Citrus Leaves Disease Detection and Classification on Image Recognition Using Deep Convolutional Neural Networks

Khalid Mahmood [1], Qamar Abbas [2], Tahani Alsaedi [3], Mahwish Kundi [4], Rahim Khan[4] , Weam M. Binjumah [3] , Aftab Ahmad[4]

Abstract: Agriculture production is a crucial economic backbone for any country, and citrus plant disease poses a significant threat to this sector, leading to decreased yields and heavy losses. Automated systems for disease detection and classification can aid in combating this issue and promoting growth and development. Deep learning models in recent years have demonstrated promising results in various artificial intelligence tasks. This paper presents an integrated and enhanced approach for detecting citrus leaf disease using a deep convolutional neural network. Healthy citrus leaves may be distinguished from seven prevalent illnesses using our proposed approach, comprising citrus powdery mildew, greening, melanose, bacterial spot, black spot, canker, and healthy. The model incorporates multiple hidden layers and data augmentation for improved image recognition and classification. Our proposed model is tested against other deep learning models on a citrus leaves dataset and outperformed previous studies in various measurement metrics. With a test accuracy of 97.66%, our model serves as a reliable tool for citrus plant disease detection.

Keywords: Citrus leaves, Disease detection and classification, Image recognition, Deep convolutional neural networks.

1. Introduction

In Pakistan, sustainable agricultural expansion is essential for rural development and food security. It contributes 22.7% of the GDP, roughly 37.4% of the labor force is employed, it controls the rural environment, and it serves as an environmental buffer to protect and enhance the development of a climate-resilient ecosystem. [1]. Between 2021 and 2022, the agriculture sector had a spectacular growth of 4.40 percent, exceeding the target of 3.5% and the rise of 3.48% the year before. Second, only to bananas in terms of global fruit output, citrus is grown on more than 200,400 hectares and produces 158 million tons of fruit annually. China produces the most citrus fruit on the planet with 35.2% of all fruit production, or 2.4 million tons, produced in the citrus industry in 2014–2015. Mandarins (Kinnow), oranges, grapefruit, lemons, and limes are citrus fruits, and mandarins (Kinnow) are particularly significant to Pakistan

[1] *Institute of Computing and Information Technology, Gomal University, D.I.Khan, Pakistan*

[2] *Department of Computer Science, Faculty of Computing and Information Technology, International Islamic University, Islamabad, Pakistan*

[3] *Applied College, Taibah University, Madina 42353, Saudi Arabia*

[4] *Abdul Wali Khan University, Mardan, Pakistan*

* *Corresponding author: mkundi@awkum.edu.pk;*

[2]. There are more than 140 countries in the world where citrus is grown. Brazil, China, Mexico, the United States, Spain, and India are the world's top citrus-producing nations. Pakistan is one of the top ten countries in the world for citrus production (Source: FAOSTAT). With a 95% market share, the most prevalent citrus species in Pakistan is called Kinnow (Citrus reticulate). In the area and citrus production, Punjab contributes 94% and 96%, respectively [3]. Citrus is a great source of vitamins C and E, sugar, organic acids, amino acids, and minerals including calcium and magnesium. It also has a high nutritional value. The average yield in Pakistan is 2.36million tons, spread across 1,000 hectares, with an annual export of 282,000 tons, selling for 7,313 million rupees [4]. Citrus fruit plants are very vulnerable to various infections and bacterial diseases including Bacterial Spot, Black Spot, Canker, Citrus Powdery Mildew, Greening and Melanose represent a persistent threat to citrus farming and have a significant economic impact on all citrus-growing regions worldwide [5]. Citrus trees can get the highly contagious canker, which is typically found on the leaves or fruit. Deep learning algorithms have become a potent tool for disease identification in numerous fields, including agriculture, in recent years. Convolutional neural networks (CNNs), in particular, have demonstrated encouraging results in the detection and classification of a variety of citrus plant diseases. [6,7]. In this paper, we suggest a CNN-based deep learning method for identifying citrus plant disease. Our model is designed to distinguish healthy citrus leaves from a variety of common diseases, including bacterial spots, blackspots, canker, citrus powdery mildew, greening, melanose, and healthy. We incorporate multiple hidden layers and data augmentation techniques to extract relevant features and improve image recognition and classification accuracy. Our model is tested on a citrus leaves dataset and evaluated against other deep learning models. The results of our study are expected to provide a reliable and efficient tool for citrus plant disease detection and contribute to the sustainable expansion of the agriculture industry in Pakistan.

2. Literature Review

Plant diseases can have significant impacts on both food safety and agricultural product output. A significant portion of the automated systems developed for plant disease detection have been based on digital images, allowing for the swift implementation of algorithms. The challenges associated with autonomous identification of plant illness have been addressed using conventional machine learning techniques such as, Multilayer Perceptron Neural Networks, Support Vector Machines (SVMs) and Decision Trees. Varshneyet al. [8] propose a novel DL method on leaf plant disease detection utilizing a transfer learning methodology When SVM is used for classification and a Convolutional Neural Network (CNN) is used to extract features. Utilizing the PlantVillage benchmark dataset, the suggested model is assessed. Results show that the suggested model performs better than earlier work, achieving an accuracy of 88.77% on a training dataset. Bharateet al. [9] gives a summary of the various image-processing methods used to detect plant diseases. These methods include research on disease detection in various plants such as apples, grapes, peppers, pomegranates, and tomatoes. Researchers have developed a variety of methods for recognizing and classifying different types of fruit diseases, for example, the use of the YCR color model, HSV color model [10], and VGG16. The YCR color model was utilized by Dhanesha et al. [11] in order to segment the areca nut bunches, calculating the dice similarity coefficient and volumetric overlap error to determine how similar the input image and the ground truth are. Ghosaland Sarkar [12] developed a VGG16 model [13] that incorporates transfer learning for the purpose of diagnosing diseases affecting rice plants. The researchers trained this classification system with the help of four different image classes, and now, VGG16 has an accuracy percentage of up to 92.4%. Kumar et al. [14] created a technique that can identify diseases based on the look of coffee plant leaves, using CNN with data augmentation, fuzzy logic-based expert systems, transfer learning methods, and radial basis function neural networks. Coulibaly et al. [15] use a GG16 model that had been trained through transfer learning to detect diseases in millet crops, 124 images of leaves are separated into two categories: those with mild infections and those that appeared to be healthy. The level of precision achieved by the VGG16 model is 95%.

