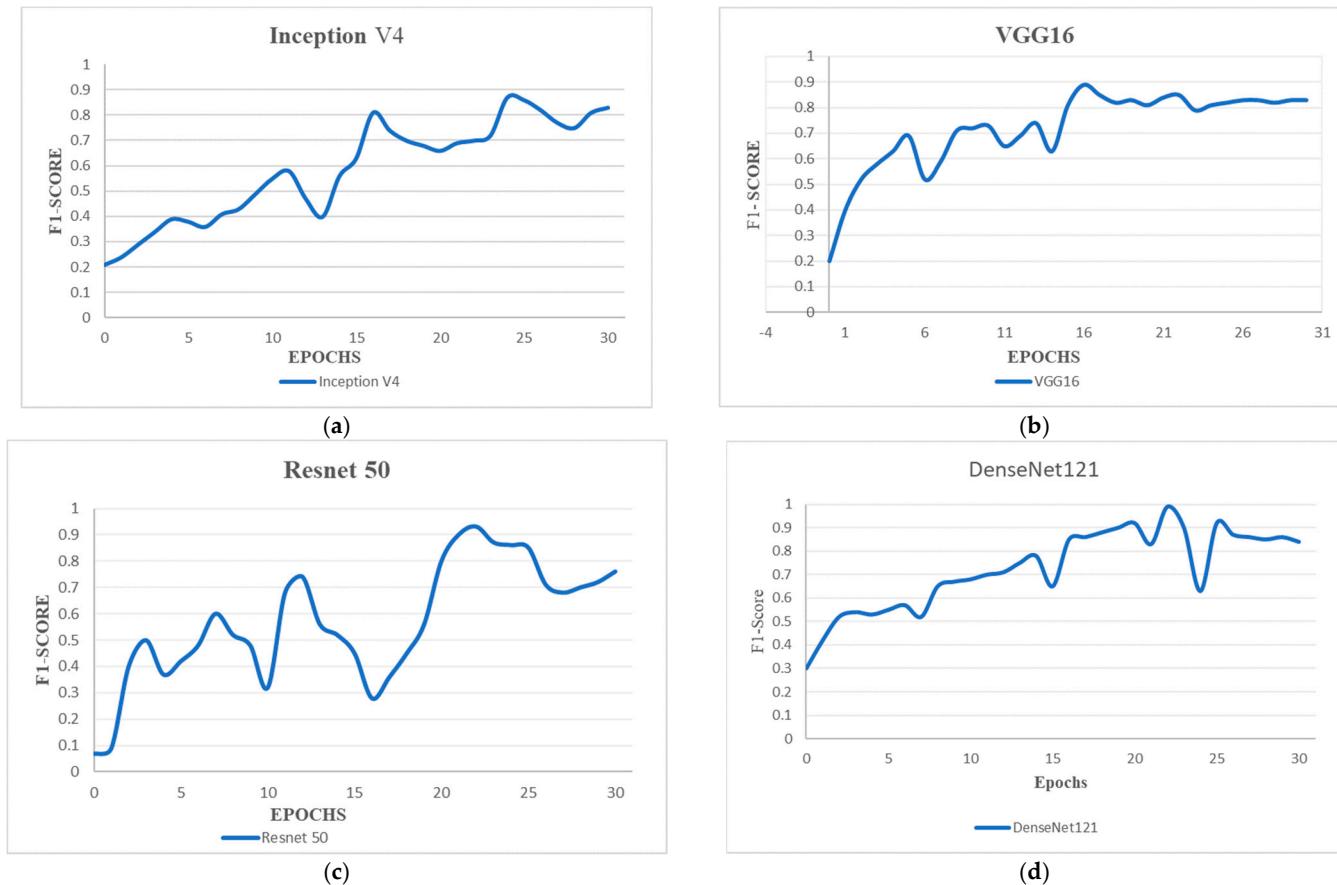
Network Models	Training Accuracy (%)	Training Loss (%)	Test Accuracy (%)	Test Loss (%)
Inception V4	99.78	0.01	97.59	0.0586
VGG-16	84.27	0.52	82.75	0.64
ResNet-50	99.82	6.12	98.73	0.027
DenseNet-121	<b>99.87</b>	<b>0.016</b>	<b>99.81</b>	<b>0.0154</b>