2.1 Deep learning models for plant disease detection

By identifying patterns and features in photos of both healthy and diseased plants, deep learning algorithms have been successfully utilized to detect plant illnesses. A particular kind of deep learning model called Convolutional Neural Networks (CNNs) has demonstrated success in categorizing photos of plant leaves, and RetinaNet [16] is another example of a deep learning model that can detect multiple diseases in a single image, It can assist farmers in reducing the spread of illness and boosting crop yield.

Pavan et al. [17] give a thorough analysis of the application of convolutional neural networks (CNNs) for plant disease detection from images. The authors argue various CNN architectures, such as AlexNet, VGG, and ResNet, and their performance on different plant species and diseases. Additionally, they discuss the limitations and challenges of using CNNs to detect plant diseases as well as the state of the art. Saleem et al. [18] utilized DL meta-architectures and TensorFlow object detection framework to tackle the intricate tasks of identifying the location and categorizing diseases in plant leaves. This led to a high level of accuracy in recognizing different types of damaged and healthy leaves, achieving a mean average precision of 73.07%. The methodology can be utilized in other areas of agriculture and has the potential for future utilization in real-the time detection of plant diseases in both controlled and non-controlled environments. Tan et al. [19] introduce EfficientDet, a family of convolutional neural network (CNN) models that are designed to be both accurate and efficient for object detection tasks. The authors present results of using EfficientDet models for plant disease detection, showing that they achieve comparable or higher accuracy than existing methods while being more scalable and efficient.

2.2 Image datasets for plant disease detection

Image datasets for plant disease detection typically incorporate a lot of pictures of images of both healthy and diseased plants and can include a large number of diverse plants species and disease types. These datasets are essential for the creation of precise and trustworthy algorithms since they are utilized to train identification of plant diseases using deep learning models. To evaluate and compare the effectiveness of various models, a standard benchmark is employed. Table 1 lists various datasets that are used for plant disease detection. The plant species and disease categories also vary among the datasets. Some datasets, such as PlantVillage and Embrapa Dataset, have specific information about the number of plant species and disease categories included. Other datasets, such as IPM and Bing, do not provide this information. The backgrounds of the photographs in the datasets also differ, with some being captured in lab settings with fixed backdrops and others in actual field settings.

Table 1. Plant diseases datasets

Dataset Name	Institution	Number of Images	Plant Species	Disease Categories	Background
PlantVillage [20]	Penn State University	54,305	14	38	Lab conditions with fixed background
IPM [21]	N/A	119	N/A	N/A	Fixed and background conditions
Bing [21]	N/A	121	N/A	N/A	Fixed background and in-field
PlantVillage (extended) [22]	N/A	87,848	25	58	Infield and laboratory conditions
Embrapa Dataset [23]	Embrapa Agriculture Institute	46,513	18	93	In-field
Strawberry Dataset [24]	N/A	3531	N/A	4	N/A
Rice dataset[25]	Bangladesh Rice Research Institute	1426	N/A	N/A	Real-field conditions
Apple Dataset [26]	N/A	3651	N/A	4	N/A
Maize Dataset [27]	Fujian Institute of Subtropical Botany	481	1	4	In-field
Rice Dataset [27]	Fujian Institute of Subtropical Botany	560	N/A	5	Laboratory and in-field
PlantDoc [28]	N/A	2598	N/A	17	Field
Turkey-PlantDataset [29]	Agricultural Faculty of Bingol and Inonu Universities	4447	N/A	15	N/A

Other datasets on plant diseases are also available in the literature not limited to [30–33].

2.3 *Transfer Learning for Plant Disease*

Transfer learning is a deep learning technique that makes use of previously trained models on similar tasks to enhance performance and speed up training for new tasks. It is especially useful when data is limited, and allows for better generalization and helps avoiding overfitting by fine-tuning the pre-trained models on large dataset for the new task [34]. Sagar et al. [35] explains how neural networks can be utilized for plant disease recognition in the context of picture classification, and discusses the issue of multi-class classification. The study examines the effectiveness of transfer learning methods for detecting plant diseases, including fine-tuning, feature extraction, and training on 38 disease classes are included in the Plant Village dataset, which is publicly available. Five distinct network backbone architectures VGG16, ResNet50, InceptionV3, InceptionResNet, and DenseNet169 are compared, ResNet50 outperforms other networks on the test set. When employing ResNet50, the model produced the best results, with accuracy of 0.982, precision of 0.94, recall of 0.94, and F1 score of 0.94. These measurements include metrics for recall, accuracy, and class-wise confusion. Other studies that use transfer learning for plant disease detection can be found at [36–40].

2.4 Handheld device and smartphone-assisted plant disease diagnosis

One way to adapt deep learning models to run on handheld devices and smartphones is by reducing the model's complexity and size. This can be achieved through techniques such as model compression and pruning, which remove redundant and unnecessary parameters from the model. Another way is to use specialized deep learning architectures designed for mobile devices, such as MobileNet, which use depthwise separable convolutions to reduce computational complexity while maintaining high accuracy. Additionally, quantization of the model can also be done to run on specific hardware of mobile devices. With the help of the dataset's 7176 photos of tomato leaves, Elhassouny et al. [41] suggested an effective smart mobile application model that is inspired by the MobileNet CNN model that can identify the 10 most prevalent forms of tomato leaf disease. A deep learning-based rice disease detection system is proposed by Andrianto et al. [42] and comprises of a machine learning application on a cloud server and an application on a smartphone. The system boasts a 100% train accuracy and a 60% test accuracy thanks to its VGG16 design. By adding more data and improving the dataset quality, the test accuracy can be increased. The aim of this system is to improve the control of rice plant disease and maximize yields. The diagnosis of common illnesses on terrestrial plants in the Philippines using image processing and deep learning neural network is emphasized by Valdoria et al. [43]. Images of the terrestrial plant were taken using Android-based smartphones to identify the disease of the plant, and a deep learning neural network algorithm was used to differentiate the sickness of the terrestrial plants. Given the quantity of photos used, classification algorithms that could accurately and quickly detect diseases were employed to train the findings. The information received from plant leaves, such as color, texture, and shape, is used by Diah et al. [44] to detect and categorize tomato plant illnesses using a deep learning model based on convolutional neural network (CNN) architecture. The model's overall validation and training accuracy rate for the outcomes of the categorization was 95.8%.

2.5 Key findings from the literature review

The future of agricultural management and the ability to detect plant diseases could both be significantly impacted by deep learning. Convolutional neural networks (CNNs) and other deep learning techniques have been proven to detect plant diseases with high accuracy rates that frequently outperform more conventional image processing techniques [45]. Additionally, deep learning models can be trained to detect diseases by using IoT-based systems using edge devices for farmers with rapid and reliable information to make decisions about treatment and management [46]. One potential impact of deep learning in plant disease detection is the ability to improve crop yields and reduce crop losses. Early detection of plant diseases can enable farmers to take preventative measures, such as applying pesticides or isolating infected plants, to reduce the spread of disease and minimize damage [22]. This could lead to increase crop yields and improved food security, particularly in regions where crop diseases are prevalent. Making better use of resources is another possible benefit of deep learning for detecting plant diseases. Deep learning algorithms can quickly and precisely identify plant diseases by analyzing enormous volumes of data, including photos and sensor data [47]. This can enable farmers to focus their efforts on specific areas of a field or specific plants, rather than treating the entire field, reducing the number of resources required for disease management. Deep learning has the ability to decrease the number of false positives and negatives that can be brought on by employing conventional approaches while also increasing the accuracy of plant disease identification [48]. Using deep learning models in crop management can help farmers to make more informed decisions on disease management and reduce the number of chemical inputs used in agriculture. In summary, deep learning has the potential to greatly impact plant disease detection and the future of agricultural management by enabling farmers to detect diseases, improve crop yields, and make more efficient use of resources quickly and accurately.

3. Methodology

A Deep Convolutional Neural Network (DCNN) is suggested in this research study. The proposed system is elaborated with a flowchart depicted in Fig. 1

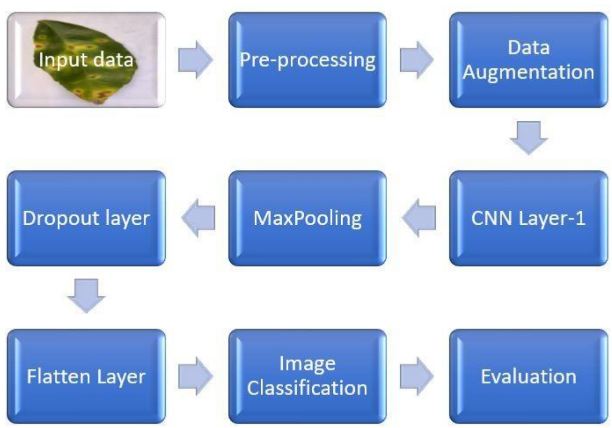

Figure 1. Workflow model for the citrus leaf disease detection

3.1 Dataset Acquisition and Splitting

Datasets are required at all steps comprising from training to evaluation of algorithm during image analysis study. 2489 images in total are gathered from citrus dataset [49] , self-collected dataset and Plat Village dataset [20]. The images are divided into infected and healthy images grouped into seven categories, comprising of Bacterial spot, Black spot, Canker, Citrus Powdery Mildew, Greening, Melanose and healthy images shown in Table 2.

Table 2. This table is providing information about the number of images of different categories of plant diseases and healthy plants that were used for training and testing using a machine learning model. Plants belonging to various categories are shown.

Category	Disease	Image Count (Training)	Image Count (Testing)
Leaf	Bacterial Spot	88	58
	Black Spot	471	61
	Canker	513	147
	Citrus Powdery Mildew	35	14
	Greening	684	121
	Melanose	65	31
	Healthy	128	76
	Total	1984	505
	Grand Total	2489	

Dataset is divided into three parts: Using the Python Split Folders library, (i) training data, (ii) test data, and (iii) validation data were used. The tests make use of an Intel CORE i7, 7th generation, 64-bit operating system, and 8 GB of RAM. The Keras package, Tensorflow, and Anaconda Environment are utilized to run and execute the proposed DCNN model.

3.1.1 Train Data

With regards to the training data, 75% is used to create a DCNN model. Our model is trained using the provided data. The training data consists of both the input and the expected results.

3.1.2 Test Data

25% of the initial dataset is used for evaluation, that makes up the testing data. After the model has finished training, the test data is utilized to forecast its behavior.

3.1.3 Validation Data

In order to reduce the model's underfitting and overfitting problems [50], which usually occurs during testing phase, when a machine exhibits excellent behavior during training but suffers during testing with fresh data. The test dataset is then used to evaluate the final model's generalizability and predict how well it will perform with hypothetical data. It is crucial to keep in mind that the validation dataset serves as a representation of the unobserved data that the model would face in practice; otherwise, the model could not work well when applied to data from the actual world.

4. Overview of the Proposed Approach

The following modules make up the suggested technique: (i) pre-processing input image dataset (ii) Data Augmentation (iii) CNN layer-1 (iv) Maxpooling Layer-1 , (v) CNN layer-2, (iv) Maxpooling Layer-2 (v) Dropout Layer, (vi) Flatten layer and (vii) Classification Layer.

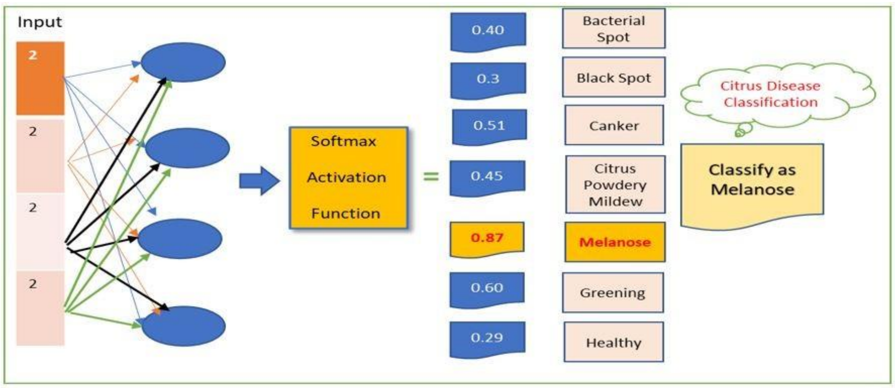

Figure 2. The figure depicts DCNN model module consisting of several layers piled on top of one another. Convolutional layers that apply filters are applied to the input image as it passes through each layer to extractfeatures from the image, max pooling layers to reduce spatial resolution and increase robustness, a flatten layer to convert the image into a one-dimensional array, and to tackle the overfitting problem a dropout layer is suggested. The output of the model is generated by a fully connected layer and is the probability ofeach class, with the highest probability representing the predicted class. The DCNN model extractsfeatures from the input image, and uses them to classify the image into one of the predefined classes.