In agricultural production, early diagnosis of crop disease is essential for high yields. To maintain a high production rate, the latest technologies should be implemented in the early diagnosis of plant disease. It was observed from the literature study that deep learning models are efficient in image classification, and transfer learning based models are efficient in eliminating training complexity and huge dataset requirements. Hence, in this work, we evaluated four pre-trained models—VGG-16, ResNet-50, Inception V4, and DenseNet-121—to determine the model that was best capable of classifying various plant diseases. The results for the pre-trained models were evaluated with evaluation metrics, such as specificity, sensitivity, and F1 score values. The validation accuracy in terms of the F1 score was calculated and a graphical representation the validation accuracy for the pre-trained models is depicted in Figure 9. It was inferred that DenseNet-121 (Figure 9d) outperformed the other network models (Figure 9a–c) and attained the highest validation peak with 0.998, which is very close to an F1 score of 1. In general, the value of an F1 score ranges from 0 to 1. A model's performance is relatively better when it is closer to 1. In our analysis, after repeating the same experiments for all the pre-trained models, we found that the highest validation accuracy in terms of the F1 score was achieved by DenseNet-121 at 0.998, whereas it was 0.887 for Inception V4, 0.901 for VGG-16, and 0.935 for ResNet-50.

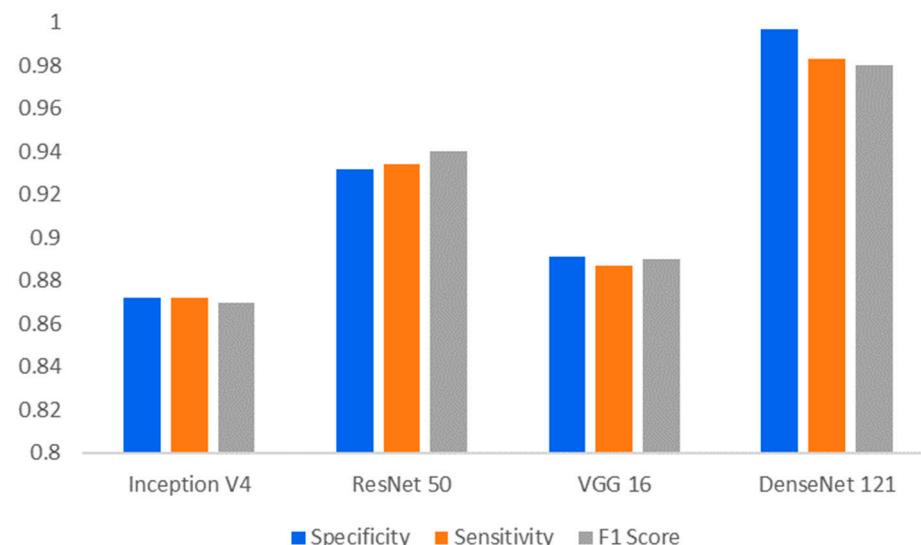
A statistical representation of the pre-trained network models based on the evaluation metrics is shown in Figure 10. The vanishing gradient issues resulting from skip connections were eliminated using regularization techniques, such as batch normalization. With deeper models, various challenges, such as overfitting, covariant shifts, and training time complexity, occurred. To overcome these challenges in our experiments, we fine-tuned the hyperparameters. The experiments used sensitivity to predict the proportion of actually healthy plants classed as healthy (true positive) and actually healthy plants classed as unhealthy (false negative). From the evaluation, it was observed that ResNet-50 and DenseNet-121 performed better than the VGG-16 and Inception V4 models. A performance

analysis of the different pre-trained models based on the specificity, sensitivity, and F1 score is shown in Figure 10.

$$\text{sensitivity (recall)} = \frac{\text{True Positive}}{(\text{True positive} + \text{False Negative})} \quad (3)$$



**Figure 9.** F1 score vs. validation scores using PlantVillage dataset for (a) Inception V4; (b) VGG-16; (c) ResNet-50; (d) DenseNet-121.



**Figure 10.** Performance analysis of pre-trained models based on different evaluation metrics.

Specificity is a measure of the proportion of actually unhealthy plants predicted to be unhealthy (true negative) and the actually unhealthy leaves predicted to be healthy (false positive)

$$\text{specificity} = \frac{\text{True Negative}}{(\text{True Negative} + \text{False positive})} \quad (4)$$

Table 6 presents a comparison of the obtained results with those from state-of-the-art studies from the literature that used transfer learning models. We considered state-of-the-art studies from the literature that experimented on the PlantVillage dataset. It was observed from the analysis that our work considered more plant disease classes. Further, our fine-tuned, pre-trained model achieved the best accuracy of 99.81%.

**Table 6.** Comparison with state-of-the-art transfer learning models.

References	Dataset Used	Pre-Trained Model	Multi-Classes	Recognition Accuracy (%)
[53]	PlantVillage	VGG-16	10	91.2
[54]	PlantVillage	ResNet-50	6	97.1
[55]	PlantVillage	AlexNet	7	98.8
<b>Our Work</b>	PlantVillage	Inception V4	38	97.59
		VGG-16	38	82.75
		ResNet-50	38	98.73
		<b>DenseNet-121</b>	<b>38</b>	<b>99.81</b>

## 6. Conclusions

In this work, we successfully analysed the different transfer learning models suitable for the accurate classification of 38 different classes of plant disease. Standardization and evaluation of state-of-the-art convolutional neural networks using transfer learning techniques were undertaken based on the classification accuracy, sensitivity, specificity, and F1 score. From the performance analysis of the various pre-trained architectures, it was found that DenseNet-121 outperformed ResNet-50, VGG-16, and Inception V4. Training the DenseNet-121 model seemed to be easy, as it had a smaller number of trainable parameters with reduced computational complexity. Hence, DenseNet-121 is more suitable for plant disease identification when there is a new plant disease that needs to be included in the model, demonstrating reduced training complexity. The proposed model achieved a classification accuracy of 99.81% and F1 score of 99.8%.

In future work, we will address the problems in real-time data collection and develop a multi-object deep learning model that can even detect plant diseases from a bunch of leaves rather than a single leaf. Furthermore, we are working towards implementing a mobile application with the trained model from this work. It will help farmers and the agricultural sector in real-time leaf disease identification.

**Author Contributions:** Conceptualization, J.E. and J.H.; methodology, A.J. and J.E.; validation, J.H., J.E., D.E.P. and A.J.; formal analysis, J.H., J.E., M.K.C. and A.J.; investigation, J.H., J.E., D.E.P., M.K.C. and A.J.; data curation, J.E. and A.J.; writing—original draft preparation, J.E., A.J. and J.H.; writing—review and editing, J.H., J.E., D.E.P., M.K.C. and A.J.; visualization, J.H., J.E. and A.J.; supervision, J.H.; project administration, J.H. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

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

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The dataset used for the experiments is available at <https://plantvillage.psu.edu/> accessed on 29 September 2022.

**Acknowledgments:** The authors thank the reviewers and editors for their valuable suggestions for the improvement of the manuscript. The authors also thank their respective institutes.