4.0.1. Pre-processing of the Acquired Image dataset:

The input image is pre-processed in this module using the Keras image Data Generator module [51]. Image scaling, normalization and data normalization functions are applied in this stage to normalize and scaled the image repository for further processing.

4.0.2. Data Augmentation:

Data augmentation aims to improve image identification and lessen the overfitting issue. [52]. It helps in better data visualization by artificially expanding datasets using different options like Image Zoom, Rotation, Flipping and Blur options.

4.0.3. CNN Layer-1:

Obtaining a feature map of the input image is the layer's goal. This objective is achieved using convolution operation on input image [53].

4.0.4. **MaxPooling of Layer-1:**

The features obtained from CNN Layer-1 are reduced in size, when they are transmitted to this layer, different pooling options can be chosen in this layer for feature selection. This layer reduces the noise and variations [54].

4.0.5. CNN Layer-2:

This layer works in similar fashion as Layer-1 of CNN, but it gets low-level features while the Layer-1 gets high level features.

4.0.6. MaxPooling of Layer-2:

This layer works in similar fashion as MaxPooling Layer-1, with the aim of dimensionality reduction.

4.0.7. Dropout Layer

To avoid overfitting of the data in testing phase dropout layer is used [55]. In the proposed study we have set the rate=0.2.

4.0.8. Flatten Layer:

This layer is responsible for conversion of resultant two-dimensional arrays from pool feature maps into single continuous vector [56].

4.0.9. Classification Layer

The output obtained from flatten layer is used to get the one of the possible images for classification using activation function. Using SoftMax Activation function the leaf disease is inspected [57].

5. Experimental Results and Discussion

5.1. Dataset Description:

To carry out the experiment, we selected a sample of images from the benchmark databases. The image dataset has been collected from Kaggle Citrus dataset [58], Plant Village dataset [20] and our own, amassed image library for the analysis of the proposed study. Citrus leaf diseases such as Bacterial spot, Black spot, Canker, Citrus Powdery Mildew, Greening, Melanose and healthy images are detected and classified using these datasets shown in Fig. 3.

5.2. Preprocessing of the Input Image:

The system initiates by taking input images from the citrus leaf dataset. The input shape parameter is set to accepts a three-dimensional matrix image. After input, the image is rescaled and normalized for further processing.

5.3. Data Augmentation:

For accurate prediction, deep learning models for classification tasks need a lot of data. On the other hand, the lack of updated image collections for citrus illnesses and the small number of image repositories continue to hinder the automatic detection of diseases that affect citrus leaves. To enhance the quantity of the data and prevent the overfitting issue, data augmentation operations on the training set were carried out to address this issue [59]. When the model performs estimably on the training data but poorly on the test data, overfitting problems occur. Data augmentation parameters, such as RandomRotation, RandomZoom, RandomFlip, RandomContrast and RandomMirroring employing principal component analysis, are applied to the datasets. To make implementation easier and speed up processing, the images are then downsized to a standard 256 x 256 dimension.

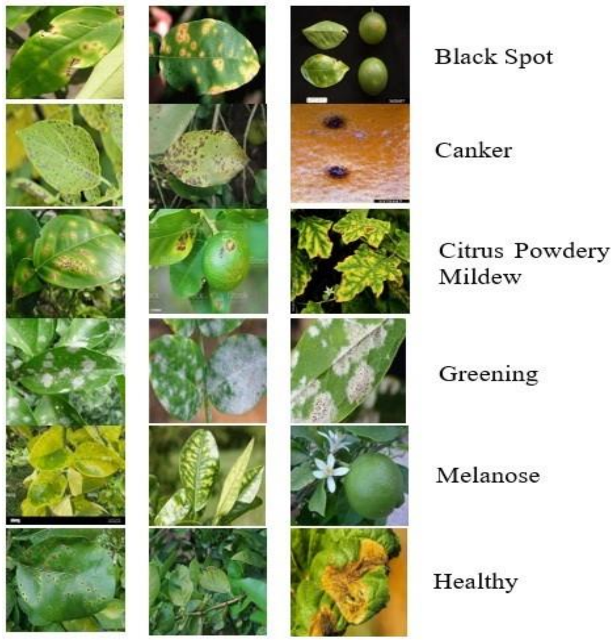

Figure 3. The figure shows a dataset of citrus leaves images, which are divided into infected and healthy images. The images are grouped into seven categories, each representing a different citrus leaf disease: bacterial spot, black spot, canker, citrus powdery mildew, greening, melanose, and healthy images. The dataset is used to train and test a DCNN model for citrus plant disease detection. Based on the features that were retrieved from the images, the model is trained to categorize the images into one of the seven illness categories.

5.4 CNN Layer-1:

The convolutional layers are the essential building blocks of CNN. The convolution Layer-1 performs the function of feature extraction. Convolution is a mathematical operation which act as a filter to an input image *I* with respect to its dimensions W_{in} *X* H_{in}. Its hyper parameters are filter of dimensions *K x K* size, stride S and padding P. A feature map is generated as an output of repeated applications of the same filter, which indicates the locations and strength of detected feature of an input image. Equation (1) can be used to predict the output size.

$$Mx = \frac{Ix-Kx}{Sx} \qquad My = \frac{Iy-Ky}{Sy} \qquad (1)$$

Where in the above equation *(Mx, My)* represents the matrix, *(Ix,Iy)* represents the input size, *(Kx,Ky)* represents the Kernel size and *(Sx,Sy)* represents the Stride with respect to row and column respectively.

5.5 MaxPooling Layer-1:

The pooling layer performs the down sampling (sub sampling) operation on the output generated by convolution layer to reduce its dimensionality. In Particular, the maximum and average pooling are the special type of pooling, in which the maximum and in later case the average value is considered for pooling operation. The output is determined with the following equation by applying max operation on the input feature matrix.

$$M(r,c) = Max(Mx, My) \quad (2)$$

5.6 CNN Layer-2:

CNN's Layer-2 is in charge of extracting high level feature, received from the previous Maxpooling layer. The second layer of CNN is similar in function to the first convolution layer. Equation 1 is used to calculate the CNN layer-2.

5.7 MaxPooling Layer-2:

The objective of this layer is to condense the matrix scale. This layer is computer in similar function with regard to first pooling layer. The Maxpooling layer-2 is obtained using equation 2.

5.8 Dropout Layer:

Co-adoption is the phenomena which occurs in fully connected layers due to large number of neurons. Due to this co-adoption issue multiple neurons extracts similar hidden features from the input data, because of identical neurons. It leads towards wastage of machine resources, and more importantly leads towards overfitting in the test data due to duplicated extracted features (Srivastava, Hinton et al. 2014). So, in training phase we randomly dropout some fraction of layer's neuron by zeroing out the values of neuron. In the proposed study we have set the dropout rate to 0.2, meaning to randomly dropout 2 neurons.