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

## References

1. Alston, J.M.; Pardey, P.G. Agriculture in the Global Economy. *J. Econ. Perspect.* **2014**, *28*, 121–146. [[CrossRef](#)]
2. Contribution of Agriculture Sector Towards GDP Agriculture Has Been the Bright Spot in the Economy despite COVID-19. Available online: <https://www.pib.gov.in/indexd.aspx> (accessed on 29 September 2022).
3. Li, L.; Zhang, S.; Wang, B. Plant Disease Detection and Classification by Deep Learning—A Review. *IEEE Access* **2021**, *9*, 56683–56698. [[CrossRef](#)]
4. Dawod, R.G.; Dobre, C. Upper and Lower Leaf Side Detection with Machine Learning Methods. *Sensors* **2022**, *22*, 2696. [[CrossRef](#)] [[PubMed](#)]
5. Khan, M.A.; Akram, T.; Sharif, M.; Javed, K.; Raza, M.; Saba, T. An automated system for cucumber leaf diseased spot detection and classification using improved saliency method and deep features selection. *Multimed. Tools Appl.* **2020**, *79*, 18627–18656. [[CrossRef](#)]
6. Scientist, D.; Bengaluru, T.M.; Nadu, T. Rice Plant Disease Identification Using Artificial Intelligence. *Int. J. Electr. Eng. Technol.* **2020**, *11*, 392–402. [[CrossRef](#)]
7. Dubey, S.R.; Jalal, A.S. Adapted Approach for Fruit Disease Identification using Images. In *Image Processing: Concepts, Methodologies, Tools, and Applications*; IGI Global: Hershey, PA, USA, 2013; pp. 1395–1409. [[CrossRef](#)]
8. Yun, S.; Xianfeng, W.; Shanwen, Z.; Chuanlei, Z. PNN based crop disease recognition with leaf image features and meteorological data. *Int. J. Agric. Biol. Eng.* **2015**, *8*, 60–68. [[CrossRef](#)]
9. Li, G.; Ma, Z.; Wang, H. Image Recognition of Grape Downy Mildew and Grape. In Proceedings of the International Conference on Computer and Computing Technologies in Agriculture, Beijing, China, 29–31 October 2011; pp. 151–162.
10. Rauf, H.T.; Saleem, B.A.; Lali, M.I.U.; Khan, M.A.; Sharif, M.; Bukhari, S.A.C. A citrus fruits and leaves dataset for detection and classification of citrus diseases through machine learning. *Data Brief* **2019**, *26*, 104340. [[CrossRef](#)] [[PubMed](#)]
11. Sujatha, R.; Chatterjee, J.M.; Jhanjhi, N.; Brohi, S.N. Performance of deep learning vs machine learning in plant leaf disease detection. *Microprocess. Microsyst.* **2021**, *80*, 103615. [[CrossRef](#)]
12. Karthik, R.; Hariharan, M.; Anand, S.; Mathikshara, P.; Johnson, A.; Menaka, R. Attention embedded residual CNN for disease detection in tomato leaves. *Appl. Soft Comput.* **2019**, *86*, 105933. [[CrossRef](#)]
13. Barbedo, J.G.A. Factors influencing the use of deep learning for plant disease recognition. *Biosyst. Eng.* **2018**, *172*, 84–91. [[CrossRef](#)]
14. Vardhini, P.H.; Asritha, S.; Devi, Y.S. Efficient Disease Detection of Paddy Crop using CNN. In Proceedings of the 2020 International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE), Bengaluru, India, 9–10 October 2020; pp. 116–119.
15. Mohanty, S.P.; Hughes, D.P.; Salathé, M. Using Deep Learning for Image-Based Plant Disease Detection. *Front. Plant Sci.* **2016**, *7*, 1419. [[CrossRef](#)]
16. Panigrahi, K.P.; Sahoo, A.K.; Das, H. A CNN Approach for Corn Leaves Disease Detection to support Digital Agricultural System. In Proceedings of the 4th International Conference on Trends in Electronics and Information, Tirunelveli, India, 15–17 June 2020; pp. 678–683.
17. PlantVillage. Available online: <https://plantvillage.psu.edu/> (accessed on 29 September 2022).
18. Aldhyani, T.H.; Alkahtani, H.; Eunice, R.J.; Hemanth, D.J. Leaf Pathology Detection in Potato and Pepper Bell Plant using Convolutional Neural Networks. In Proceedings of the 2022 7th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 22–24 June 2022; pp. 1289–1294. [[CrossRef](#)]
19. Panigrahi, K.P.; Das, H.; Sahoo, A.K.; Moharana, S.C. Maize leaf disease detection and classification using machine learning algorithms. In *Progress in Computing, Analytics and Networking*; Springer: Singapore, 2020. [[CrossRef](#)]
20. Mohsin Kabir, M.; Quwsar Ohi, A.; Mridha, M.F. A Multi-plant disease diagnosis method using convolutional neural network. *arXiv* **2020**, arXiv:2011.05151.
21. Prodeep, A.R.; Hoque, A.M.; Kabir, M.M.; Rahman, M.S.; Mridha, M.F. Plant Disease Identification from Leaf Images using Deep CNN’s EfficientNet. In Proceedings of the 2022 International Conference on Decision Aid Sciences and Applications (DASA), Chiangrai, Thailand, 23–25 March 2022; pp. 523–527. [[CrossRef](#)]
22. 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. [[CrossRef](#)]
23. Andrew, J.; Fiona, R.; Caleb, A.H. Comparative Study of Various Deep Convolutional Neural Networks in the Early Prediction of Cancer. In Proceedings of the 2019 International Conference on Intelligent Computing and Control Systems (ICCS), Madurai, India, 15–17 May 2019; pp. 884–890. [[CrossRef](#)]
24. Onesimu, J.A.; Karthikeyan, J. An Efficient Privacy-preserving Deep Learning Scheme for Medical Image Analysis. *J. Inf. Technol. Manag.* **2020**, *12*, 50–67. [[CrossRef](#)]
25. Mhatheesh, T.S.R.; Andrew, J.; Sagayam, K.M.; Henesey, L. A 3D Convolutional Neural Network for Bacterial Image Classification. In *Intelligence in Big Data Technologies—Beyond the Hype*; Springer: Singapore, 2021; pp. 419–431. [[CrossRef](#)]

26. Maria, S.K.; Taki, S.S.; Mia, J.; Biswas, A.A.; Majumder, A.; Hasan, F. Cauliflower Disease Recognition Using Machine Learning and Transfer Learning. In *Smart Systems: Innovations in Computing*; Springer: Singapore, 2022; pp. 359–375. [CrossRef]
27. Too, E.C.; Yujian, L.; Njuki, S.; Yingchun, L. A comparative study of fine-tuning deep learning models for plant disease identification. *Comput. Electron. Agric.* **2019**, *161*, 272–279. [CrossRef]
28. Hussain, M.; Bird, J.J.; Faria, D.R. A Study on CNN Transfer Learning for Image Classification. In Proceedings of the UK Workshop on Computational Intelligence, Nottingham, UK, 5–7 September 2018; Volume 840, pp. 191–202. [CrossRef]
29. Barbedo, J.G.A. Impact of dataset size and variety on the effectiveness of deep learning and transfer learning for plant disease classification. *Comput. Electron. Agric.* **2018**, *153*, 46–53. [CrossRef]
30. Upadhyay, S.K.; Kumar, A. A novel approach for rice plant diseases classification with deep convolutional neural network. *Int. J. Inf. Technol.* **2022**, *14*, 185–199. [CrossRef]
31. Panchal, A.V.; Patel, S.C.; Bagyalakshmi, K.; Kumar, P.; Khan, I.R.; Soni, M. Image-based Plant Diseases Detection using Deep Learning. *Mater. Today Proc.* **2021**. [CrossRef]
32. Narayanan, K.L.; Krishnan, R.S.; Robinson, Y.H.; Julie, E.G.; Vimal, S.; Saravanan, V.; Kaliappan, M. Banana Plant Disease Classification Using Hybrid Convolutional Neural Network. *Comput. Intell. Neurosci.* **2022**, *2022*, 9153699. [CrossRef]
33. Jadhav, S.B.; Udupi, V.R.; Patil, S.B. Identification of plant diseases using convolutional neural networks. *Int. J. Inf. Technol.* **2021**, *13*, 2461–2470. [CrossRef]
34. Abayomi-Alli, O.O.; Damaševičius, R.; Misra, S.; Maskeliūnas, R. Cassava disease recognition from low-quality images using enhanced data augmentation model and deep learning. *Expert Syst.* **2021**, *38*, e12746. [CrossRef]
35. Abbas, A.; Jain, S.; Gour, M.; Vankudothu, S. Tomato plant disease detection using transfer learning with C-GAN synthetic images. *Comput. Electron. Agric.* **2021**, *187*, 106279. [CrossRef]
36. Anh, P.T.; Duc, H.T.M. A Benchmark of Deep Learning Models for Multi-leaf Diseases for Edge Devices. In Proceedings of the 2021 International Conference on Advanced Technologies for Communications (ATC), Ho Chi Minh City, Vietnam, 14–16 October 2021; pp. 318–323. [CrossRef]
37. Astani, M.; Hasheminejad, M.; Vaghefi, M. A diverse ensemble classifier for tomato disease recognition. *Comput. Electron. Agric.* **2022**, *198*, 107054. [CrossRef]
38. Gokulnath, B.V. Identifying and classifying plant disease using resilient LF-CNN. *Ecol. Inform.* **2021**, *63*, 101283. [CrossRef]
39. Enkvetchakul, P.; Surinta, O. Effective Data Augmentation and Training Techniques for Improving Deep Learning in Plant Leaf Disease Recognition. *Appl. Sci. Eng. Prog.* **2022**, *15*, 3810. [CrossRef]