5.9 Flatten Layer:

This layer receives the output from maxpooling layer-2 as input. This layer's is responsible to transform the output of the pooled feature matrix M into a feature vector or column. By restructuring the function, the feature vector can be obtained from feature matrix M.

5.10 Classification Layer:

The probabilities of different categories for citrus leaf images are calculated through dense layer comprising of different neurons by using Softmax function. The resultant output is obtained by the following equation.

$$z = \sum_{i}^{l} wixi + b \quad (3)$$

Whereas, x represents the input vector, w is a weight vector and b represent the biasness. Z is the activation function which generate maximum probability value as the classification output.

6. Applying a Citrus Leaf Disease Classification Model:

This segment elaborates the process about citrus leaf disease recognition and classification using proposed model. The suggested study's workflow model to identify the seven different forms of leaf diseases is shown in Fig. 2 including Bacterial spot, Black spot, Canker, Citrus Powdery Mildew, Greening, Melanose and healthy images were identified and classified using said approach.

6.1. Data Acquisition of Input Image:

The first step is to collect input images dataset. These images are converted into pixel with three dimensions' matrix (256,256,3), with height, width and RGB color combination ratio respectively. These images are rescaled and normalized for further processing.

6.2. Data Augmentation:

Data augmentation job is to supplement existing data in order to improve model performance and overcome the overfitting problem. By applying data augmentation hyper parameters, the image dataset size can be increased for better training.

6.3. Convolutional Layer-1:

This layer is responsible for high level feature extraction. The input images are transformed into feature map using filter and padding process. The process is described in section 4.4.

6.4 Maxpooling Layer-1:

The Maxpooling layer performs the downsampling operation on the feature matrix obtained from convolution layer. It results in dimensionality reduction of the feature matrix.

6.5 Convolution Layer-2:

With the intention of extracting the feature matrix, this layer operates identically to the first convolution layer. It gets input from the previous Maxpooling layer-1. The pooling layer condense the image feature matrix.

6.6 MaxPooling Layer-2:

The second maxpooling layer's goal is to decrease the dimension size so that features are more easily discernible.

6.7 Dropout Layer:

The dropout layer drops the identical neuron with similar weights to reduce the burden in training phase.

6.8 Flatten Layer:

The pooled feature matrix is transformed into a single-dimensional feature vector via the flatten layer.

6.9 Classification Layer:

The categorization layer determines the likelihood that any of the seven different citrus leaf diseases will occur. The output probability ranges between 0 and 1. We obtain the following probabilities of each citrus leaf disease by applying the Softmax function with score: Bacterial spot (0.40), Black spot (0.3), Canker (0.51), Citrus Powdery (0.45) Mildew (0.60), Greening (0.27), Melanose (0.87) and healthy (0.29). Using the above probability computation, the Melanose gets the higher probability i.e., 0.87, so the Melanose picture is the resultant output.

7. Results and Discussion

This section assesses and displays the results and conclusions attained via the use of various experiments and empirical settings for the validation of our suggested study. The Citrus detection function is shown in Algo 1. Establishing Parameters for CNN Model for Citrus leaf Disease detection and classification are explained in Table 3.

Table 3. Parameters used for DCNN

Parameter	Value
Image Dimensions	256 x 256
# of CNN Layers	2..4
# MaxPool Layers	2..3
# of Filters	64,32,16
Dimension of Filters	2,3
Activation Functions	Softmax, Relu
# of Epochs	10..50
Batch size	64

Algorithm 1 CNNWorkflowModel

Input: train

Output: test

Data: Training/Testing set *x*

Function CNNWorkflowModel(*train, test*)**:**

Import ImageDataGenerator from tensorflow.keras.preprocessing.image

Import ImageDataGenerator from tensorflow.keras.preprocessing.image

Define data_augmentation as a sequence of RandomContrast and RandomZoom layers

Define model as a Sequential model

Add data_augmentation to model

Add a Conv2D layer with 64 filters, 3x3 kernel, relu activation, and input shape of(256,256,3) to model

Add a MaxPool2D layer with 2x2 pool size to model

Add a Conv2D layer with 32 filters, 3x3 kernel, relu activation to model

Add a MaxPool2D layer with 2x2 pool size to model

Add a Flatten layer to model

Add a Dropout layer with rate 0.2 to model

Add a Dense layer with 7 units and softmax activation to model

Compile model with loss of 'categorical_crossentropy', optimizer of 'adam', and metricof 'accuracy'

Define train_dataset as the flow of images from the directory

Define'Citrus_Dataset/Processed_output/train/' with target size of (256,256), batchsize of 64, and class mode of 'categorical'

Define validation_dataset as the flow of images from the directory
Define'Citrus_Dataset/Processed_output/val/' with target size of

Define Fit the model with train_dataset, validation_data as

Add validation_dataset, steps_per_epoch as 3 and 30 epochs

CNNWorkflowModel

To gauge the effectiveness of the suggested task, we used various ratios of training and testing samples. We have used 75-25 and 60-20-20 split ratio for evaluation purpose. These variations in the train-test samples provide a way to check the accuracy of model, because as the training sample grows, its efficiency also increases. The precision [61], refers to the number of accurately detected positive occurrences and measures the accuracy of positive instances, and can be calculated using equation (1) Whereas in memory the TP denotes the number of properly detected positive cases, the FN denotes the number of false negatives, or the number of positive cases that were mistakenly labeled as negatives.

$$Precision = \frac{TP+TN}{TP} \qquad (4)$$

The recall [62], represents the percentage of negative instance, and refers to the number of negative instances which were wrongly marked as positive. The TN indicates the amounts of cases or genuine negatives that were actually negative but were mistakenly categorized as positive, and the FP represents those quantities. It is represented in equation (5).

$$Recall = \frac{TN+FP}{TN} \qquad (5)$$

The accuracy [63], represent the joint distribution of precision and recall. The accuracy returns the overall performance of model. It is represented in equation.

$$Accuracy = \frac{TN+FP+TP+TN}{TN+TP} \qquad (6)$$

We set different experiment settings for DCNN, with various epochs settings. In training phase for 30 epochs, we achieve 93.76% accuracy score, for 50 epochs we get 95.80% accuracy score. and for 100 epochs we get 97.66% highest accuracy score. The proposed study achieves an accuracy of 97.66% score, with an increase of 12% accuracy score with comparison to [64] and increase of 4.16% with comparison to [65] respectively. Fig 4 elaborate the results.

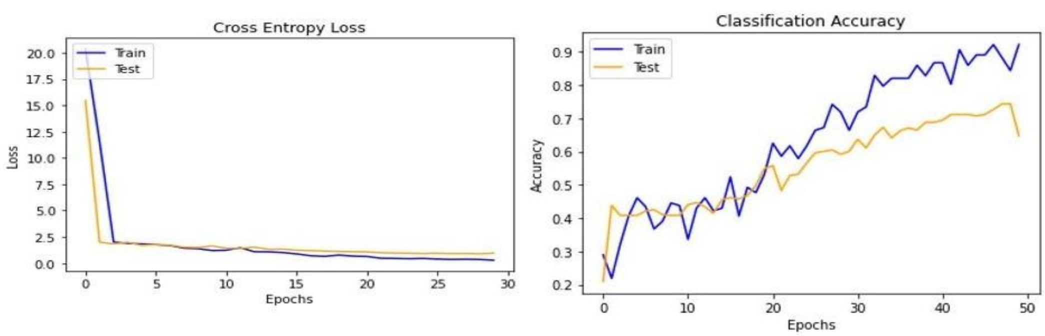

Figure 4. On the left: For cross entropy loss graph, it can be seen that with the increasing number of epochs the training/testing error decreases. On the Right: For the accuracy graph, it is vibrant that the accuracy of the DL model increases with more iterations.

Table 4. DCNN score with different settings.

Recall	Precision	F-measure	Accuracy	
DCNN (30)	91.74	90.14	92.15	93.76 %
DCNN (50)	93.74	94.50	95.00	95.80 %
DCNN (100)	95.76	95.99	96.56	97.66 %

Table 5. Comparative analysis with other studies.

Baseline Study		Technique	Accuracy Score
Liu et al. [64]	Image recognition of citrus diseases	CNN	87.28%
Elaraby et al. [65]	Classification of citrus diseases using	CNN + AlexNet	93.5 %
Luaibi et al. [66]	Detection of citrus leaf diseases using	CNN	91.75 %
Proposed Study	**Citrus leaves disease detection and classification**	**CNN + Data Augmentation**	97.66%

8. Threats to Validity

The following shortcomings are observed for the suggested study.

- Dataset Size: Due to limited dataset size, a diverse dataset will further improve accuracy of the system.
- Limited classes: The model may not be able to generalize well to other classes if the number of classes is limited in the training dataset.
- Limited images: The model may not be able to generalize well if the number of images per class is limited in the training dataset.
- Environmental conditions: the model performance may vary depending on the conditions under which the images are taken (e.g., lighting, temperature, humidity).

9. Conclusions

Convolutional neural networks (CNNs), for example, have been discovered to be extremely efficient at accurately identifying plant illnesses. The proposed DCNN model is capable of classifying diseases that affect citrus leaves. It distinguishes the healthy leaf images from the infected leaf images. The proposed model performs data acquisition, pre-processing, and application of DCNN model to the acquired dataset into multiple distinct classes. The DCNN module is compiled of multiple convolutional and max pooling layers. While the second convolutional layer concentrates on high-level characteristics, the first convolutional layer collects low level features. Coupled with maxpool's layers to downsample the feature matrix generated by the convolution layer. The dropout layer is used to randomly drop the identical weights neurons to tackle the overfitting issue. Our proposed model classifies the citrus leaf diseases into Bacterial spot, Black spot, Canker, Citrus Powdery Mildew, Greening, Melanose, and healthy images with an accuracy of 97.66% outperforming the state of the state deep learning models. The proposed model can be further extended to mobile applications (IoT), which facilitates the farmers with prompt and timely messaging to adopt precautionary measures to take necessary actions to save the crops. The real-time image capturing (sensory data) can be used for image extraction. To incorporate further hybrid combinations of deep learning approaches like combining CNN with LSTM, or Bi-LSTM. The proposed work can be extended further from leaf disease detection to other parts of plants such as stems and flowers.

Funding: No funding has been received for this study.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data can be provided on request.

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

References

[1] Naqvi, S.A.H.; Wang, J.; Malik, M.T.; Umar, U.U.D.; Hasnain, A.; Sohail, M.A.; Shakeel, M.T.; Nauman, M.; Hassan, M.Z.; Fatima, M.; et al. Citrus Canker—Distribution, Taxonomy, Epidemiology, Disease Cycle, Pathogen Biology, Detection, and Management:A Critical Review and Future Research Agenda. *Agronomy* **2022**, *12*, 1075.

[2] Siddique, M.I.; Garnevska, E. Citrus value chain (s): A survey of Pakistan citrus industry. *Agric. Value Chain* **2018**, *37*.

[3] Catara, A.; Polizzi, G. Dry root rot of citrus: symptoms, causing and susceptibility of rootstocks. *Rivisto di fruticolture* **1999**, *6*, 38–41.

[4] Dandurand, L.; Menge, J. Influence of Fusarium solani on citrus root rot caused by Phytophthora parasitica and Phytophthora citrophthora. *Plant and soil* **1992**, *144*, 13–21.

[5] Mendonça, L.; Zambolim, L.; Badel, J. Bacterial citrus diseases: major threats and recent progress. *Journal of Bacteriology &Mycology: Open Access* **2017**, *5*, 340–350.

[6] Liaqat, M.I.; Hassan, M.A.; Shoaib, M.; Khurshid, S.K.; Shamseldin, M.A. Sentiment analysis techniques, challenges, and opportunities: Urdu language-based analytical study. *PeerJ Computer Science* **2022**, *8*, e1032.

[7] Sultana, F.; Sufian, A.; Dutta, P. Advancements in image classification using convolutional neural network. In Proceedings of the 2018 Fourth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN). IEEE, 2018, pp. 122–129.

[8] Varshney, D.; Babukhanwala, B.; Khan, J.; Saxena, D.; Singh, A.K. Plant disease detection using machine learning techniques. In Proceedings of the 2022 3rd International Conference for Emerging Technology (INCET). IEEE, 2022, pp. 1–5.

[9] Bharate, A.A.; Shirdhonkar, M. A review on plant disease detection using image processing. In Proceedings of the 2017 International Conference on Intelligent Sustainable Systems (ICISS). IEEE, 2017, pp. 103–109.

[10] Dhanesha, R.; Shrinivasa, N.C. Segmentation of arecanut bunches using HSV color model. In Proceedings of the 2018 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT). IEEE, 2018, pp. 37–41.