40. Militante, S.V.; Gerardo, B.D.; Dionisio, N.V. Plant Leaf Detection and Disease Recognition using Deep Learning. In Proceedings of the 2019 IEEE Eurasia Conference on IOT, Communication and Engineering (ECICE), Yunlin, Taiwan, 3–6 October 2019; pp. 579–582. [CrossRef]
41. Fuentes, A.; Yoon, S.; Kim, S.C.; Park, D.S. A Robust Deep-Learning-Based Detector for Real-Time Tomato Plant Diseases and Pests Recognition. *Sensors* **2017**, *17*, 2022. [CrossRef]
42. Ferentinos, K.P. Deep learning models for plant disease detection and diagnosis. *Comput. Electron. Agric.* **2018**, *145*, 311–318. [CrossRef]
43. Lee, S.H.; Goëau, H.; Bonnet, P.; Joly, A. New perspectives on plant disease characterization based on deep learning. *Comput. Electron. Agric.* **2020**, *170*, 105220. [CrossRef]
44. Zhong, Y.; Zhao, M. Research on deep learning in apple leaf disease recognition. *Comput. Electron. Agric.* **2020**, *168*, 105146. [CrossRef]
45. Zhang, Y.; Song, C.; Zhang, D. Deep Learning-Based Object Detection Improvement for Tomato Disease. *IEEE Access* **2020**, *8*, 56607–56614. [CrossRef]
46. Chen, J.; Chen, J.; Zhang, D.; Sun, Y.; Nanehkaran, Y. Using deep transfer learning for image-based plant disease identification. *Comput. Electron. Agric.* **2020**, *173*, 105393. [CrossRef]
47. Shrivastava, V.K.; Pradhan, M.K.; Minz, S.; Thakur, M.P. Rice plant disease classification using transfer learning of deep convolution neural network. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2019**, *3*, 631–635. [CrossRef]
48. Shrivastava, V.K.; Pradhan, M.K. Rice plant disease classification using color features: A machine learning paradigm. *J. Plant Pathol.* **2021**, *103*, 17–26. [CrossRef]
49. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [CrossRef]
50. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arXiv:1409.1556. [CrossRef]
51. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely Connected Convolutional Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
52. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A.A. Inception-v4, inception-ResNet and the impact of residual connections on learning. In Proceedings of the 31st AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; Volume 31, pp. 4278–4284. [CrossRef]
53. Agarwal, M.; Gupta, S.K.; Biswas, K. Development of Efficient CNN model for Tomato crop disease identification. *Sustain. Comput. Inform. Syst.* **2020**, *28*, 100407. [CrossRef]

54. Kaushik, M.; Prakash, P.; Ajay, R.; Veni, S. Tomato Leaf Disease Detection using Convolutional Neural Network with Data Augmentation. In Proceedings of the 2020 5th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 10–12 June 2020; pp. 1125–1132. [[CrossRef](#)]
55. Rangarajan, A.K.; Purushothaman, R.; Ramesh, A. Tomato crop disease classification using pre-trained deep learning algorithm. *Procedia Comput. Sci.* **2018**, *133*, 1040–1047. [[CrossRef](#)]