[11] Dhanesha, R.; Naika, C.S.; Kantharaj, Y. Segmentation of arecanut bunches using YCgCr color model. In Proceedings of the 2019 1st International Conference on Advances in Information Technology (ICAIT). IEEE, 2019, pp. 50–53.

[12] Ghosal, S.; Sarkar, K. Rice leaf diseases classification using CNN with transfer learning. In Proceedings of the 2020 IEEE Calcutta Conference (CALCON). IEEE, 2020, pp. 230–236.

[13] Qassim, H.; Verma, A.; Feinzimer, D. Compressed residual-VGG16 CNN model for big data places image recognition. In Proceedings of the 2018 IEEE 8th annual computing and communication workshop and conference (CCWC). IEEE, 2018, pp. 169–175.

[14] Kumar, M.; Gupta, P.; Madhav, P.; et al. Disease detection in coffee plants using convolutional neural network. In Proceedings of the 2020 5th International Conference on Communication and Electronics Systems (ICCES). IEEE, 2020, pp. 755–760.

[15] Coulibaly, S.; Kamsu-Foguem, B.; Kamissoko, D.; Traore, D. Deep neural networks with transfer learning in millet crop images. *Computers in Industry* **2019**, *108*, 115–120.

[16] Wang, Y.; Wang, C.; Zhang, H.; Dong, Y.; Wei, S. Automatic ship detection based on RetinaNet using multi-resolution Gaofen-3 imagery. *Remote Sensing* **2019**, *11*, 531.

[17] V, P.K.; Rao, E.G.; Anitha, G.; Kumar, G.K. Plant Disease Detection using Convolutional Neural Networks. In Proceedings of the 2021 5th International Conference on Trends in Electronics and Informatics (ICOEI), 2021, pp. 1473–1476. https://doi.org/10.1109/ICOEI51242.2021.9453045.

[18] Saleem, M.H.; Khanchi, S.; Potgieter, J.; Arif, K.M. Image-based plant disease identification by deep learning meta-architectures. *Plants* **2020**, *9*, 1451.

[19] Tan, M.; Pang, R.; Le, Q.V. Efficientdet: Scalable and efficient object detection. In Proceedings of the Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 10781–10790.

[20] Hughes, D.; Salathé, M.; et al. An open access repository of images on plant health to enable the development of mobile disease diagnostics. *arXiv preprint arXiv:1511.08060* **2015**.

[21] Mohanty, S.P.; Hughes, D.P.; Salathé, M. Using deep learning for image-based plant disease detection. *Frontiers in plant science* **2016**, *7*, 1419.

[22] Ferentinos, K.P. Deep learning models for plant disease detection and diagnosis. *Computers and electronics in agriculture* **2018**, *145*, 311–318.

[23] Barbedo, J.G.A.; Koenigkan, L.V.; Halfeld-Vieira, B.A.; Costa, R.V.; Nechet, K.L.; Godoy, C.V.; Junior, M.L.; Patricio, F.R. Talamini, V.; Chitarra, L.G.; et al. Annotated plant pathology databases for image-based detection and recognition of diseases. *IEEE Latin America Transactions* **2018**, *16*, 1749–1757.

[24] Nie, X.; Wang, L.; Ding, H.; Xu, M. Strawberry verticillium wilt detection network based on multi-task learning and attention. *IEEE Access* **2019**, *7*, 170003–170011.

[25] Rahman, C.R.; Arko, P.S.; Ali, M.E.; Khan, M.A.I.; Apon, S.H.; Nowrin, F.; Wasif, A. Identification and recognition of rice diseases and pests using convolutional neural networks. *Biosystems Engineering* **2020**, *194*, 112–120.

[26] Thapa, R.; Zhang, K.; Snavely, N.; Belongie, S.; Khan, A. The Plant Pathology Challenge 2020 data set to classify foliar disease of apples. *Applications in Plant Sciences* **2020**, *8*, e11390.

[27] Chen, J.; Chen, J.; Zhang, D.; Sun, Y.; Nanehkaran, Y.A. Using deep transfer learning for image-based plant disease identification. *Computers and Electronics in Agriculture* **2020**, *173*, 105393.

[28] Wang, C.; Du, P.; Wu, H.; Li, J.; Zhao, C.; Zhu, H. A cucumber leaf disease severity classification method based on the fusion of DeepLabV3+ and U-Net. *Computers and Electronics in Agriculture* **2021**, *189*, 106373.

[29] Turkoglu, M.; Yanikoğ lu, B.; Hanbay, D. PlantDiseaseNet: Convolutional neural network ensemble for plant disease and pest detection. *Signal, Image and Video Processing* **2022**, *16*, 301–309.

[30] Falaschetti, L.; Manoni, L.; Di Leo, D.; Pau, D.; Tomaselli, V.; Turchetti, C. A CNN-based image detector for plant leaf diseases classification. *HardwareX* **2022**, *12*, e00363.

[31] Kumar, N.; Belhumeur, P.N.; Biswas, A.; Jacobs, D.W.; Kress, W.J.; Lopez, I.C.; Soares, J.V. Leafsnap: A computer vision system for automatic plant species identification. In Proceedings of the European conference on computer vision. Springer, 2012, pp. 502–516.

[32] Munisami, T.; Ramsurn, M.; Kishnah, S.; Pudaruth, S. Plant leaf recognition using shape features and colour histogram with K-nearest neighbour classifiers. *Procedia Computer Science* **2015**, *58*, 740–747.

[33] Li, Y.; Yang, J. Few-shot cotton pest recognition and terminal realization. *Computers and Electronics in Agriculture* **2020**, *169*, 105240.

[34] Tan, C.; Sun, F.; Kong, T.; Zhang, W.; Yang, C.; Liu, C. A survey on deep transfer

learning. In Proceedings of the International conference on artificial neural networks. Springer, 2018, pp. 270–279.
[35] Sagar, A.; Jacob, D. On using transfer learning for plant disease detection. *BioRxiv* **2021**, pp. 2020–05.
[36] Udawant, P.; Srinath, P. Leaf Diagnosis Using Transfer Learning. In Proceedings of the Applied Information Processing Systems. Springer Singapore, 2022, pp. 235–246.
[37] Sharma, M.; Nath, K.; Sharma, R.K.; Kumar, C.J.; Chaudhary, A. Ensemble Averaging of Transfer Learning Models for Identification of Nutritional Deficiency in Rice Plant. *Electronics* **2022**, *11*, 148.
[38] Maria, S.K.; Taki, S.S.; Mia, M.J.; Biswas, A.A.; Majumder, A.; Hasan, F. Cauliflower Disease Recognition Using Machine Learning and Transfer Learning. In Proceedings of the Smart Systems: Innovations in Computing. Springer Singapore, 2022, pp. 359–375.
[39] Paymode, A.S.; Malode, V.B. Transfer Learning for Multi-Crop Leaf Disease Image Classification using Convolutional Neural Network VGG. *Artificial Intelligence in Agriculture* **2022**, *6*, 23–33.
[40] Vallabhajosyula, S.; Sistla, V.; Kolli, V.K.K. Transfer learning-based deep ensemble neural network for plant leaf disease detection. *J. Plant Dis. Prot.* **2022**, *129*, 545–558.
[41] Elhassouny, A.; Smarandache, F. Smart mobile application to recognize tomato leaf diseases using Convolutional Neural 478 Networks. In Proceedings of the 2019 International Conference of Computer Science and Renewable Energies (ICCSRE), 2019, pp. 1–4.
[42] Andrianto, H.; Suhardi.; Faizal, A.; Armandika, F. Smartphone Application for Deep Learning-Based Rice Plant Disease Detection. In Proceedings of the 2020 International Conference on Information Technology Systems and Innovation (ICITSI), 2020, pp. 482 387–392.
[43] Valdoria, J.C.; Caballeo, A.R.; Fernandez, B.I.D.; Condino, J.M.M. iDahon: An Android Based Terrestrial Plant Disease Detection Mobile Application Through Digital Image Processing Using Deep Learning Neural Network Algorithm. In Proceedings of the 2019 4th International Conference on Information Technology (InCIT), 2019, pp. 94–98.
[44] Diah, N.M.; Jahim, M.L.M.; Roslan, N.A.M.; Ibrahim, Z.; Abdullah, A. Development of Mobile Application for Plant Disease Recognition using Convolutional Neural Network Method. In Proceedings of the 2021 6th IEEE International Conference on Recent Advances and Innovations in Engineering (ICRAIE), 2021, Vol. 6, pp. 1–6.
[45] Kaushik, M.; Prakash, P.; Ajay, R.; Veni, S.; et al. 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). IEEE, 2020, pp. 1125–1132.
[46] Shahidur Harun Rumy, S.M.; Arefin Hossain, M.I.; Jahan, F.; Tanvin, T. An IoT based System with Edge Intelligence for Rice Leaf Disease Detection using Machine Learning. In Proceedings of the 2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), 2021, pp. 1–6. https://doi.org/10.1109/IEMTRONICS52119.2021.9422499.
[47] Pandian, J.A.; Kumar, V.D.; Geman, O.; Hnatiuc, M.; Arif, M.; Kanchanadevi, K. Plant disease detection using deep convolutional neural network. *Applied Sciences* **2022**, *12*, 6982.
[48] Saheb, M.S.K.; Narayanan, B.; Rao, T.V.N. Review Of Machine Learning-Based Disease Diagnosis and Severity Estimation of Covid-19. *JOURNAL OF ALGEBRAIC STATISTICS* **2022**, *13*, 2267–2280.
[49] Ali, H.; Lali, M.; Nawaz, M.Z.; Sharif, M.; Saleem, B. Symptom based automated detection of citrus diseases using color histogram and textural descriptors. *Computers and Electronics in agriculture* **2017**, *138*, 92–104.
[50] Singh, H.; Rani, R.; Mahajan, S. Detection and classification of citrus leaf disease using hybrid features. In *Soft Computing: Theories and Applications*; Springer, 2020; pp. 737–745.

[51] Salamon, J.; Bello, J.P. Deep convolutional neural networks and data augmentation for environmental sound classification. *IEEE Signal processing letters* **2017**, *24*, 279–283.
[52] Taylor, L.; Nitschke, G. Improving deep learning with generic data augmentation. In Proceedings of the 2018 IEEE Symposium Series on Computational Intelligence (SSCI). IEEE, 2018, pp. 1542–1547.
[53] Albawi, S.; Mohammed, T.A.; Al-Zawi, S. Understanding of a convolutional neural network. In Proceedings of the 2017 international conference on engineering and technology (ICET). Ieee, 2017, pp. 1–6.
[54] Giusti, A.; Cireşan, D.C.; Masci, J.; Gambardella, L.M.; Schmidhuber, J. Fast image scanning with deep max-pooling convolutional neural networks. In Proceedings of the 2013 IEEE International Conference on Image Processing. IEEE, 2013, pp. 4034–4038.
[55] Baldi, P.; Sadowski, P.J. Understanding dropout. *Advances in neural information processing systems* **2013**, *26*.
[56] Jin, J.; Dundar, A.; Culurciello, E. Flattened convolutional neural networks for feedforward acceleration. *arXiv preprint arXiv:1412.5474* **2014**.
[57] Liang, X.; Wang, X.; Lei, Z.; Liao, S.; Li, S.Z. Soft-margin softmax for deep classification. In Proceedings of the International Conference on Neural Information Processing. Springer, 2017, pp. 413–421.
[58] 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 in brief* **2019**, *26*, 104340.
[59] Shorten, C.; Khoshgoftaar, T.M. A survey on image data augmentation for deep learning. *Journal of big data* **2019**, *6*, 1–48.
[60] Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* **2014**, *15*, 1929–1958.
[61] Buckland, M.; Gey, F. The relationship between recall and precision. *Journal of the American society for information science* **1994**, *45*, 12–19.
[62] Gillund, G.; Shiffrin, R.M. A retrieval model for both recognition and recall. *Psychological review* **1984**, *91*, 1.
[63] Swets, J.A. Measuring the accuracy of diagnostic systems. *Science* **1988**, *240*, 1285–1293.
[64] Liu, Z.; Xiang, X.; Qin, J.; Ma, Y.; Zhang, Q.; Xiong, N.N. Image recognition of citrus diseases based on deep learning. *CMC-COMPUTERS MATERIALS & CONTINUA* **2021**, *66*, 457–466.
[65] Elaraby, A.; Hamdy, W.; Alanazi, S. Classification of Citrus Diseases Using Optimization Deep Learning Approach. *Computational Intelligence and Neuroscience* **2022**, *2022*.
[66] Luaibi, A.R.; Salman, T.M.; Miry, A.H. Detection of citrus leaf diseases using a deep learning technique. *International Journal of Electrical and Computer Engineering* **2021**, *11*, 1719.